BASIS AI

Руководство администратора

Версия 2.3.7

Москва
2024

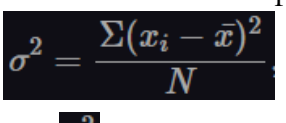
СОДЕРЖАНИЕ

1. Общие положения

1.1. Основные понятия и определения

Таблица 1 – Основные термины, используемые в документе, и их определения

Термин	Определение
Apache Kafka	Брокер сообщений, реализующий паттерн Producer-Consumer. Данные из одного и того же топика могут считываться множеством консьюмер-групп одновременно.
Apache Spark	Фреймворк с открытым исходным кодом для реализации распределенной обработки неструктурированных и слабоструктурированных данных.
BPMN	С англ. Business Process Model and Notation. Нотация, определяющая способ визуализации процессов в виде диаграмм с определенным набором блоков и взаимосвязей.
YOLO	С англ. You Only Look Once. Архитектура нейронных сетей, предназначенная для *детекции объектов* на изображении. Отличительной особенностью является подход к решению задачи детекции: исходное изображение сжимается таким образом, чтобы получить *квадратную матрицу* размером 13 на 13, в каждой клетке которой записана информация о наличии объекта и классе этого объекта на соответствующей части картинки. Таким образом, YOLO просматривает картинку один раз, что существенно увеличивает скорость обработки.
YOLOv5	Усовершенствованная пятая версия YOLO, реализованная на фреймворке PyTorch.
Временной ряд	Совокупность наблюдений, собранных за определенный временной интервал. Этот тип данных используется для поиска долгосрочного тренда, прогнозирования будущего и прочих видов анализа. Анализ временных рядов позволяет обнаруживать тенденции и закономерности в исследуемых процессах, строить прогнозы и предсказывать будущие изменения в бизнесе, на производстве, и в других областях.
Выборка	Случайное подмножество генеральной совокупности.
Датасет	С англ. *Data set*, набор данных. Коллекция из логических записей, хранящихся в виде *кортежа*. Набор данных можно сравнить с файлом, но в отличие от файла он является одновременно и каталогом, и файлом файловой системы, и не может содержать в себе другие наборы.

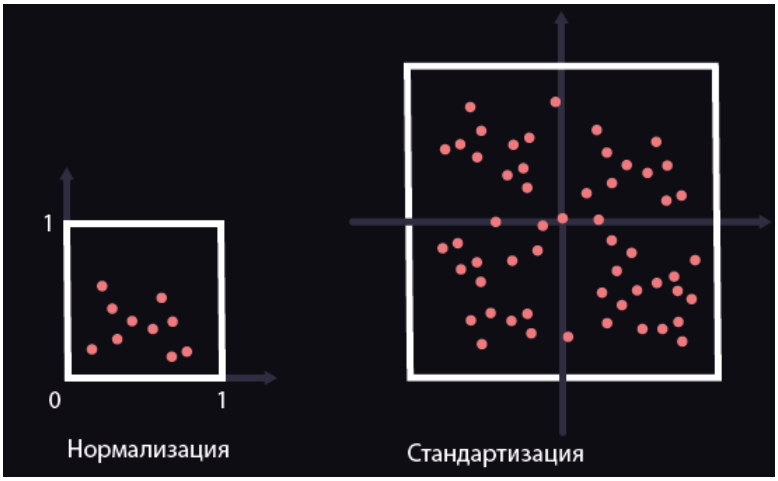
	Файловая система ориентирована на хранение записей, которые являются неделимыми единицами хранения. Множества записей объединяются в группы, которые и называются наборами данных. Записи в наборах данных используются приложениями, например, как входные данные. Так, записями набора данных могут быть как текстовые данные, так и изображения. К набору данных можно обратиться, указав точное место его хранения, или, если ранее для набора было зазервировано имя в файловой системе, по имени (второй вариант не реализован). Также *датасетом* называются данные, которые пользователь загрузил с локального компьютера, а Система при загрузке выполнила их предварительную обработку. Например, в систему загружается временной ряд в формате csv, и при загрузке он преобразуется в *структурированные данные* или по-другому датасет. Такой датасет пригоден для использования в моделях машинного обучения. В качестве *датасета* в настоящем документе также выступает набор изображений или файлов, которые используются для решения одной задачи.
Дерево решений	Инструмент прогнозного моделирования. Строится с помощью алгоритмического подхода, который разделяет набор данных на основе различных условий. Относится к классу обучения с учителем, используется для задач классификации и регрессии. Цель в том, чтобы создать модель, которая предсказывает значение целевой переменной, изучая правила принятия решений, выведенные из характеристик данных. Алгоритм: 1. Выбрать лучший атрибут, который разделяет наблюдения на группы. 2. Задать соответствующий вопрос. 3. Следовать по путям ответов. 4. Вернуться к шагу 1.
Дисперсия	Мера удаленности того или иного значения выборки от среднего значения. Рассчитывается по формуле:      Стандартное отклонение рассчитывается как квадратный корень из полученной цифры.

Интеллектуальный анализ текстов	Направление в искусственном интеллекте, целью которого является получение информации из коллекции текстовых документов, основываясь на применении эффективных в практическом плане методов машинного обучения и обработки естественного языка. Ключевыми группами задач ИАТ являются: категоризация текстов, извлечение информации и информационный поиск, изменение информации в коллекциях текстов. Категоризация документов заключается в отнесении документов из коллекции к одной или нескольким группам (классам, кластерам) схожих между собой текстов. Категоризация с участием человека называется *классификацией документов*, система ИАТ должна отнести тексты к уже определенным классам. Для этого производится обучение с учителем, для чего пользователь предоставляет системе ИАТ как множество классов, так и образцы документов, принадлежащих этим классам. Второй случай категоризации называется *кластеризацией документов*. При этом система ИАТ сама определяет множество кластеров, по которым могут быть распределены тексты (производится обучение без учителя). В этом случае пользователь сообщает системе ИАТ количество кластеров, на которое ему хотелось бы разбить обрабатываемую коллекцию (в алгоритме программы уже заложена процедура выбора признаков).
Категориальная переменная (качественные данные)	Это данные с ограниченным числом уникальных значений или *категорий* (например, пол, страна проживания, номер группы, категория товаров, и т.п.). Категориальные поля могут быть как текстовыми, так и числовыми, в которых категории закодированы *числовыми кодами* (например, 0=женский, а 1=мужской). Номинальные поля, порядковые поля и флаги являются категориальными полями. -*Набор* (номинальная переменная). Поле, значения которого представляют категории без естественного упорядочивания (например, подразделение компании, в котором работает сотрудник). -*Упорядоченный набор* (порядковая переменная). Поле, значения которого представляют категории с некоторым естественным для них упорядочением (например, оценки, представляющие степень удовлетворенности или уверенности, или баллы, оценивающие предпочтение). -*Флаг*. Поле или переменная с двумя отдельными значениями, например Да и Нет.
Классификация	Задача машинного обучения, которая ставит своей целью назначить метку класса наблюдениям из предметной области.

	Основные типы классификации: – бинарная классификация; – мультиклассовая классификация; – классификация по нескольким меткам; – несбалансированная классификация.
Кластеризация	Техника обучения без учителя, которая включает в себя группирование или кластеризацию точек данных. Чаще всего она используется для сегментации потребителей, выявления мошенничества, классификации документов и определении ключевых слов. Кластеризация (или кластерный анализ) – это задача разбиения множества объектов на группы, называемые кластерами. Главное отличие от классификации состоит в том, что перечень групп четко не задан и определяется в процессе работы алгоритма.
Машинное обучение (ML – Machine learning)	Тренировка математической модели на исторических данных для того, чтобы прогнозировать какое-то событие или явление на новых данных. То есть попытка заставить алгоритмы программ совершать действия на основе предыдущего опыта, а не только на основе имеющихся данных. Для обучения нужны исторические данные (обучающая выборка) и значение целевой переменной (то, что прогнозируем), которое соответствует заданным историческим данным. Модель наблюдает и находит зависимости между данными и целевой переменной. Эти зависимости используются моделью для нового набора данных, чтобы прогнозировать целевую переменную, которая неизвестна. Машинное обучение включает в себя целый набор методов и алгоритмов, которые могут предсказать какой-то результат по входным данным. Алгоритмов машинного обучения большое множество: одни эффективны для решения одного типа задач, вторые – для другого. **Суть технологии машинного обучения** Говоря в общем, машинное обучение – это обучение компьютерной программы или алгоритма постепенному улучшению исполнения поставленной задачи. Машинное обучение обозначает множество математических, статистических и вычислительных методов для разработки алгоритмов, способных решить задачу не прямым способом, а на основе поиска закономерностей в разнообразных входных данных. Решение вычисляется не по четкой формуле, а по установленной зависимости результатов от конкретного

	набора признаков и их значений. Например, если каждый день в течение недели земля покрыта снегом и температура воздуха существенно ниже нуля, то вероятнее всего, наступила зима. Поэтому машинное обучение применяется для диагностики, прогнозирования, распознавания и принятия решений в различных прикладных сферах: от медицины до банковской деятельности.
Мониторинг	Процесс наблюдения и регистрации данных о каком-либо объекте на неразрывно примыкающих друг к другу интервалах времени, в течение которых значения данных существенно не изменяются.
Мониторинг состояния	Наблюдение за состоянием объекта для определения и предсказания момента перехода в предельное состояние. Результат мониторинга состояния объекта представляет собой совокупность диагнозов составляющих его субъектов, получаемых на неразрывно примыкающих друг к другу интервалах времени, в течение которых состояние объекта существенно не изменяется. Принципиальное отличие от мониторинга параметров является наличие интерпретатора измеренных параметров в терминах состояния – экспертной системы поддержки принятия решений о состоянии объекта и дальнейшем управлении.
Наблюдение (строка, запись, точка, сущность)	Ценные данные, собираемые во время исследования или эксперимента. Вместе с масштабом анализа определяет совокупность. *Эмпирические исследования* – практические эксперименты с результатами на основе реального опыта, а не теории или убеждений. Основополагающим принципом *Науки о данных* является приоритет наблюдения над предположением. Типы наблюдений: — *Числовой*: целые (integer), вещественные (real number), числа с плавающей точкой (float). — *Булевый* (boolean) – принимает значения 1/0 (да/нет). — *Категориальный*. Например, жанры кино: комедия, ужасы, мелодрама. — *Текстовый*. — *Вектор*.
Нейронная сеть (или Искусственная нейронная сеть)	Представляет собой *математическую модель*, а также её программное или аппаратное воплощение, построенную по принципу организации и функционирования биологических нейронных сетей – сетей нервных клеток живого организма. Нейронные сети решают задачу: по точкам находят функцию. Происходит это путем *минимизации ошибки* – сводится к

	минимуму «расстояние» между значениями, предсказываемыми нейронной сетью, и значениями, которые наблюдаются. Под *архитектурой нейронной сети* понимается ее устройство – последовательность нейронов и связей между ними.
Нормализация	Техника преобразования значений признака, масштабирующая значения таким образом, что они расположены в диапазоне от 0 до 1.
Обучение с учителем	Контролируемое обучение – метод машинного обучения, при котором модель обучается на размеченных данных. Например, исследовав опухли, установив их размер, плотность и другие метрики, мы передаем эти данные модели с обязательной пометкой, какое наблюдение к какому строению (доброкачественному или злокачественному) относится. Алгоритмы контролируемого обучения подразделяются на следующие модели: классификация, регрессия.
Пайплайн	Последовательность стадий, внутри которых расположены задачи. Расположены они таким образом, что выход каждого элемента является входом следующего.
Признак	Объективная характеристика, характерная черта или свойство, которое может быть определено или измерено. В статистике независимые переменные X используются для предсказания зависимого признака Y
Регрессия (в математической статистике)	Математическое выражение, отражающее связь между зависимой переменной **y** и независимыми переменными **x**. Алгоритмы регрессии используются для контролируемого обучения моделей искусственного интеллекта. Модели обучают прогнозировать числовые значения целевых переменных.

Стандартизация	Техника преобразования значений признака, адаптирующая признаки с разными диапазонами значений к моделям машинного обучения, использующих дистанцию для прогнозирования. Это разновидность нормализации с использованием стандартизированной оценки преобразует значения так, что из каждого наблюдения каждого признака вычитается среднее значение и результат делится на стандартное отклонение этого признака: $$x_{i,\text{станд.}} = \frac{x_i - \mu}{\sigma}$$ $x_{i,\text{станд.}}$ x_i μ σ Преобразование необходимо, поскольку признаки датасета могут иметь большие различия между своими диапазонами, и для моделей машинного обучения это спровоцирует искаженное восприятие данных. Стандартизация, в отличие от нормализации, не имеет ограничивающего диапазона: 
Стандартизированная оценка	Метрика, характеризующая удаленность наблюдения от среднего значения совокупности данных.
Стандартное отклонение	Мера разброса в наборе числовых данных. Показывает, насколько далеко от среднего арифметического находятся точки данных. Чем меньше стандартное отклонение, тем более сгруппированы данные вокруг центра (среднего). Чем отклонение больше, тем больше разброс значений.
Тренировочные (обучающие) данные	Часть датасета, обучающая основа модели машинного обучения. Является одной из составляющих разделенного набора данных наряду с *тестовыми* и *валидационными* данными.

Валидационные (тестовые) данные	Часть датасета, основа для проверки работоспособности модели машинного обучения.
Числовая переменная	Numeric variable - переменная, выраженная различными видами чисел.
Целевая (зависимая) переменная	Target variable – признак датасета, который предстоит предсказывать модели машинного обучения. Зависимой ее называют, поскольку в ходе разведочного анализа данных выявляется корреляция между одной или несколькими переменными-предикторами и рассматриваемым целевым признаком.

2. Введение

2.1. Область применения

Программное обеспечение BASIS AI (далее – Платформа, платформа BASIS AI) обрабатывает структурированные и неструктурированные массивы данных, обучает модели ИИ (искусственного интеллекта) для создания баз знаний, предиктивной аналитики в промышленности и сфере финансов и здравоохранения. Платформа предоставляет все необходимые инструменты и ресурсы для выполнения полного цикла работы в области Data science.

2.2. Уровни подготовки пользователей

На Платформе предусмотрены две роли пользователей: *аналитик*, который моделирует работу бизнес-процесса, используя конструктор ИИ, и *оператор*, который имеет ограниченный доступ и может только просматривать результаты, полученные при запуске конструктора. В разделе **Администрирование** описаны правила и возможности настройки уровней доступа к разделам Системы для различных пользователей.

3. Регистрация и авторизация пользователя в Системе

Для начала работы с Платформой необходимо зарегистрироваться, для этого:

1. Перейдите на страницу регистрации:

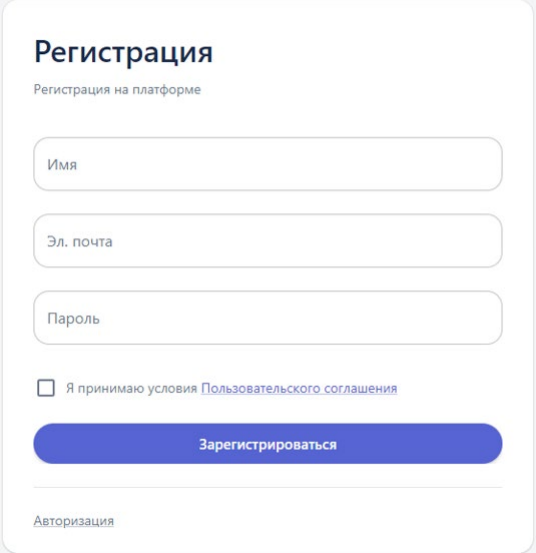


Рисунок 3.1 – Страница регистрации на Платформе

По ссылке «Авторизация» доступен переход на страницу авторизации.

2. Заполните следующие поля:
   - имя пользователя;
   - электронная почта;
   - пароль для входа в Систему.
3. Ознакомьтесь с условиями пользовательского соглашения. Регистрация возможна только при соглашении с данными условиями.
4. Нажмите кнопку «Зарегистрироваться».

Далее в БД MongoDB создается новая запись с уникальным идентификатором пользователя.

5. Вы можете сохранить связку логин и пароль для автоматического заполнения при последующей авторизации на текущем устройстве.

Если пользователь уже зарегистрирован в Системе, при входе осуществляется процедура авторизации**.** Для этого:

1. Перейдите на страницу авторизации:

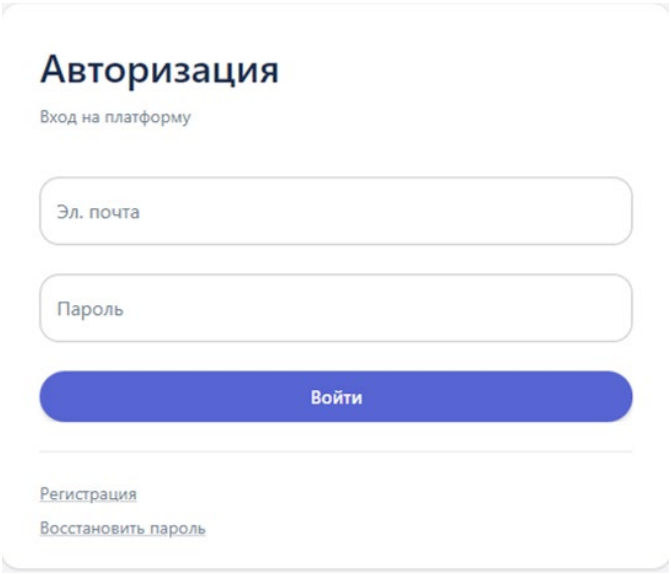


Рисунок 3.2 – Страница авторизации на Платформе

По ссылке «Регистрация» доступен переход на страницу регистрации.

По ссылке «Восстановить пароль» доступен переход на страницу восстановления пароля.

2. Введите следующую информацию:
    - адрес электронной почты;
    - пароль для входа в Систему.
3. Нажмите кнопку «Войти».

4. Личный кабинет пользователя

Личный кабинет – это персональная страница на Платформе, доступ к которой есть только у одного пользователя (его владельца).
Как работать с личным кабинетом:

1. Перейдите на страницу личного кабинета одним из способов:

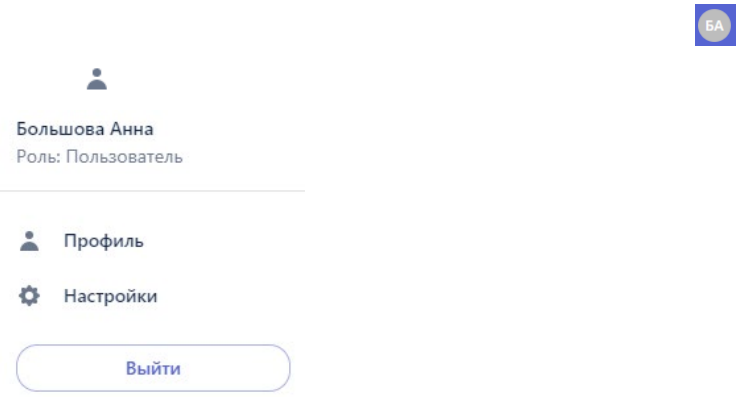


Рисунок 4.1 – Переход в личный кабинет

1.2. В левом верхнем углу главного окна в разделе с информацией о пользователе нажмите кнопку с инициалами/аватаркой пользователя:

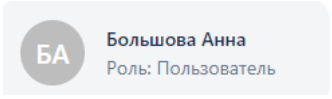


Рисунок 4.2 – Отображение данных пользователя

2. Откроется страница «Профиль»:

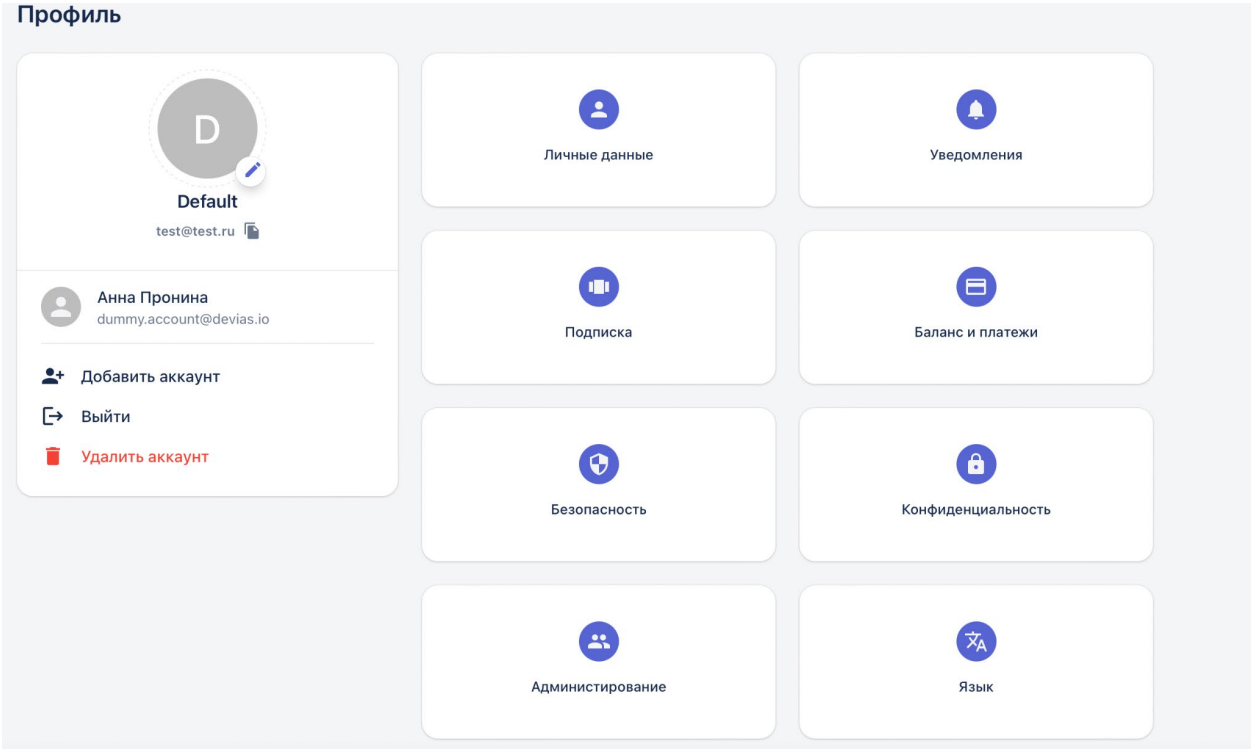


Рисунок 4.3 – Страница «Профиль»

На странице отображаются следующие элементы:

— аватарка пользователя (по умолчанию с инициалами имени пользователя);

— фамилия, имя и адрес электронной почты пользователя.

— блок команд для действий с аккаунтом.

— блок команд для управления профилем.

Для изменения адреса электронной почты в поле «Электронная почта» укажите новый адрес и нажмите кнопку «Сохранить» (далее сохранение новых настроек в профиле предполагается по умолчанию). Подтверждение e-mail не является обязательным действием при смене адреса.

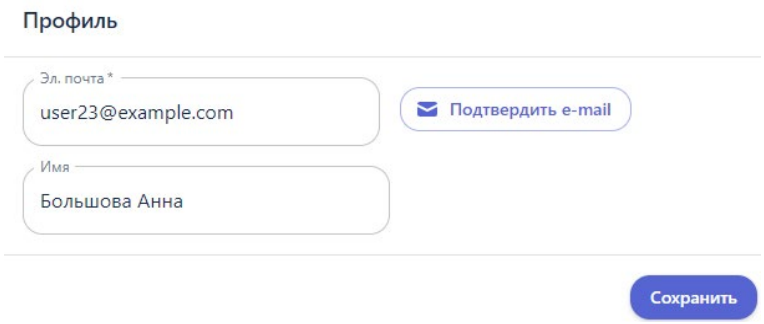


Рисунок 4.4 – Изменение адреса электронной почты в настройках профиля

Чтобы изменить имя пользователя, в поле «Имя» укажите свой новый логин. Тогда следующая авторизация на Платформе будет выполняться с новым логином.

Чтобы изменить аватар пользователя, нажмите кнопку «Редактировать» рядом с изображением аватара и выберите изображение. Убедитесь, что после окончания загрузки выбранное изображение будет установлено в качестве аватара.

Чтобы изменить параметры авторизации, выполните следующее:

— На странице профиля пользователя выберите **Личные данные**.

— В поле **Электронная почта** введите новый адрес электронной почты и нажмите кнопку **Сохранить изменения**. Убедитесь, что в указанном поле будет отображаться введенный вами адрес, а в указанный почтовый ящик будет отправлено уведомление для подтверждения нового адреса электронной почты.

— На странице профиля пользователя выберите **Безопасность**.

— Напротив поля **Пароль** нажмите кнопку **Изменить** и задайте новый пароль. Убедитесь, что выполняется вход в Систему с использованием нового пароля.

5. Интерфейс Платформы

5.1. Меню интерфейса

Пункты меню имеют древовидную структуру и представлены в виде вложенных папок:

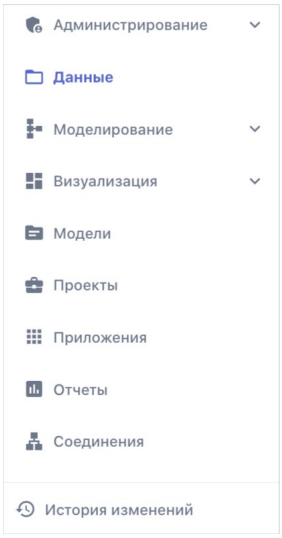


Рисунок 5.1 – Пункты меню платформы **BASIS AI**

Состав:

Таблица 2 – Описание пунктов меню

	Администрирование	Данный раздел позволяет создавать ролевые модели пользователей - *группы*, согласно которым разделяются уровни доступа к разным модулям системы. Тут же осуществляется управление и назначение ролей всем пользователям системы. В разделе также реализована возможность настройки *каналов уведомлений* (это может быть email или telegram), которые могут быть использованы для получения автоматических оповещений от системы в случае выполнения заданных условий.
	Данные	Данный блок предназначен для загрузки в Систему *входных данных*, это могут быть файлы в различных форматах: — **Таблицы в форматах csv, txt, xlsx, xls** - табличные данные для создания интеллектуальной базы знаний, в том числе временные ряды. *Временной ряд* – это собранный в разные моменты времени статистический материал о значении каких-либо параметров исследуемого процесса. Возникают временные ряды в результате измерения некоторого показателя. Это могут быть как показатели технических систем, так и показатели природных

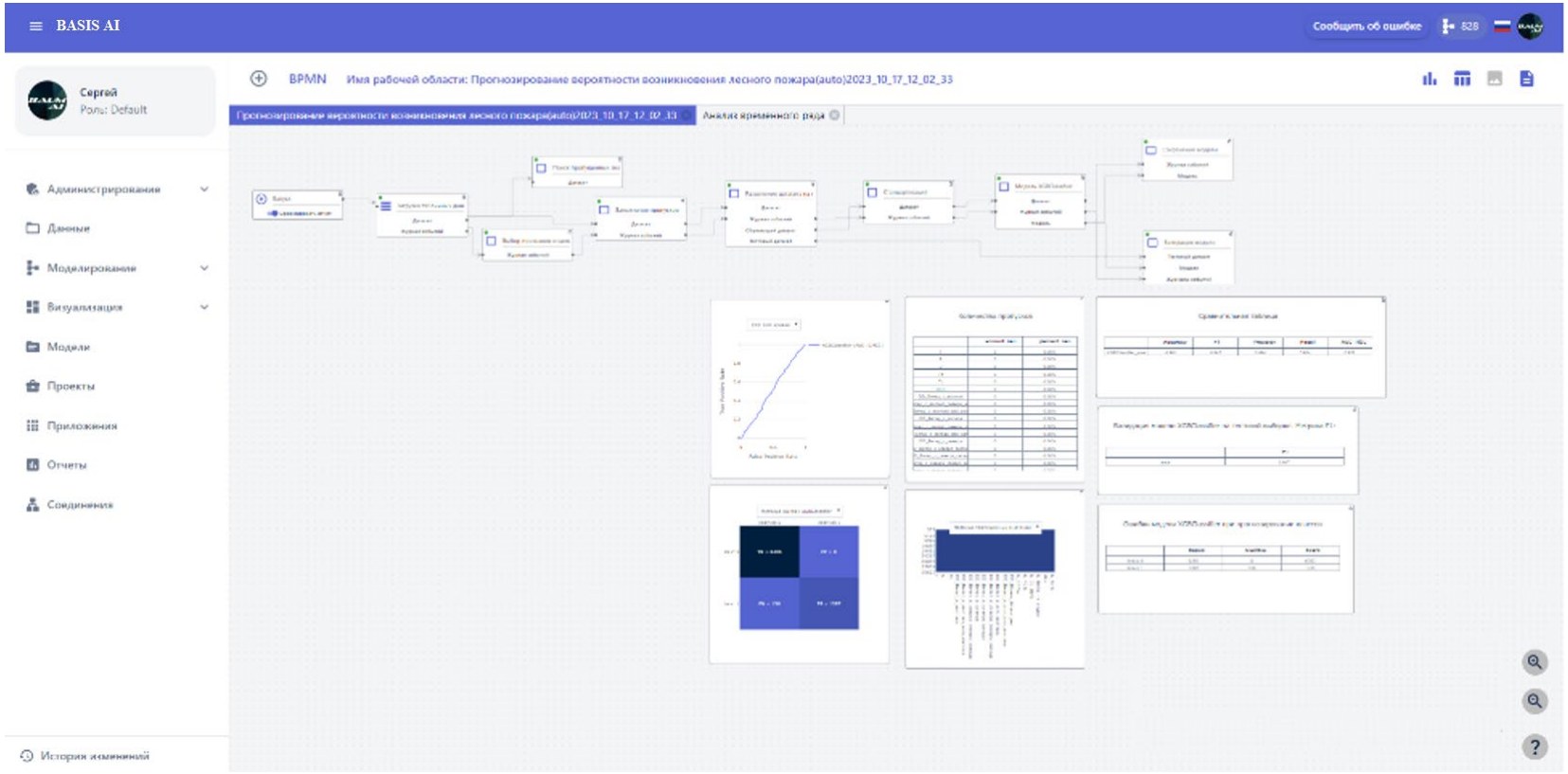
		(погодные условия), социальных, экономических и других систем. Пример временного ряда – показания датчиков на производстве, анализ которых позволяет прогнозировать выход за критические отметки целевых переменных, принять меры и предотвратить возможную поломку оборудования или аварию. Уже загруженные в систему файлы с данными, преобразованные в датасеты, отображаются в формате таблицы (реализовано только для файлов с расширением *.csv*). Эти датасеты используются в качестве входных данных для обучения модели искусственного интеллекта. — **Изображения в форматах: jpeg, jpg, png.** — **Видео в форматах: avi, mp4.** — **Текстовые файлы в форматах: txt, doc, docx**. Примеры задач для работы с текстом – классификация текста (например, определение авторства), распознавание и оцифровка рукописного текста, чтение и анализ новостного фона, определение тенденций в науке по научным статьям «Скопус», и т.д.
	**Моделирование**	Сердце программы, которое позволяет описывать бизнес-процессы и выполнять целевые действия. — **Рабочая область**. Для описания бизнес-процессов заказчика на рабочей области создаются *элементные блок-схемы*, которые позволяют выстраивать цепочки, взаимосвязи, условия и т.д. Блок-схема отвечает на вопрос «Что делает процесс». — **Сохраненные рабочие области**. Рабочие области с ранее созданными блок-схемами. Есть возможность в любое время вернуться и продолжить работу над блок-схемой.
	**Визуализация**	— **Дашборды**. Динамически настраиваемые дашборды. Содержат загруженные и преобразованные массивы данных, представленные в виде таблиц, графиков, гистограмм. — **Сохранённые дашборды**. Дашборды, созданные пользователем или группой пользователей.
	**Модели**	*Модель на основе машинного обучения* – это абстракция, которая обучена распознаванию определенного типа закономерностей. Хранится в виде файла. Для обучения модели нужны исторические данные (обучающая выборка) и значение целевой переменной (то, что прогнозируем), которое соответствует заданным

		историческим данным. С помощью алгоритмов машинного обучения модель наблюдает и находит зависимости между данными и целевой переменной. Эти зависимости используются моделью для нового набора данных (тестовой выборки), чтобы прогнозировать целевую переменную, которая неизвестна. Все модели разделяются на обучение с учителем и без учителя. *Обучение с учителем* подразделяется на две подкатегории: регрессия и классификация. В *регрессионных моделях* вывод является непрерывным. Наиболее распространенные типы регрессионных моделей – линейная регрессия, деревья решений, случайный лес, нейронная сеть. В *классификационных моделях* вывод является дискретным. Наиболее распространенные типы классификационных моделей – логистическая регрессия, метод опорных векторов. В отличие от обучения с учителем, *обучение без учителя* используется для того, чтобы сделать выводы из входных данных без отсылок на отмеченные результаты. Два основных метода, используемых в обучении без учителя, включают *кластеризацию* и *снижение размерности*. **Сохранённые модели** – обученные модели, которые используются в моделировании бизнес-процессов с данными в режиме реального времени.
	Проекты	Сущность «Проект» объединяет в себе: загруженные в Систему файлы, созданные рабочие области, обученные модели ИИ, визуализацию результатов на дашбордах и сформированные отчеты. Так Система позволяет консолидировать всю информацию по проекту в одном разделе для систематизации и удобного доступа пользователей.
	Приложения	Приложение содержит «упакованную» обученную модель ИИ – без необходимости разработки отдельного кода для создания приложения с моделью. Такое приложение можно скачать и развернуть за пределами программы, интегрировать с внешними системами, настроить получение входных данных и выполнить прогнозы.
	Отчеты	Подготовленные по установленной форме отчеты. Могут хранится результаты ИИ исследований, результаты спроектированного бизнес-процесса.
	Соединения	Раздел «Соединения» предназначен для настройки подключения Платформы к внешним источникам данных

		с целью получения данных из других систем. В этом разделе пользователь может создавать новые и просматривать уже созданные коннекторы. *Коннектор* – это сущность, которая объединяют в себе источник подключения и запрос на получение данных из него.

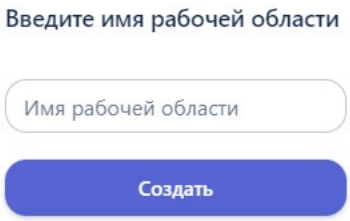
5.2. Окно построения модели ИИ

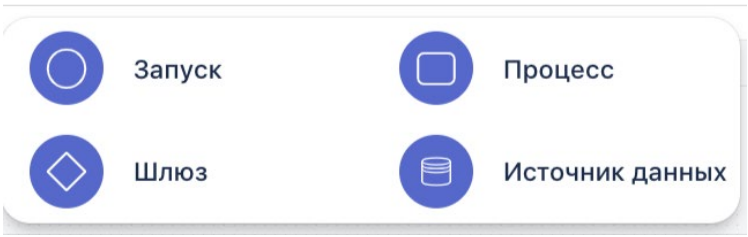
Ниже приведен иллюстративный пример страницы Платформы, где представлены различные компоненты интерфейса:



На верхней панели инструментов рабочей области представлены следующие кнопки:

Таблица 4 – Кнопки панели инструментов рабочей области

<table>
<tr>
<td></td>
<td>Создание рабочей области</td>
<td>При нажатии на кнопку открывается форма, в которой нужно указать название рабочей области:
 

Одна рабочая область может содержать несколько блок-схем. Реализована возможность запуска как отдельной блок-схемы, так и всех блок-схем на рабочей области.
Рабочую область можно добавить в проект, таким образом организовав доступ к рабочей области для всех пользователей этого проекта.</td>
</tr>
<tr>
<td></td>
<td>Добавить элемент</td>
<td>При нажатии на кнопку « BPMN » открывается блок, содержащий графические элементы нотации BPMN 2.0:
 

Чтобы добавить элемент на блок-схему достаточно нажать на кнопку с элементом.
Типы элементов нотации BPMN 2.0:
<table>
<tr><td>Запуск</td><td>Инициирующее событие – главный элемент, обозначающий начало блок-схемы.</td></tr>
<tr><td>Процесс</td><td>Действие, выполняемое в ходе бизнес-процесса. Является основным элементом.</td></tr>
<tr><td>Шлюз</td><td>Действие предназначено для разветвления алгоритма по веткам – прохождение сценария по каждой из веток выполняется при определенных условиях. Когда выполняется раздвоение потока операций, прописывается одно условие, и при его выполнении сценарий проходит по одной из веток.</td></tr>
<tr><td>Источник данных</td><td>Объект данных – это информационный объект (датасет, файл, модель и т.д.), который</td></tr>
</table>
</td>
</tr>
</table>

		обрабатывается и передается в ходе выполнения бизнес-процесса. Действие предназначено для выбора уже загруженной и/или сохраненной в Системе сущности.
	**Графики**	После успешной отработки пайплайна здесь будет отображаться список доступных для визуализации графиков.
	**Таблицы**	После успешной отработки пайплайна здесь будет отображаться список доступных для визуализации таблиц.
	**Изображения**	После успешной отработки пайплайна здесь будет отображаться список доступных для визуализации изображений.
	**Описание**	Отображается описание обученной модели ИИ.

- **Карточка элемента блок-схемы**. В карточке прописываются условия обработки данных – откуда берутся данные, какие операции над ними выполняются. Для перехода в карточку нажмите кнопку «Настройки» на элементе. Чтобы скрыть карточку элемента, используйте кнопку «Свернуть».

  Карточка содержит *программный код* с основными блоками:
  - входные данные (источник входных данных);
  - преобразование данных;
  - выходные данные – куда и в каком виде передаются преобразованные данные.
- **Вкладки с рабочими областями** - вы можете работать с несколькими рабочими областями одновременно, для удобного перехода между ними используйте вкладки.

5.3. Настройка внешнего вида интерфейса

Чтобы изменить настройки интерфейса, кликните на свою аватарку и перейдите в пункт меню «Настройки». Откроется окно, в котором можно изменить следующее:

– **Тема** – светлая или темная.

– **Адаптивные размеры шрифтов** – адаптация размеров шрифтов программы для работы с ней в мобильных устройствах.

– **Компактный вид -** фиксированная ширина на некоторых экранах.

– **Закругленные углы**. Если выбрать настройку, то окна в программе будут иметь закругленные углы. Иначе окна будут иметь прямые углы.

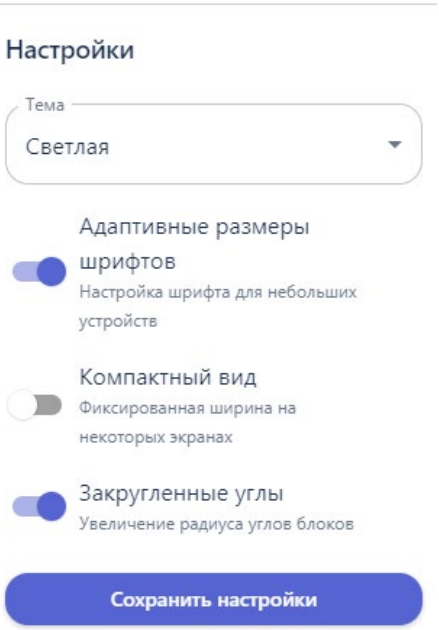


Рисунок 5.3 – Настройки внешнего вида интерфейса программы

Чтобы применить выбранные настройки к интерфейсу нажмите кнопку «Сохранить настройки».

5.4. Встроенные функции

Каждая функция относится к одному из модулей Системы: *модулю препроцессинга входных данных*, *модулю машинного обучения*, *модулю нейронных сетей/глубокого обучения*, *модулю анализа данных* или *модулю визуализации*.

Описание всех функций представлено в Приложении 1 настоящего документа.

За каждым типом блока закреплен определенный набор функций:

5.4.1 Функции элемента «Источник данных»

Таблица 5 – Набор функций элемента «Источник данных»

Группа	Функция
Загрузка данных	1. Загрузка изображений для object detection
	2. Загрузка модели
	3. Загрузка изображений для классификации
	4. Загрузка табличных данных из коннектора

Группа	Функция
	5. Загрузка графа
	6. Загрузка табличных данных
	7. Загрузка текстовых файлов для классификации
	8. Загрузка текстовых файлов для кластеризации
	9. Загрузка модели сегментации
Spark	10. Загрузка модели (Spark)
	11. Загрузка табличных данных из папки CSV (Spark)
	12. Загрузка табличных данных из файла CSV (Spark)
	13. Загрузка табличных данных из коннектора (Spark)
Оптимизация	14. Простой генетический алгоритм
Глубокое обучение	15. Сегментация(обучение)

5.4.2 Функции элемента «Процесс»

Таблица 6 – Набор функций элемента «Процесс»

Группа	Подгруппа	Функция
Анализ данных	-	1. Косинусное расстояние
		2. Матрица корреляции
		3. Визуализация Real Time
		4. Поиск и удаление выбросов
		5. Визуализация временного ряда (аномалии)
		6. Запись в датасет логирования
		7. Выбор признаков и целевых признаков

Группа	Подгруппа	Функция
		8. Поиск пропущенных значений
		9. Анализ временных рядов
	Тесты на стационарность временного ряда	10. Тест Дики-Фуллера
	Препроцессинг	11. Стандартизация
		12. One-Hot Encoding
		13. Кодирование целевого признака
		14. Создание признаков для временного ряда
		15. Дифференцирование временного ряда
		16. Стабилизация дисперсии
		17. Порядковое кодирование категориальных признаков
		18. Нормализация
	Тесты на нормальность распределения	19. Коэффициент асимметрии Skewness
	Предобработка данных	20. Сглаживание временного ряда
		21. Срез временного ряда по индексу
		22. Лемматизация текста
		23. Фильтрация текстового шума
		24. Векторизация текста
		25. Заполнение пропусков

Группа	Подгруппа	Функция
	Загрузка данных	26. Преобразование данных во временной ряд
Управление моделями	**-**	27. Сохранение модели
		28. Сохранение модели классификации изображений
		29. Сохранение модели YoloV5
		30. Сохранение модели Spark
		31. Сохранение модели сегментации
Машинное обучение	-	32. Разделение датасета на обучающую и тестовую выборки
		33. Расчет метрик
		34. Прогноз модели
		35. Валидация модели
	Классификация	36. Дерево решений для классификации
		37. Случайный лес для классификации
		38. Categorial Naive Bayes
		39. Multinomial Naive Bayes
		40. Complement Naive Bayes
		41. Gaussian Naive Bayes
		42. Bernoulli Naive Bayes
		43. Логический анализ данных
		44. Стекинг классификация

Группа	Подгруппа	Функция
		45. Модель XGBClassifier
		46. Логистическая регрессия
	Обучение без учителя	47. Метод локтя K-Means
		48. Изоляционный лес
		49. Кластеризация K-Means
		50. Агломеративная иерархическая кластеризация
		51. Кластеризация DBSCAN
	Регрессия	52. Линейная регрессия
		53. Полиномиальная регрессия
		54. Дерево решений для регрессии
		55. Случайный лес для регрессии
		56. Метод опорных векторов для регрессии
		57. Байесовская гребневая регрессия
		58. Метод k-ближайших соседей для регрессии
		59. Стекинг регрессия
	Авторегрессия	60. ARIMA/SARIMAX
	Работа с текстом	61. Автореферирование текста
Отправка уведомлений	-	62. Отправка уведомлений
Spark	-	63. Разделение датасета на обучающую и тестовую выборки
		64. Валидация модели (Spark)

Группа	Подгруппа	Функция
		65. Выбор признаков и целевых признаков (Spark)
		66. Прогноз модели (Spark)
		67. Сохранение датасета Spark в CSV
		68. Косинусное расстояние (Spark)
	Препроцессинг	69. Порядковое кодирование признаков
		70. Нормализация признаков
	Классификация	71. Модель градиентного бустинга Spark для бинарной классификации
	Кластеризация	72. Кластеризация Spark DBSCAN
Глубокое обучение	**Сегментация (тестирование)**	73. Сегментация (тестирование)
	Сегментация	74. Сегментация (Прогноз)
	Классификация	75. Классификация изображений
		76. Классификация (табличные данные)
	Регрессия	77. Регрессия (табличные данные)
	Обнаружение объектов	78. YOLOv5
	-	79. Валидация модели классификации изображений

“–” в таблице означает, что у функции нет подгруппы, и она напрямую относится к группе функций.

6. Загрузка данных в систему

Сразу после авторизации в Системе открывается её начальная страница – раздел «Данные».

Раздел «Данные» имеет внешний вид аналогичный проводнику файлов в операционной системе компьютера. Пользователи имеют возможность создавать папки, загружать файлы и делать структуры, удобные для личного использования. По умолчанию для нового пользователя раздел Данные выглядит следующим образом:

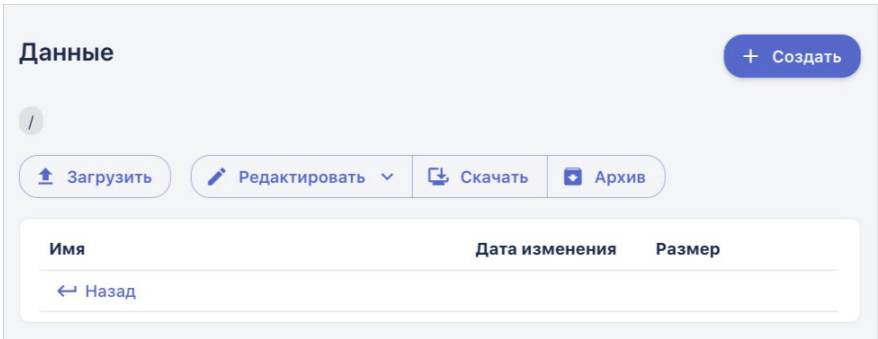


Рисунок 6.1 – Вид раздела Данные

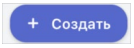
Основные кнопки для работы с разделом:

1. **«Создать»** - позволяет создавать папки и разметки.
2. **«Загрузить»** - осуществление непосредственно загрузки файлов в систему
3. ***«Редактировать»*** - *не работает в данной версии системы*
4. ***«Скачать»*** - *не работает в данной версии системы*
5. ***«Архив»*** - *не работает в данной версии системы*

В данном разделе вы можете создавать папки, добавлять в них файлы, создавая удобную и понятную структуру. Также в разделе осуществляется создание разметки для видео и изображений, и добавление папок классификации для дальнейшего использования в обучении искусственного интеллекта.

Вы можете загружать в систему файлы разных форматов и типов. Данные можно загружать напрямую в основную директорию, или создавая новые папки внутри.

6.1. Создание новой папки

Для создания новой папки нажмите на кнопку «Создать» 

и в открывшемся окне в поле «Тип» выберите «Категория» (так в системе называются папки), а в поле «Название» введите название будущей папки, например, «Табличные данные»:

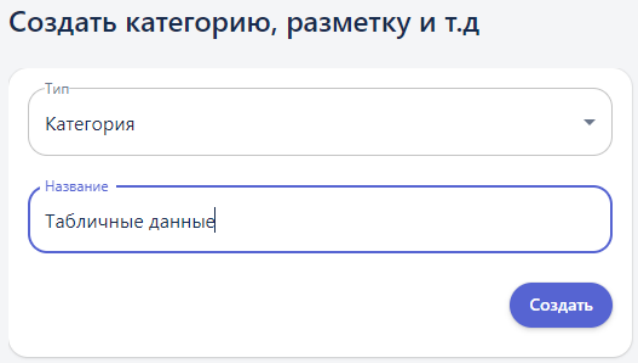


Рисунок 6.2 – Создание новой папки в разделе Данные

После этого папка появится в разделе Данные:

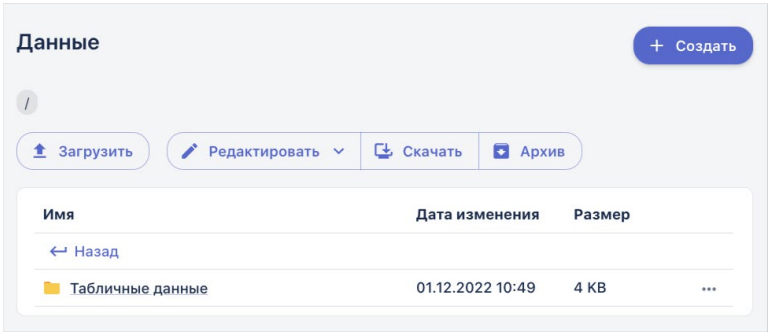


Рисунок 6.3 – Новая папка в разделе Данные

Обратите внимание, что папка будет добавлена в том разделе, из которого вы нажали кнопку «Создать». Т.е. далее вы можете перейти в папку «Табличные данные» и создать внутри еще одну категорию-папку.

6.2. Загрузка файлов

Для того чтобы загрузить файлы в папку, кликните на неё и перейдите в ее содержимое. Далее нажмите кнопку «Загрузить»:

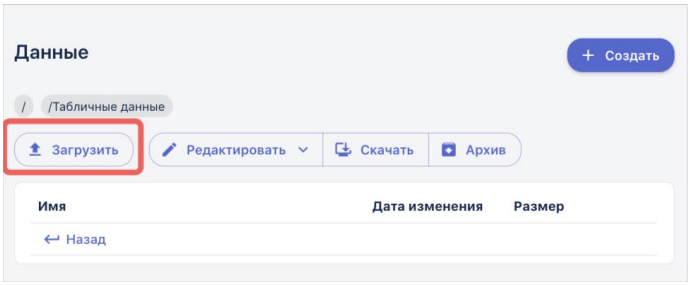


Рисунок 6.4 – Загрузка файлов в папку

В открывшемся окне левой кнопкой мыши нажмите на ссылку выбора файла. Указать путь к файлу для загрузки на вашем ПК. Второй вариант – перенести файлы с локального компьютера в этот раздел по технологии «drag n drop».

Выбранные файлы отобразятся в нижней части окна загрузки:

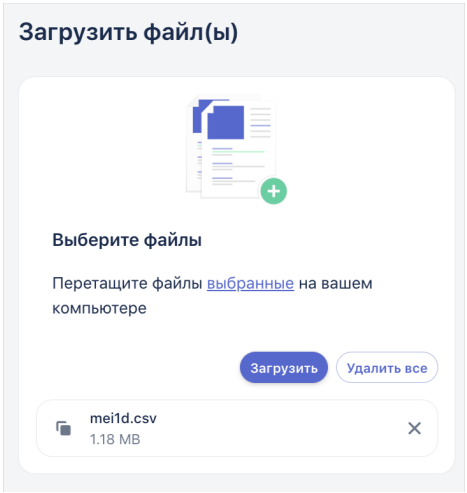


Рисунок 6.5 – Отображение выбранного файла

При необходимости выбранные файлы можно удалить по одному, нажав на крестик, или все вместе, нажав кнопку «Удалить все».

Для того чтобы загрузить выбранные файлы, нажмите на кнопку «Загрузить». Файлы отобразятся в папке:

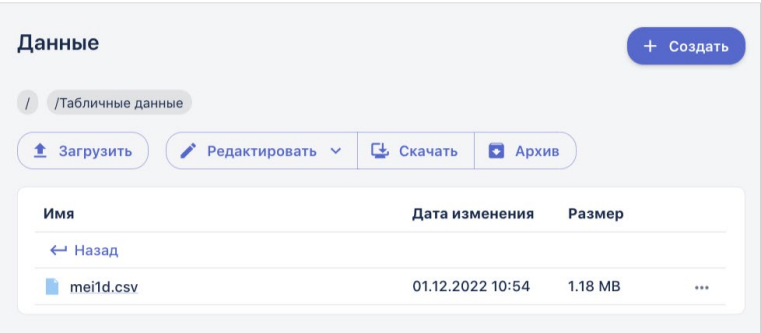


Рисунок 6.6 – Загруженный файл в разделе Данные

6.3. Предпросмотр данных

В системе реализована функция предпросмотра загруженных файлов следующих форматов:

- табличные данные формата .csv
- изображения в форматах: jpeg, jpg, png.
- видео в форматах: avi, mp4.

Если кликнуть на файл с изображением или видео - его предпросмотр начнется прямо в окне, чтобы выйти из режима предпросмотра - просто кликните в любую область экрана за пределами медиафайла.

При предпросмотре табличных данных в формате csv, в нижней части экрана отобразится окно с тремя вкладками, содержащими описательно статистический анализ.

На вкладке «Подробнее» (показано на примере файла с временным рядом) пользователь имеет возможность увидеть:

- размер датасета (количество строк, столбцов)
- гистограммы распределения каждого признака (числового)

Рисунок 6.7– Вкладка «Подробнее» окна отображения данных о датасете

На вкладке «Компактно» отображается состав строк и столбцов файлов:

area	T	P	U	Ff	Td
0	-0.59	763.32	84.87	4.37	-2.92
0	-3	770.51	74.12	1.12	-7.15
0	-2.95	769.77	89.62	2.62	-4.52
0	1.85	766.77	85.37	3.5	-0.45
0	1.48	762.78	83.5	3	-1.12
0	0.8	772.26	89.75	1.5	-0.82
0	-1.32	776.48	84.62	2.25	-3.7
0	-0.9	775.55	81.37	1.62	-3.98
1	-2.97	770.06	69.66	2.44	-8.33
0	-3.64	770.27	59.87	2.5	-11.17

Рисунок 6.8 – Вкладка «Компактно» окна отображения данных о датасете

На вкладке «Столбцы» для каждого признака отображается:

- количество пропусков (Missing) в абсолютном и в процентном выражении, под пропуском понимается пустая ячейка в таблице
- среднее значение (Mean)
- стандартное отклонение (Std. Deviation)
- квантили (quantilies):

Рисунок 6.7– Вкладка «Столбцы» окна отображения данных о датасете

6.4. Взаимодействие с данными

Загруженный файл или созданную папку можно скачать, скачать архивом, переименовать или удалить. Для этого нажмите на три точки в правой части раздела и выберите соответствующую кнопку:

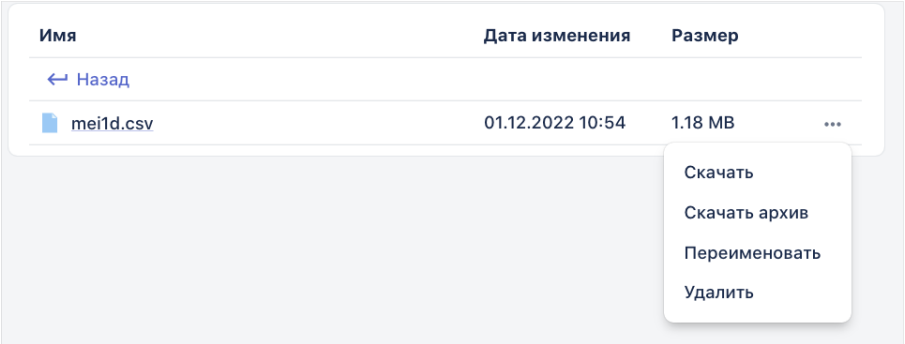


Рисунок 6.8 – Удаление или скачивание файла из раздела Данные

Вы можете удалять файлы по отдельности или сразу целую папку.

Для навигации между файлами и папками можно воспользоваться кнопкой «Назад»: 

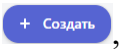
.

6.5. Создание датасетов и загрузка данных для решения задач классификации

Задача классификации относится к классу задач «обучение с учителем», которое предполагает наличие набора *размеченных* данных для тренировки модели. Наличие размеченного датасета означает, что каждому примеру в обучающем наборе соответствует ответ, который должен получить алгоритм. С помощью BASIS AI можно решать задачи классификации следующих типов данных:

- изображений
- текстов
- табличных данных

В задачах классификации алгоритм предсказывает *дискретные значения*, соответствующие номерам классов, к которым принадлежат объекты. В обучающем датасете каждый объект будет иметь соответствующую метку. Пользователь должен подготовить данные перед подачей их в алгоритм машинного обучения. В данном разделе рассмотрим, как создаются группы и классы в разделе «Данные».

1. Перейдите в пункт меню «Данные».
2. Чтобы создать *новую группу* нажмите кнопку 

, откроется *форма создания новой группы*:

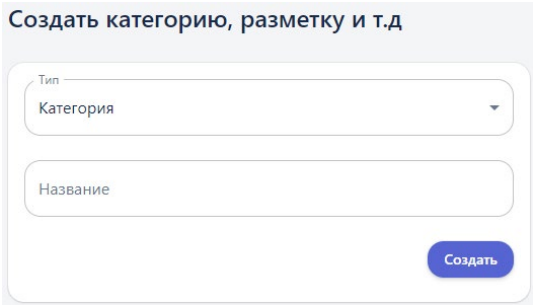


Рисунок 6.9 – Создание новой группы

3. В открывшейся форме в поле «Тип» выберите значение «Категория», в поле «Название» введите название новой группы, и нажмите кнопку «Создать».

Примечание – *здесь под группой имеется в виду папка, в которую будут загружаться данные для решения задачи. Пользователь создает такие папки самостоятельно, аналогично тому как организовывает хранение информации на компьютере в ОС Windows.*

4. Далее в папке, созданной в шаге 2, создайте две *подпапки* – «Train» и «Test» (в описании приведены примеры названий, отражающие их суть, нет необходимости давать папкам аналогичные название, главное, чтобы вам было понятно назначение каждой из них). В группу «Train» будут загружаться файлы для обучения будущей модели машинного обучения, а в группу «Test» – файлы для проверки ‘качества’ уже обученной модели. Качество алгоритма оценивается тем, насколько точно он может правильно классифицировать объекты из валидационной выборки.
5. Следующим этапом в папке «Train» создаются еще подпапки, отражающие *классы*. При этом количество классов равно двум, если решается задача *бинарной классификации*, и больше двух – для *многоклассовой классификации*. В каждую из этих папок загружаются объекты – файлы, содержащие данные соответствующего класса. Например, создается папка-класс «самолеты» и туда загружаются изображения самолетов. Количество данных для обучения модели должно быть достаточным, оно определяется аналитиком самостоятельно, и зависит от направленности и специфики решаемой задачи.
6. Повторите действия из шага 4 для папки «Test». **Важно** – Названия создаваемых классов в тестовой выборке должны абсолютно совпадать с названиями классов в обучающей выборке. Как правило, данные для валидации составляют 20% от общего объема выборки. Итоговая структура группы:

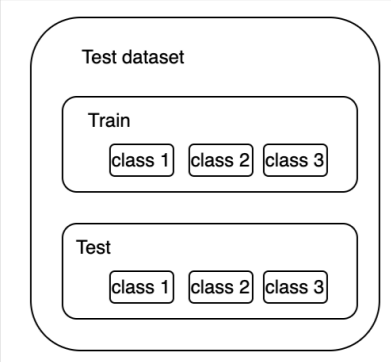


Рисунок 6.10 – Принцип создания групп и классов для решения задач классификации

7. Последний шаг – это присвоение метки «Классификация» созданным группам. Такая метка назначается папкам «Train» и «Test» (примеры названий). Это действие позволяет сформировать *датасеты*. Для этого нажмите на три точки в строке с названием папки и выберите действие «Классификация»:

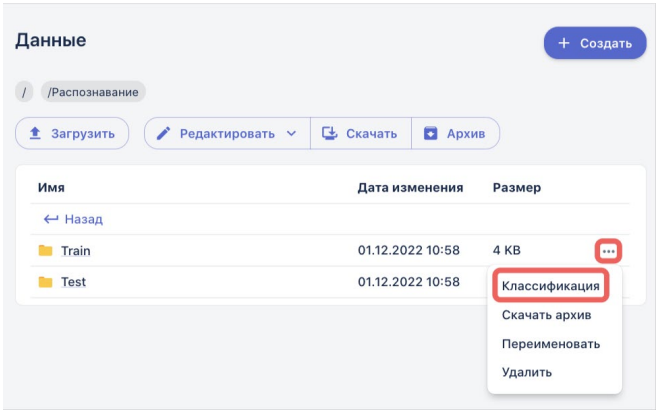


Рисунок 6.11 – Присвоение метки ‘Классификация’

На этом этап подготовки данных для решения задачи классификации завершается. Когда вы будете строить блок-схему и выбирать исходные данные, папки с меткой ‘Классификация’ будут доступны для выбора.

Для удаления ошибочно созданных или неактуальных папок используется действие «Удалить», а для сохранения папки на локальном компьютере в виде архива – действие «Скачать архив».

7. Создание модели ИИ

В разделе «Моделирование» осуществляется процесс построения блок-схем - соединение последовательных функции процессов анализа, обработки и преобразования исходных данных для построения и обучения моделей искусственного интеллекта. Построение таких блок-схем осуществляется на рабочих областях.

7.1 Создание новой и открытие сохраненной рабочей области

7.1.1. Создание новой рабочей области

1. Перейдите в пункт меню системы Моделирование -> Рабочая область:

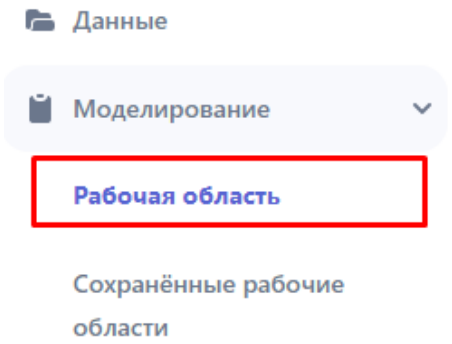


Рисунок 7.1.1 – Переход на рабочую область

Откроется страница с пустой рабочей областью:

Рисунок 7.1.2 – Пустая рабочая область

2. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка ⊕).

В открывшейся форме введите название новой рабочей области и нажмите кнопку «Создать»:

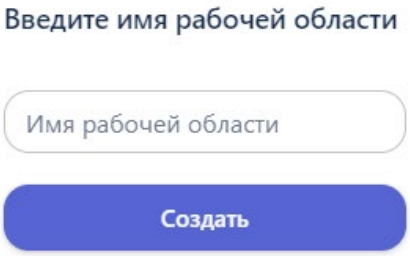


Рисунок 7.1.3 – Создание новой рабочей области

На панели инструментов отобразится название созданной рабочей области:

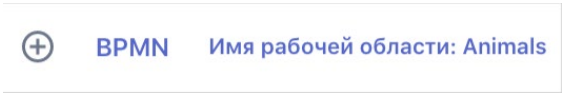


Рисунок 7.1.4 – Отображение названия рабочей области

После присвоения названия рабочая область автоматически сохраняется в раздел «Сохраненные рабочие области».

7.1.2. Открытие сохраненной рабочей области

Для того чтобы открыть ранее созданную рабочую область, перейдите в пункт меню системы Моделирование -> Сохраненные рабочие области. Откроется страница «Рабочие области»:

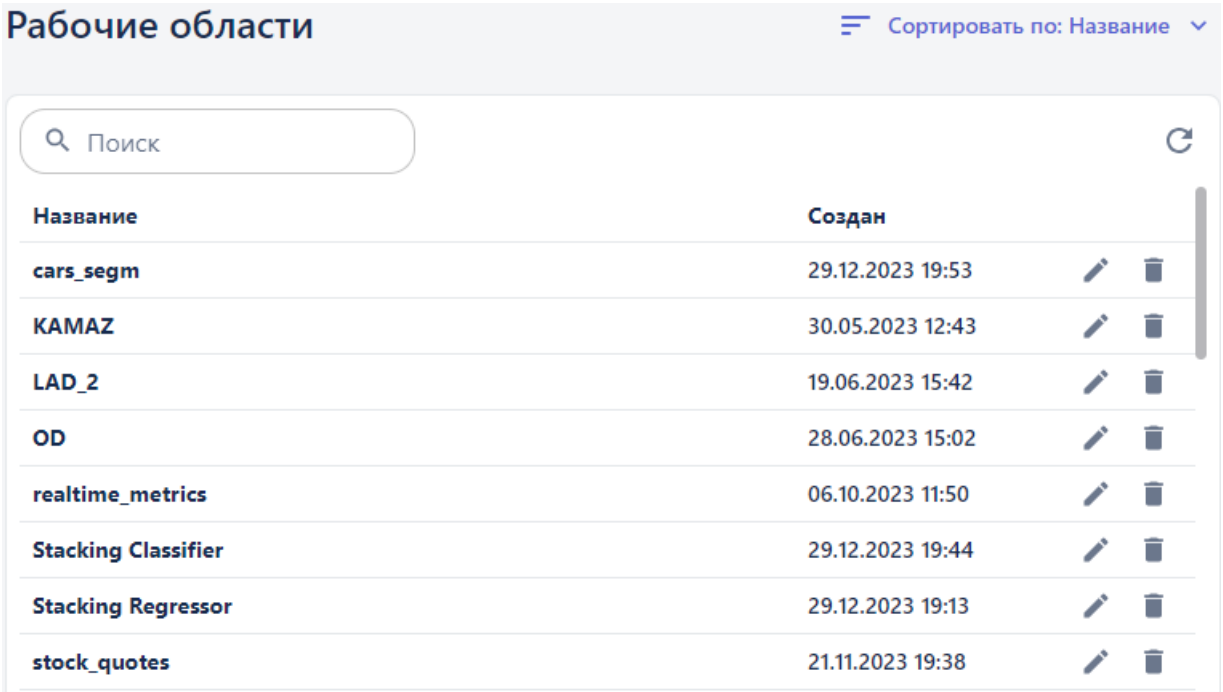


Рисунок 7.1.2 – Страница со списком созданных на Платформе рабочих областей

На странице рабочие области могут быть отфильтрованы по дате создания, либо по названию в алфавитном порядке.

Чтобы открыть сохраненную рабочую область кликните на её название. Чтобы удалить ненужную рабочую область - кликните кнопку удаления в правой части строки с названием

области, чтобы изменить название рабочей области - кликните кнопку редактирования, после нажатия откроется страница редактирования рабочей области.

7.2. Построение блок схемы

Предварительным условием для построения блок схемы является:

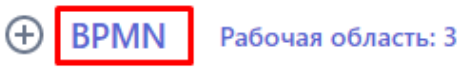
1) Загруженные в систему данные (файлы или датасеты)
2) Созданная рабочая область (на одной рабочей области можно создать неограниченное количество блок-схем.

Построение блок-схемы осуществляется путем добавления на рабочую область элементов (блоков) и соединение их между собой.

7.2.1 Блок «Запуск»

Блок «Запуск» обозначает начало блок-схемы, и всегда является её первым элементом. Так как на рабочей области может быть несколько блок-схем, именно по блоку «Запуск» определяется их количество, и идентифицируется принадлежность блоков к той или иной блок-схеме.

Для добавления на блок-схему элемента «Запуск» нажмите на кнопку «Добавить элемент» (кнопка BPMN) на панели инструментов:



Откроется меню выбора блоков – библиотека графических элементов нотации BPMN 2.0:

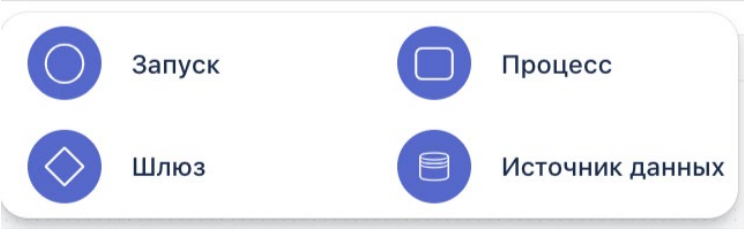
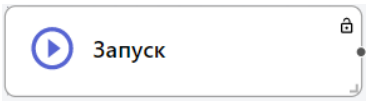


Рисунок 7.2.1 – Меню блоков

Выберите элемент «Запуск». Выбранный элемент будет добавлен на рабочую область:



Кнопка на элементе предназначена для запуска текущей блок-схемы после её сборки. У элемента «Запуск», как первого элемента блок-схемы, есть только одна точка выхода, предназначенная для соединения с последующими элементами блок-схемы.

При необходимости размер блока можно увеличить или уменьшить, потянув за уголок в правой нижней части элемента:

После того, как блок схема будет собрана и готова к запуску, нужно будет нажать на кнопку , после этого вид блока изменится и появится возможность создать отчет по результатам отработки:

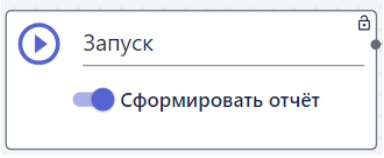


Рисунок 7.1.2 – Опция формирования отчета на блоке «Запуск»

7.2.2. Блок «Источник данных»

Следующим элементом в блок схеме после «Запуска» всегда является «Источник данных» - блок, который определяет какие данные будут использоваться в сценарии.

Для того чтобы добавить блок на рабочую область, откройте меню блоков и выберите элемент «Источник данных» (кнопка). Выбранный элемент появится на рабочей области конструктора. Вы можете передвинуть элемент на любую часть рабочей области, нажав на него. Для понятной визуализации процесса, рекомендуется расположить «Источник данных» правее элемента «Запуск».

Для объединения элементов в блок-схему, их требуется соединить между собой. Для этого нажмите на точку выхода блока, которая отображается в виде круглой точки на правой грани блока, и перетащите мышью появившуюся стрелку в сторону нужного блока. Пример показан на рисунке ниже:

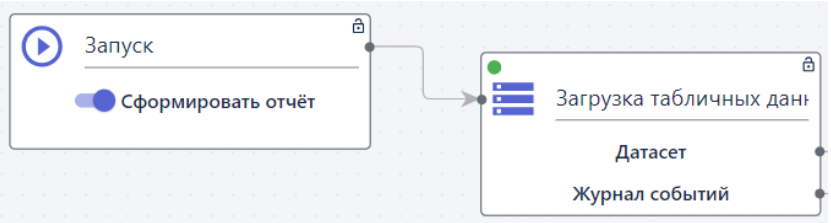


Рисунок 7.2.1 – Соединение блоков между собой

Соедините стрелкой элементы «Запуск» и «Источник данных» (далее соединение нового элемента с предыдущим является действием по умолчанию). Элемент «Источник данных» уже имеет несколько точек выхода, которые соединяются с одноименными точками последующих элементов. Для удаления соединения необходимо дважды кликнуть по линии соединения, чтобы ее выделить, и удалить. Пример приведен на рисунке ниже.

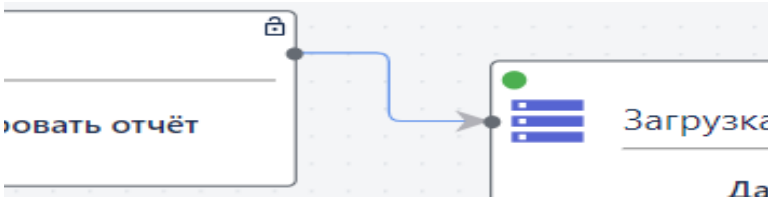


Рисунок 7.2.2 – Удаление соединения между блоками

На блоке «Источник данных» отображаются два компонента:

1) «Датасет» - это непосредственно сами данные.
2) «Журнал событий» содержит информацию обо всех преобразованиях с данными, которые выполняются в текущем блоке пайплайна. Ведение журнала позволяет сохранить историю преобразований над данными, и при необходимости выполнить обратное преобразование.

Чтобы открыть настройки элемента нажмите на значок (по умолчанию параметры элемента развернуты):

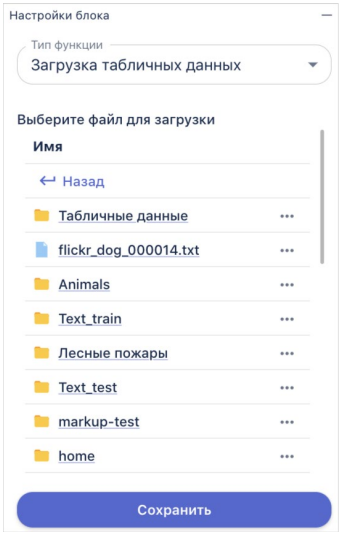


Рисунок 7.2.3 – Настройка параметров блока «Источник данных»

** Дальнейшие действия по настройке блоков указаны для примера. При работе с Системой пользователь должен выбрать необходимые параметры исходя из своей задачи и загруженных данных.*

После добавления на рабочую область, для элемента «Источник данных» по умолчанию выбрана функция: тип функции «Загрузка данных» -> функция «Загрузка табличных данных». Это можно увидеть в верхней части окна настройки параметров элемента:

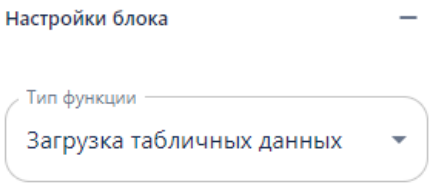


Рисунок 7.2.4 – Отображение типа функции в параметрах блока

Для того чтобы посмотреть другие доступные функции, нужно нажать на выпадающий список. Весь список доступных функций и их описания доступны в Таблице 18.1 – Перечень автоматизированных функций элемента «Источник данных».

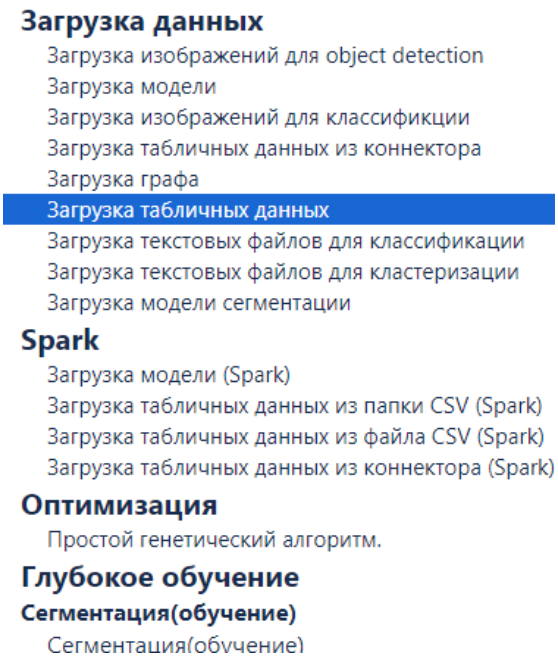


Рисунок 7.2.5 – Список возможных функций элемента «Источник данных»

В разделе «Выберите файл» отображается структура папок из разделе «Данные», чтобы выбрать файл достаточно перейти в нужную папку и кликнуть на три точки в правой части строки с названием файла и нажать «Выбрать»:

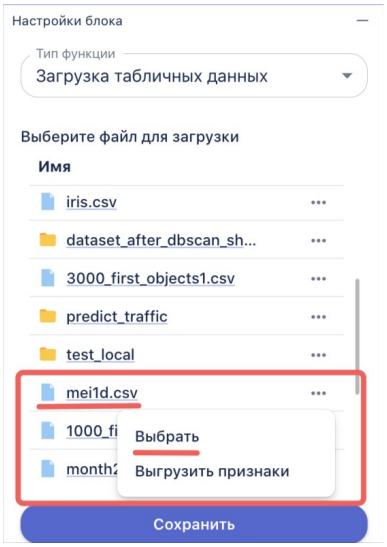


Рисунок 7.2.6 – Отображение папок и файлов из раздела «Данные» в параметрах блока «Источник данных»

После этого в нижней части окна отобразится его название:

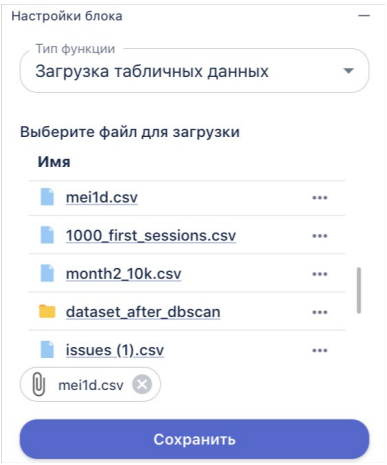


Рисунок 7.2.7 – Отображение выбранного файла, предназначенного для загрузки в блок-схему

Кнопка «Выгрузить признаки» используется для других блоков, где в настройках необходимо указать целевые признаки для конкретной функции.

Если кликнуть на название файла на рабочей области отобразится его предпросмотр:

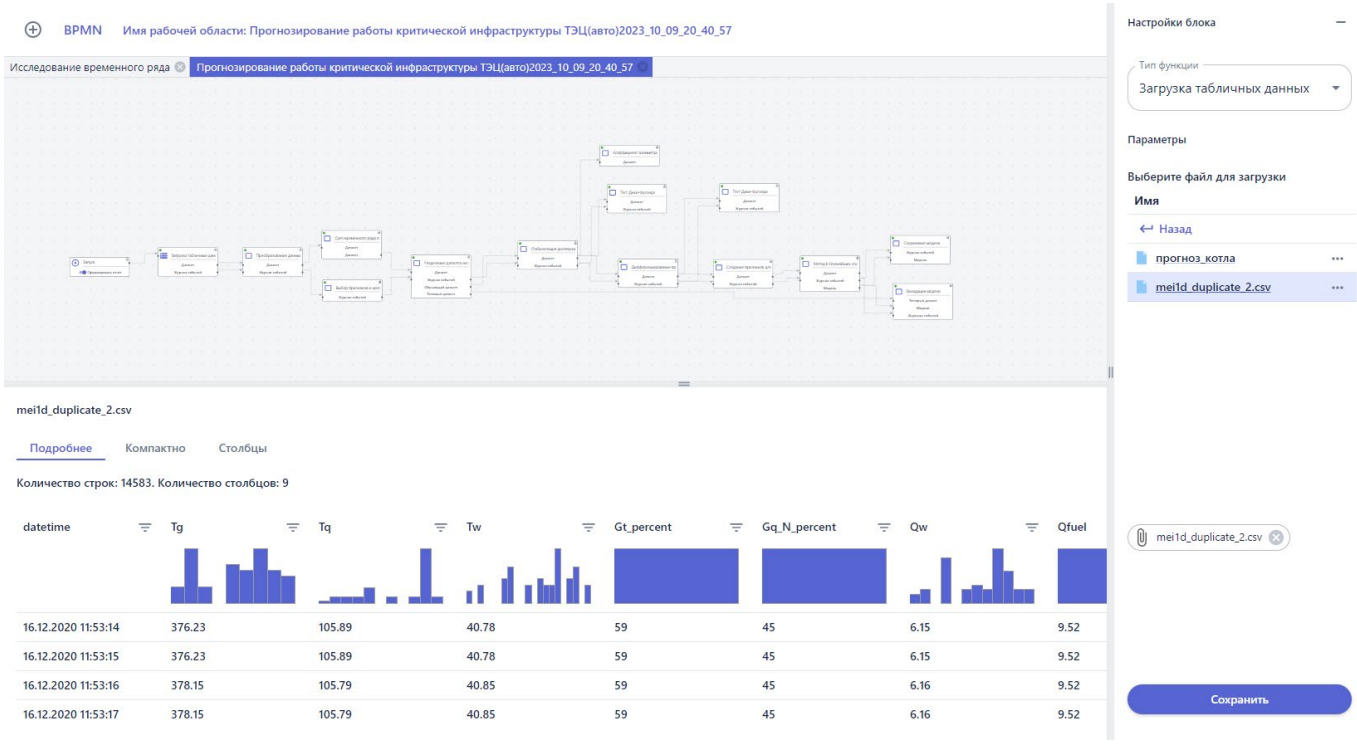


Рисунок 7.2.7.1 – Предпросмотр файла на рабочей области

Для сохранения выбранных настроек нажмите на панели параметров кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию). Для удаления неверно добавленного на рабочую область элемента предусмотрена кнопка «Удалить блок».

Любой блок можно переименовать, чтобы дать ему понятное название, отображающее суть происходящего процесса. Для этого дважды щелкните левой кнопкой мыши на текущее название элемента в рабочей области и измените его. Чтобы новое название сохранилось достаточно щелкнуть мышью в любом месте на рабочей области, исключая сам блок.

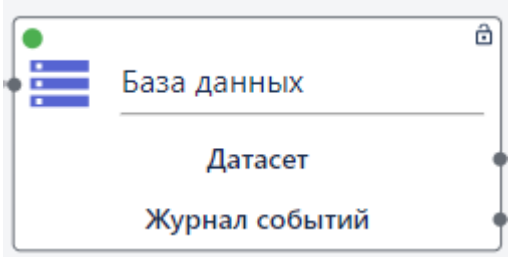


Рисунок 7.2.8 – Ввод названия элемента

Для того чтобы удалить блок, кликните по нему правой кнопкой мыши и нажмите «Удалить».

7.2.3. Блок «Процесс»

Блок «Процесс» предназначен для выполнения операций над данными. Блок-схема может содержать несколько элементов «Процесс», настроенных пользователем для выполнение определенных задач. Весь список доступных функций и их описания доступны в таблице 18.2 – Перечень автоматизированных функций элемента «Процесс».

Аналогично тому, как было описано в предыдущих пунктах, откройте меню блоков и выберите элемент «Процесс» ( ). Выбранный элемент появится на рабочей области.

Далее будет показан принцип настройки свойств блока на примере одной функции. Выбор функции определяется типом решаемой задачи.

Для примера выберите для элемента функцию: раздел «Машинное обучение» -> функция «Разделение датасета на обучающую и тестовую выборки»:

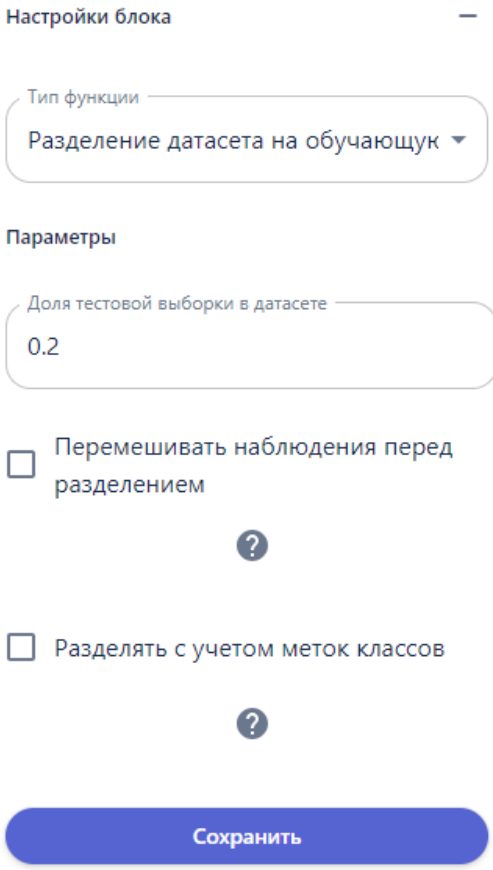


Рисунок 7.2.8 – Панель свойств блока «Процесс»

Далее осуществляется настройка параметров следующим образом:

- В разделе «Параметры» –> в поле «Доля тестовой выборки в датасете» введите значение 0.2;
- Оставьте пустым поле «Перемешивать наблюдения перед разделением». Рядом с полем есть подсказка, что не рекомендуется перемешивать наблюдения во временных рядах (выбирайте действие в зависимости от типа входных данных);
- Установите галочку в поле «Разделять с учетом меток классов» – применяется для задач классификации (выбирайте действие в зависимости от решаемой задачи).

Измените название элемента на «Сплит датасета»:

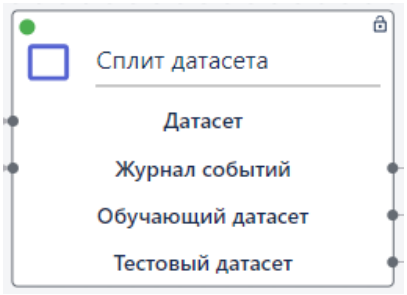


Рисунок 7.2.9 – Отображение блока на рабочей области после настройки его параметров и ввода названия

Обратите внимание, что соединение элемента «Процесс» с другими элементами блок-схемы выполняется только после настройки и сохранения его параметров. Это связано с тем, что каждая функция имеет свой набор компонентов, который отображается на элементе после сохранения его настроек.

Соединить элемент «Процесс» с предыдущими элементами блок-схемы можно следующим образом:

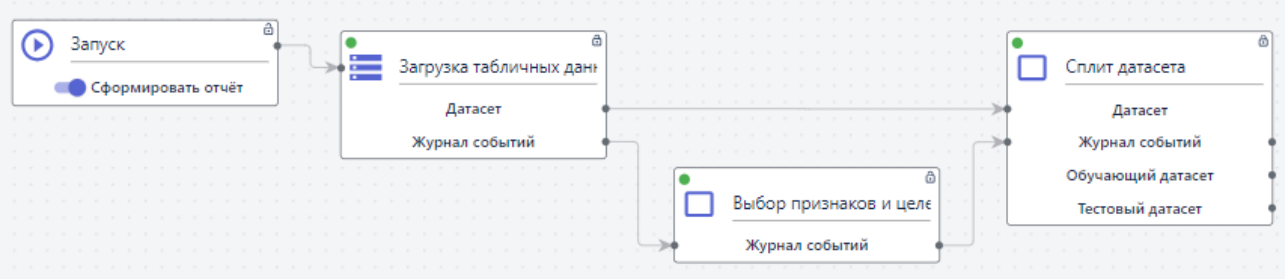


Рисунок 7.2.10 – Соединения между несколькими блоками (организация связей/пробрасывание данных)

Обратите внимание, что соединять можно только идентичные (одноименные) компоненты блоков. Больше примеров построения блок схем, сценарии сохранения моделей, отображения на рабочих областях таблиц, графиков и изображений можно прочитать в разделе Примеры работы с Платформой.

7.3. Запуск блок-схемы на рабочей области

Чтобы запустить блок схему нужно нажать на кнопку на элементе «Запуск». В результате элементы пайплайна начинают последовательно запускаться. При обработке блока он загорается оранжевым цветом, а после успешного завершения – зеленым:

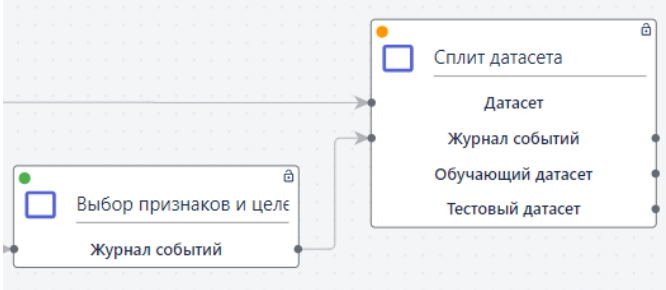


Рисунок 7.3.1 – Последовательная обработка блоков пайплайна

В случае, если блок не отработал (например, вследствие того, что был неправильно настроен или из-за неверных входных данных), на нем появится индикатор красного цвета. Такой блок необходимо проверить, изменить настройки и запустить заново:

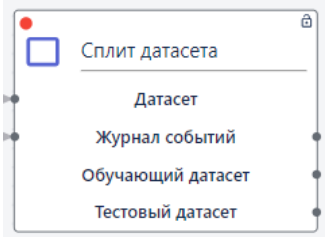


Рисунок 7.3.2 – Индикатор ошибки при обработке блока

8. Сохранение модели ИИ

Для сохранения *модели ИИ*, обученной выполнению какой-либо задачи, на блок-схему необходимо добавить специальный блок. Этот блок настраивается для элемента «Процесс» с помощью функции «Управление моделями» → функция «Сохранение модели»:

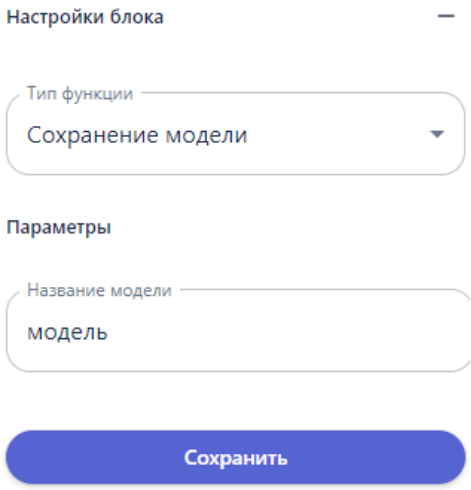


Рисунок 7.3.1 – Настройка блока для сохранения обученной модели ИИ

После успешной отработки блок схемы, которая содержит такой блок, название модели будет добавлено в список сохраненных моделей в меню «Модели»:

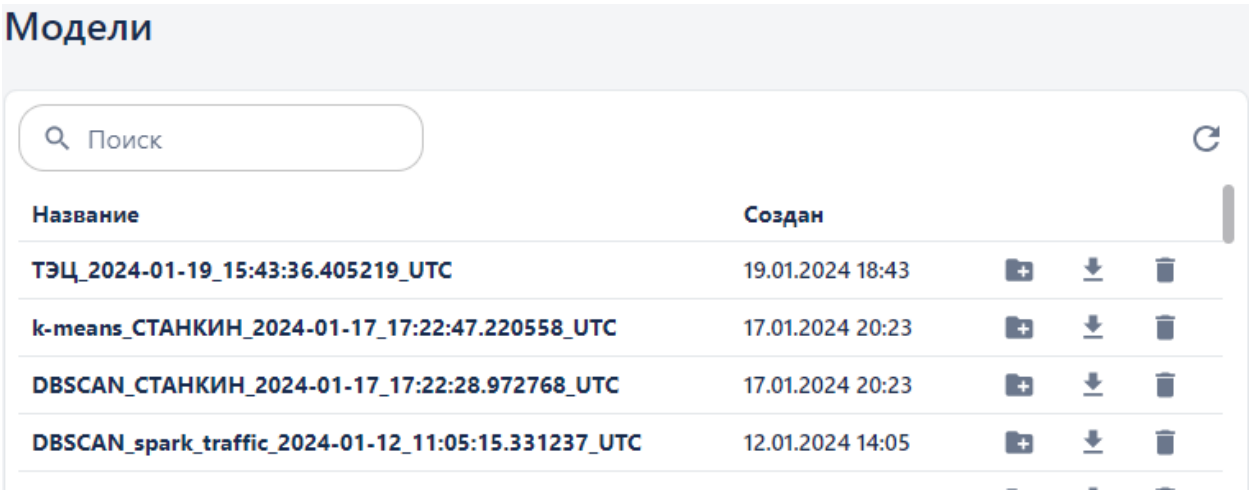


Рисунок 7.3.2 – Вкладка меню «Модели»

Обратите внимание, что модель будет сохраняться столько раз, сколько будет запущена блок схема с элементом «Сохранение модели», при этом в разделе будет меняться временная отметка создания записи.

Модель можно использовать следующими способами:

1. Создать на её основе приложение, которое предназначено для интеграции с внешними системами. Для этого нажмите на значок «Новое приложение» в строке с названием модели. Новое приложение появится в соответствующем разделе системы.
2. Скачать. Для этого нажмите на значок «Скачать» в строке с названием модели. После этого на ваш компьютер будет сохранен архив с тремя файлами:
    - **model.pkl** - сама модель.

- **vars_dict.pkl** - словарь преобразований. Преобразования необходимо сохранять, чтобы при анализе новой порции данных над ними выполнялись все те же преобразования, что и при обучении модели.
- **info.json** - служебный файл, куда прописывается тип модели.

3. Использовать при построении новой блок схемы в качестве источника данных (например, для целей прогнозирования). Пример можно посмотреть в разделе 14.4 Работа с данными в режиме реального времени.
4. Создать коннектора с обученной моделью (например, для распознавания объектов на видео или изображениях). Это позволит проверить обучение модели на новой порции данных. Подробней описано в разделе 14.2.1 Проверка обученной модели на локальных данных.

9. Графическое представление информации на рабочей области

После *сборки* и *успешного* запуска блок-схемы на рабочей области есть возможность посмотреть результаты обучения созданной модели в виде графиков, таблиц и изображений, которые можно вывести прямо на рабочую область.

> ***Примечание**: под сборкой имеется в виду, что на рабочую область добавлены все элементы блок-схемы, и они последовательно соединены между собой. А успешным считается запуск блок-схемы, когда все ее элементы отработали с «зеленым» индикатором. Если же после запуска блок-схемы его отработка останавливается на одном из элементов, и на этом элементе горит «оранжевый» индикатор, запуск считается неуспешным.*

После успешного запуска в верхней части панели инструментов рабочей области станут активными следующие кнопки: , где – графики, – таблицы, – изображения, – описание модели.

Вы можете нажать на те кнопки визуализации, которые подсвечиваются фиолетовым цветом. При этом голубым подсвечиваются только те иконки, которые актуальны для запущенной схемы, т.к. блоки могут содержать разные функции, имеющие разное графическое представление.

9.1. Графики

Чтобы отобразить результаты работы блок схемы в виде графиков необходимо нажать на кнопку .

Полный список доступных в Системе графиков с объяснением интерпретации результатов доступен в Базе знаний. За каждым типом блока закреплен определенный набор графиков, например, для блока Анализ данных -> Анализ временных данных доступны следующие графики: Линейный график, ACF/PACF, Декомпозиция, Свечной график, Time profile, Extended, Bollinger Bands Stochastic Oscillator.

Для того чтобы добавить график на рабочую область из выпадающего списка выберите нужное название. Например, «Time profile временного ряда» в анализе временных рядов:

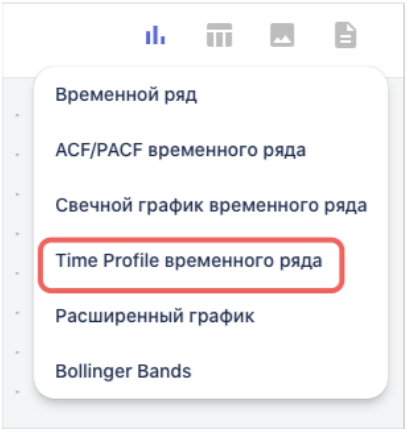


Рисунок 9.1.1 – Список доступных графиков для анализа временных рядов

В результате на рабочую область будет добавлен график:

Рисунок 9.1.2 – Time profile временного ряда на рабочей области

Для ряда графиков доступен выбор из выпадающего списка признака, для которого составляется визуализация. Например, можно два раза выбрать график Time profile и для одного указать признак Tq, а для другого Tw и сопоставлять их значения одновременно на рабочей области:

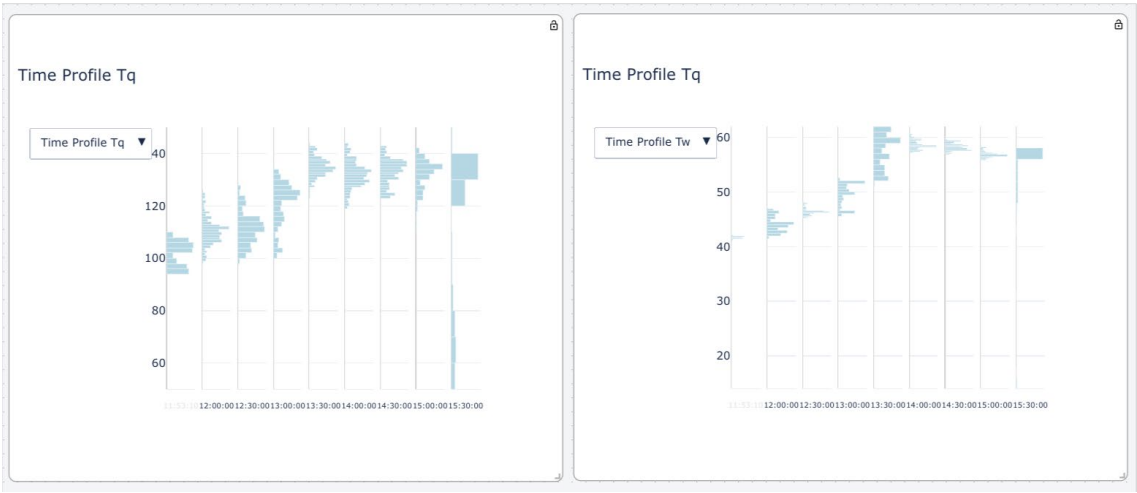


Рисунок 9.1.3 – Отображение одного типа графика для разных признаков

Также в правом углу рамки графической визуализации, при наведении на неё курсора, отображаются следующие кнопки:

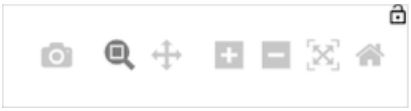


Рисунок 9.1.4 – Кнопки для работы с визуализацией

В программе также реализована возможность создания графиков с данными, получаемыми в режиме реального времени. Для таких графиков существует отдельный блок, находящийся в разделе «Анализ данных» -> «Визуализация Real Time». Главное отличие от стандартного блока «Визуализация» в том, что для каждого графика необходимо задавать число периодов в окне и период окна - параметры, которые определяют интервал, который будет отображаться на графике в рабочей области.

9.2. Таблицы

Чтобы отобразить результаты работы модели в виде таблиц нажмите кнопку .
Например:

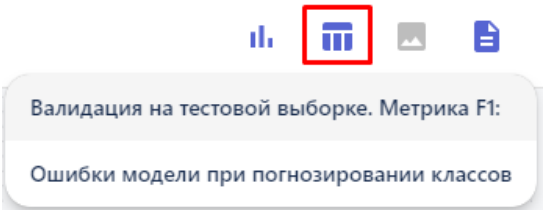


Рисунок 9.2.1 – Список таблиц для визуализации

Чтобы отобразить таблицу на рабочей области, нажмите на её название в выпадающем списке и она появится на экране, например:

Рисунок 9.2.2 – Блок визуализации «Ошибки модели при прогнозировании классов» (показан пример для сценария 14.1 «Прогнозирование лесных пожаров»)

9.3. Изображения

Чтобы отобразить результаты работы модели в виде таблиц нажмите кнопку .
Например:

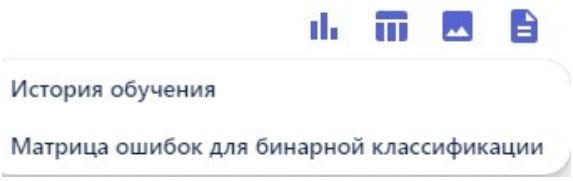


Рисунок 9.3.1 – Список графиков для визуализации

Далее выберите изображение из списка и визуализация появится на рабочей области:

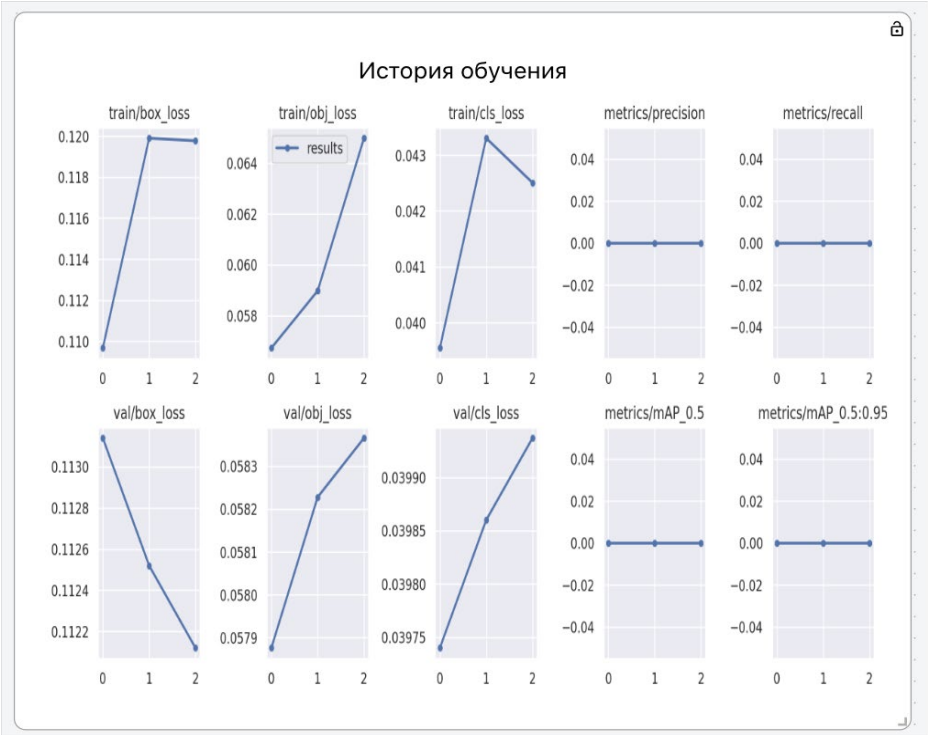


Рисунок 9.3.2 – Блок визуализации «История обучения (показан пример для сценария 14.2 «Обучение модели ИИ распознаванию объектов на изображениях»)

9.4. Описание модели

Чтобы посмотреть описание модели нажмите кнопку .

Описание варьируется в зависимости от функций, которые были применены в модели. Например, для блок-схемы, где присутствовали элементы «Стандартизация», «Валидация» и «XGB классификации» описание будет выглядеть следующим образом:

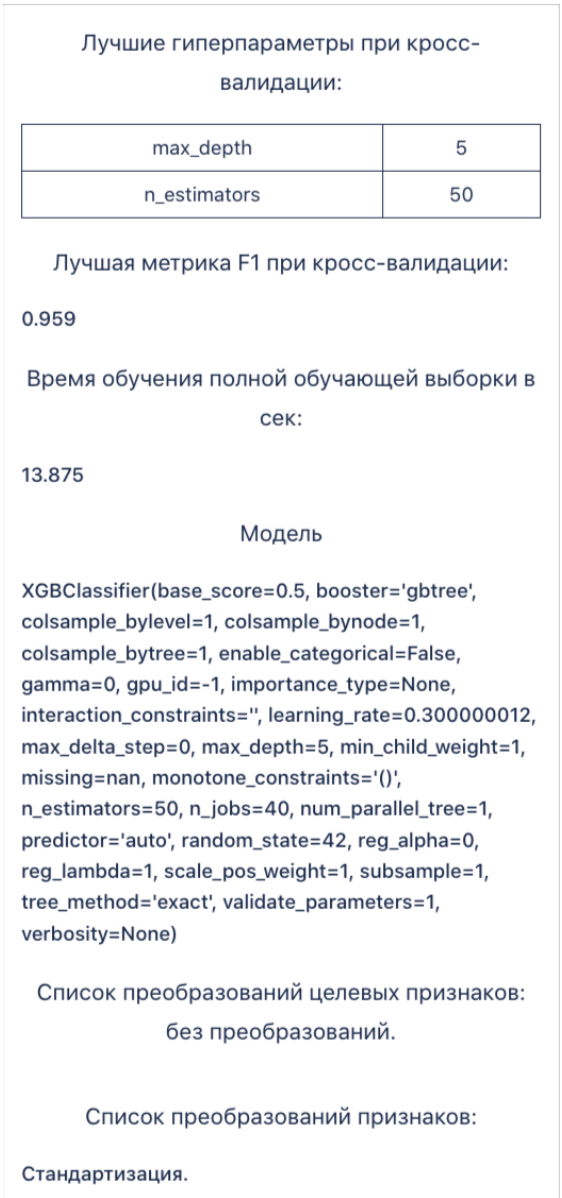


Рисунок 9.4.1 – Вариант описания модели

Если на рабочей области размещены несколько блок схем, при нажатии на описание вы увидите информацию по каждой из них.

10. Работа с Дашбордами. Раздел «Визуализация»

Дашборд – это интерактивная рабочая область, которая наглядно представляет, визуализирует, объясняет и анализирует данные. На рабочую область пользователь может добавлять графики, таблицы, диаграммамы, визуализацию пайплайнов для последующей работы с ними.

Работа с дашбордами осуществляется в разделе «Визуализация» → «Дашборды». Для создания нового дашборда на панели инструментов нажмите кнопку «⊕». В открывшемся окне введите название создаваемого дашборда, например «Таблица», и нажмите кнопку «Создать»:

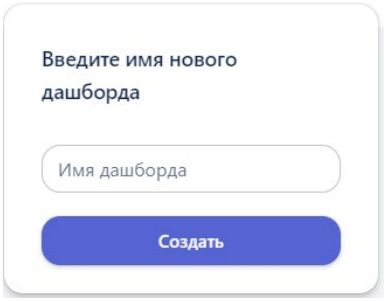


Рисунок 10.1 – Создание нового дашборда

В результате отобразится название дашборда (это текущий дашборд, с которым работает пользователь):

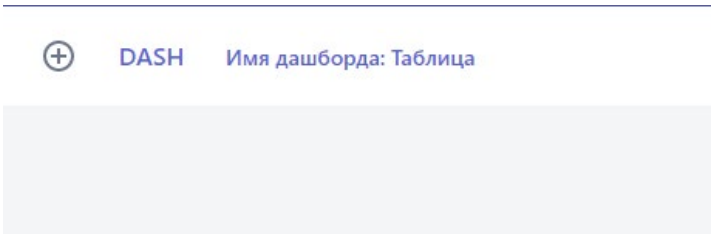


Рисунок 10.2 – Отображение наименования дашборда

Новый дашборд создается с пустой рабочей областью. Чтобы его наполнить добавляются интерактивные блоки. Для этого нажмите кнопку DASH и выберите нужный тип интерактивного блока:

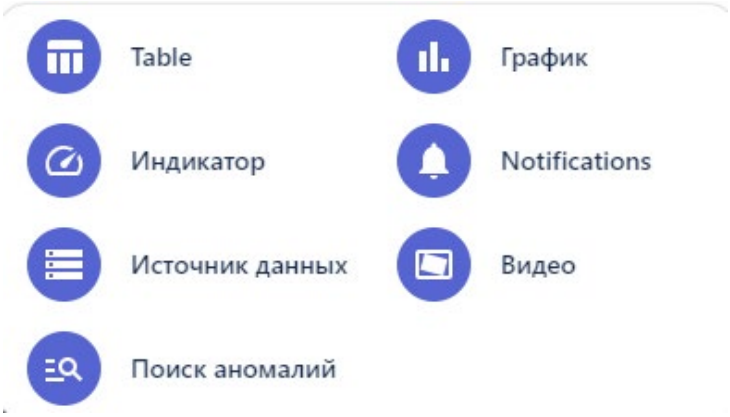


Рисунок 10.3. – Добавление интерактивного блока

В текущей версии Системы реализована работа с блоками: «Таблица», «График», «Видео» и «Поиск аномалий». Предварительным условием для добавления любого из типов блоков является создание коннектора в разделе «Соединения», в котором настроено получение данных. При этом

это могут быть как данные, получаемые из внешних источников, так и сгенерированные внутри Системы. Подробней о создании коннекторов вы можете посмотреть в разделе **Работа со всеми типами коннекторов**.

10.1. Таблица

Интерактивный блок «Таблица» предназначен для:

- Отображения подключения к внешним базам данных, в виде табличных данных, обновляющихся в режиме реального времени. В таком случае настраивается подключение к коннекторам с названиями типов баз данных (clickhouse, postgresql, mongo);
- Записи и сохранения в Системе информации, получаемой из внешних баз данных. Используется коннектор с типом «save_table»;
- Прогнозирования целевых событий, когда данные для анализа поступают из внешних баз данных. Используется коннектор с типом «table_app»;
- Отображение таблиц, полученных в результате отработки блок-схем, которые содержат блоки, имеющие в качестве выходной информации визуализации в виде таблиц. Используется коннектор с типом «constructor». Такой тип коннектора создается автоматически, после успешного запуска блок-схемы.

Например, чтобы настроить дашборд с подключением к коннектору с типом «clickhouse»:

1. Создайте новый дашборд.
2. На дашборд добавьте интерактивный блок с типом «Таблица».
3. Чтобы подгрузить данные в таблицу в правом верхнем углу нажмите кнопку . Отобразится список коннекторов, созданных в Системе:

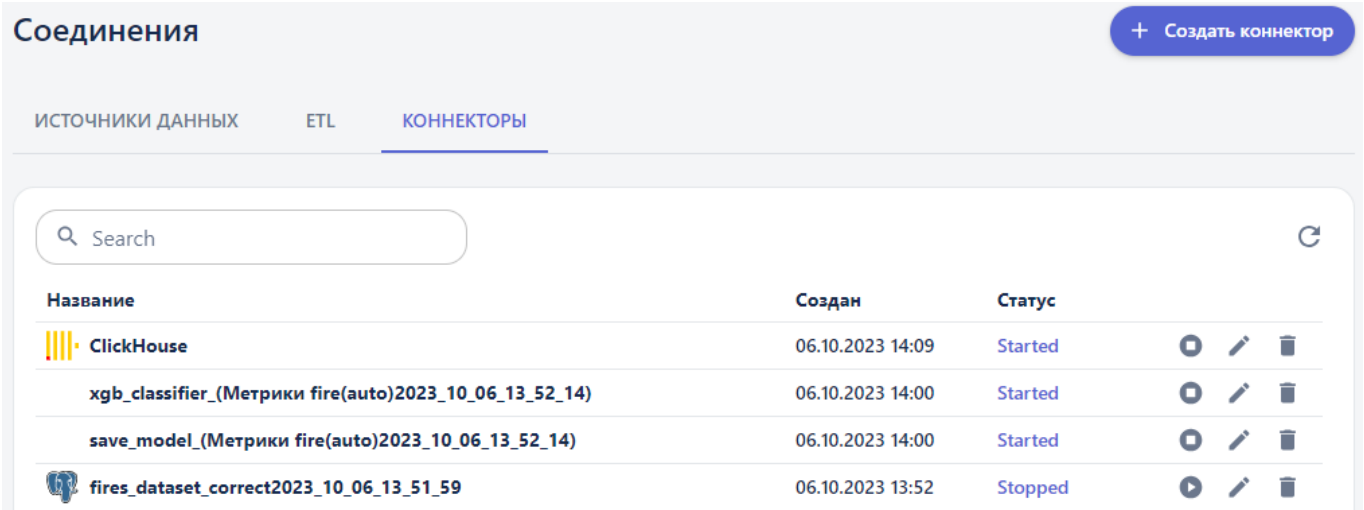
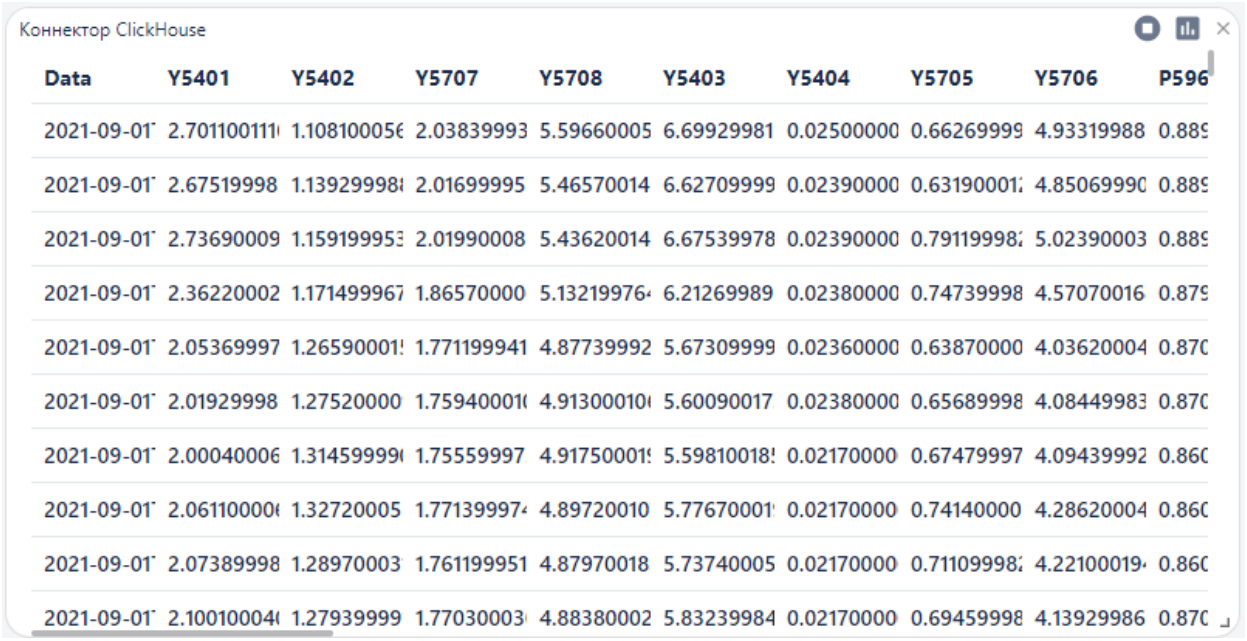


Рисунок 10.4. – Список коннекторов

В открывшемся окне для коннекторов отображаются их состояния (Запущен (Started) / Остановлен (Stopped)). Если статус коннектора «остановлен», данные из внешних источников не поступают в Систему. Из этого окна можно запустить/остановить коннектор, выполнить его редактирование при необходимости, или удалить его.

4. Запустите коннектор, нажатием кнопки в строке с коннектором (по умолчанию, коннекторы с типом «clickhouse» создаются в статусе «Stopped»).
5. Выбрать коннектор из списка нажатием на него левой кнопкой мыши.

6. На дашборде отобразится результат подключения к БД «ClickHouse» в виде таблицы с данными:



Data	Y5401	Y5402	Y5707	Y5708	Y5403	Y5404	Y5705	Y5706	P596
2021-09-01	2.701100111	1.108100056	2.03839993	5.59660005	6.69929981	0.02500000	0.66269999	4.93319988	0.889
2021-09-01	2.67519998	1.139299981	2.01699995	5.46570014	6.62709999	0.02390000	0.63190001	4.85069990	0.889
2021-09-01	2.73690009	1.159199953	2.01990008	5.43620014	6.67539978	0.02390000	0.79119998	5.02390003	0.889
2021-09-01	2.36220002	1.171499967	1.86570000	5.13219976	6.21269989	0.02380000	0.74739998	4.57070016	0.879
2021-09-01	2.05369997	1.265900015	1.771199941	4.87739992	5.67309999	0.02360000	0.63870000	4.03620004	0.870
2021-09-01	2.01929998	1.27520000	1.759400016	4.91300010	5.60090017	0.02380000	0.65689998	4.08449983	0.870
2021-09-01	2.00040006	1.314599990	1.75559997	4.91750015	5.59810018	0.02170000	0.67479997	4.09439992	0.860
2021-09-01	2.06110000	1.32720005	1.771399974	4.89720010	5.77670001	0.02170000	0.74140000	4.28620004	0.860
2021-09-01	2.07389998	1.28970003	1.761199951	4.87970018	5.73740005	0.02170000	0.71109998	4.22100019	0.860
2021-09-01	2.10010004	1.27939999	1.77030003	4.88380002	5.83239984	0.02170000	0.69459998	4.13929986	0.870

Рисунок 10.5 – Отображение табличных данных из коннектора clickhouse

Коннектор можно остановить или запустить прямо на дашборде, для этого нажмите кнопку «остановить»/«запустить». Для того чтобы удалить дашборд - нажмите на крестик в правом верхнем углу блока.

10.2. Видео

Интерактивный блок «Видео» предназначен для:

- Отображения видеопотока данных в режиме реального времени, с удаленной камеры видеонаблюдения;
- Записи и сохранения полученного видеопотока;
- Классификации изображений с локального компьютера.

Рассмотрим настройку блока на примере подключения к коннектору с типом «video_stream».

1. Создайте новый дашборд, например, «Видеопоток».
2. Добавьте интерактивный блок «Видео» на дашборд.
3. Выберите для блока предварительно созданный коннектор с типом «video_stream».
4. Для начала получения данных с коннектора запустите его нажатием кнопки (при выборе коннектора для блока, или на самом блоке).

Ниже представлены примеры подключения к камерам с трансляцией видео потока с улицы и из магазина:

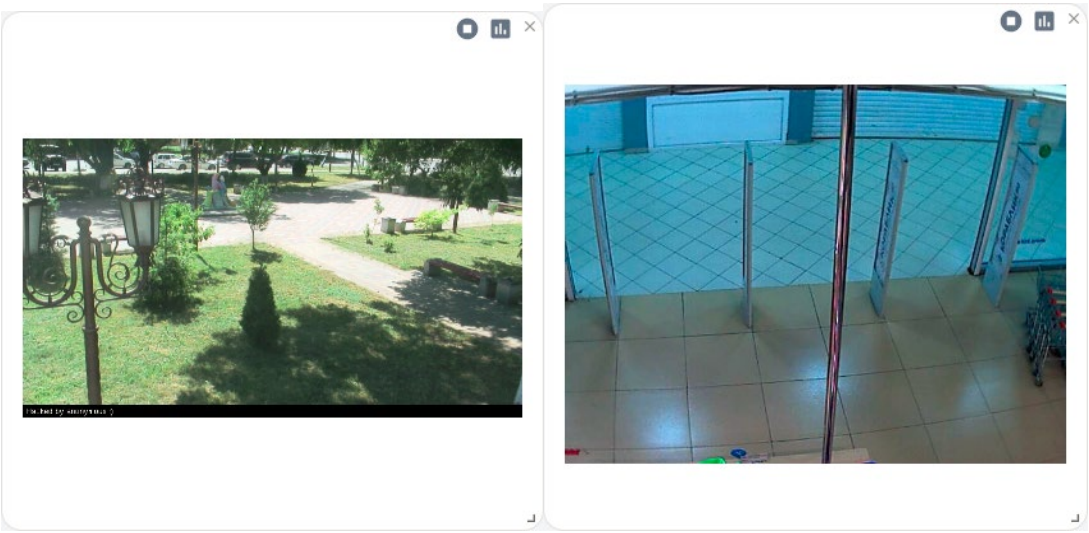
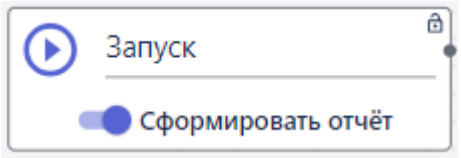


Рисунок 10.6 – Дашборды в видео потоком

Чтобы остановить визуализацию используется кнопка . При этом если в коннекторе настроено сохранение видеопотока, после нажатия этой кнопки файл с видео с камеры сохранится в разделе «Данные» в папке «video».

11. Создание отчета с результатами анализа данных

Более подробную информацию с результатами обучения модели можно просмотреть в отчете, который создается также после сборки блок-схемы. Для формирования отчета на первом блоке необходимо перевести бегунок вправо:



В названии отчета будет указано название блок-схемы, по которой формируется отчет и временная отметка его формирования. Для просмотра сформированного отчета после запуска блок-схемы нужно перейти в пункт меню «Отчеты» и выбрать из списка отчет, нажав на его название:

Рисунок 11.1 – Список автоматически сформированных отчетов

После нажатия на название отчета открывается отдельная вкладка с отчетом. В отчете кроме результатов обучения созданной модели (см. «Визуализацию») отображаются также: входные данные, отдельно выборки – обучающая и тестовая, датасет после стандартизации признаков, и т.д. Состав отчета отличается в зависимости от метода, который использовался в решении задачи ИИ.

12. Конвейер приложений

Конвейер приложений позволяет создать приложение на основе обученной модели. В таком приложении заложен шаблон, умеющий предсказывать наступление интересующих событий. Приложение можно развернуть отдельно за пределами системы, интегрировать с внешними системами, настроить получение входных данных и выполнять прогнозы.

Для того чтобы сформировать приложение необходимо перейти в раздел “Модели”:

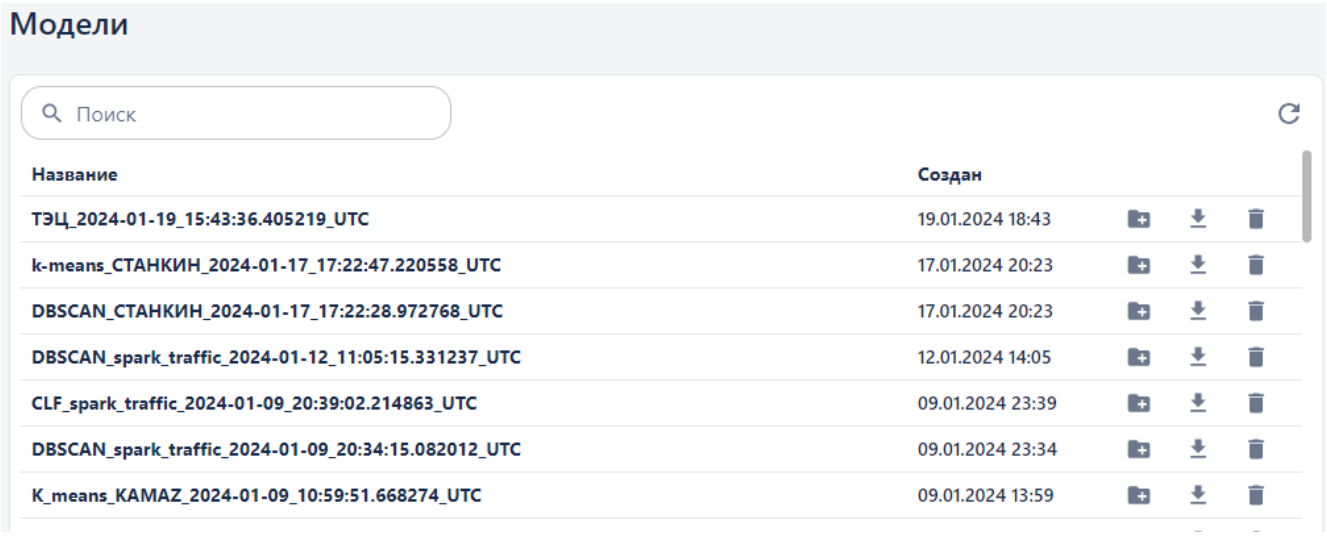


Рисунок 12.1 – Список сохраненных моделей

Для формирования приложения нажать рядом с названием обученной модели, после этого оно появится в разделе «Приложения» в папке apps:

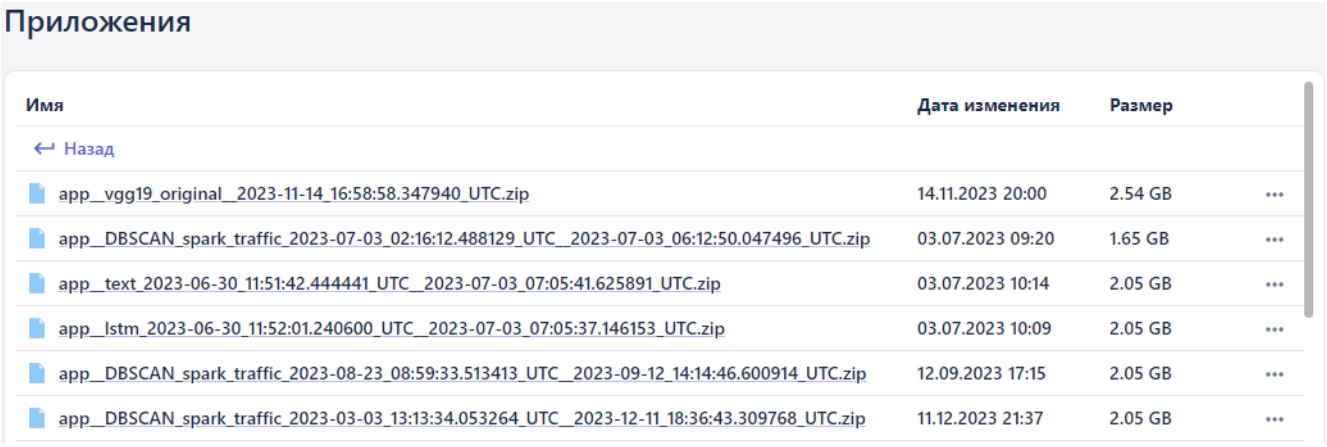


Рисунок 12.2 – Страница «Приложения»

Приложение упаковано в docker-контейнер и доступно для скачивания. Для того чтобы скачать приложение - нажмите на три точки в строке с названием приложения и скачайте его.

Комплектность приложения после создания и скачивании:

- **app.py** - файл приложения, которое принимает данные, вычисляет и возвращает результат прогнозирования на основе обученной модели
- **const.py** - переменные в приложении.
- **methods.py** - методы, используемые в модели
- **model.py** - загружает модель из файлов model.pkl, model_vars_dict.pkl, в которых хранится модель и параметры.

- **preprocess_and_predict.py** - подготовка и вычисление переменных для прогнозирования.
- **run.sh** - командный интерпретатор для Linux.
- **run.bat** - командный интерпретатор для Windows.
- **requirements.txt** - зависимости для сборки приложения.
- **dockerfile** - файл конфигурации, в котором расписано пошаговое создание среды для работы приложения.
- **docker-compose.yml** - файл с командами для запуска среды приложения.

Приложение позволяет решать задачи предиктивной аналитики для новых данных с использованием обученной модели. Предназначено для использования в сторонних системах.

13. Работа с проектом

Сущность «Проект» реализована с целью объединить *группу пользователей* для работы над одним проектом. При этом проект может объединять в себе такие сущности как: «модель», «рабочая область», «дашборд», «отчет», «файл», «коннектор». О том, как добавить каждую из этих сущностей в проект, написано в рамках данного раздела.

13.1. Создание нового проекта

1. Перейдите в пункт меню «Проекты»:

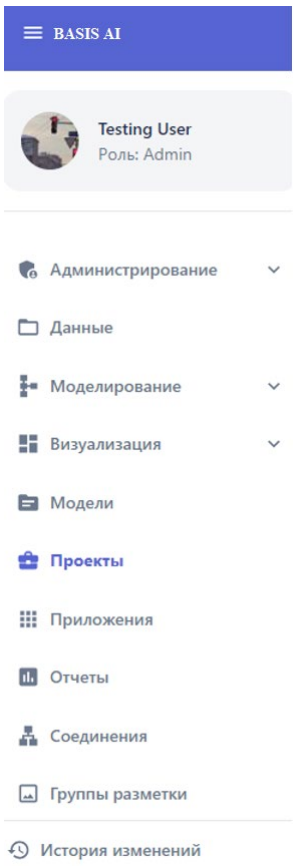


Рисунок 13.1 – Пункт меню «Проекты»

2. Откроется страница «Проекты», на которой отображаются все *проекты*, созданные *пользователем*:

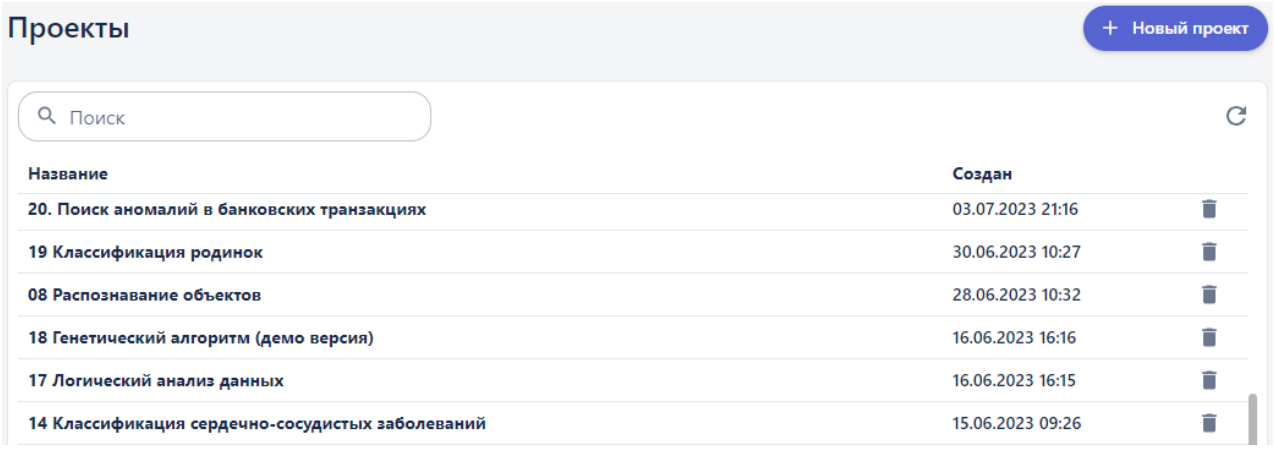


Рисунок 13.2 – Страница с проектами, доступными пользователю

3. Нажмите на кнопку «Новый проект», откроется окно создания нового проекта:

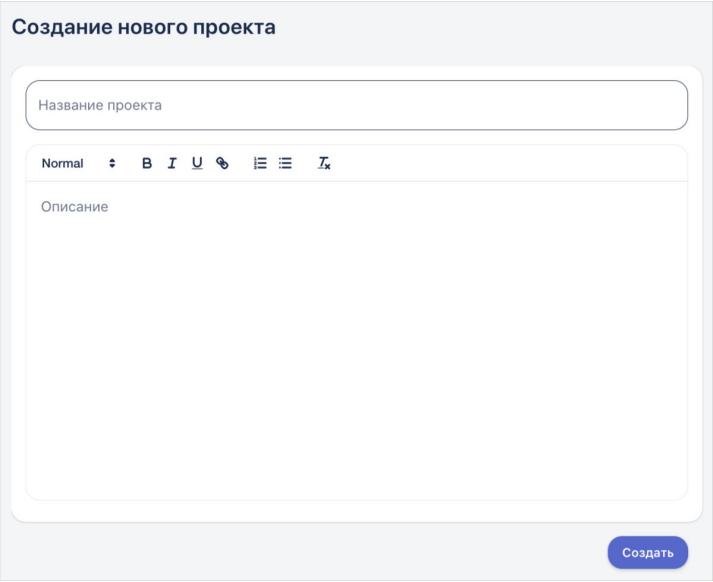


Рисунок 13.3 – Создание проекта

4. Задайте название проекта, например, «Проект тест» (обязательное поле)
5. Задайте описание проекта (необязательное поле), вы можете использовать средства форматирования текста:
    - Заголовки и подзаголовки
    - Жирный текст
    - Курсив
    - Подчеркивание
    - Вставка ссылки
    - Список
    - Нумерованный список
    - Кнопка очистки форматирования
6. Нажмите кнопку «Создать».
7. На страницу «Проекты» добавится новый проект

13.2. Редактирование проекта

Пользователь имеет возможность отредактировать название проекта и его описание:

1. На странице «Проекты» перейдите в проект, который требует редактирования
2. В открывшемся окне в верхнем правом углу нажмите кнопку «Редактировать»:

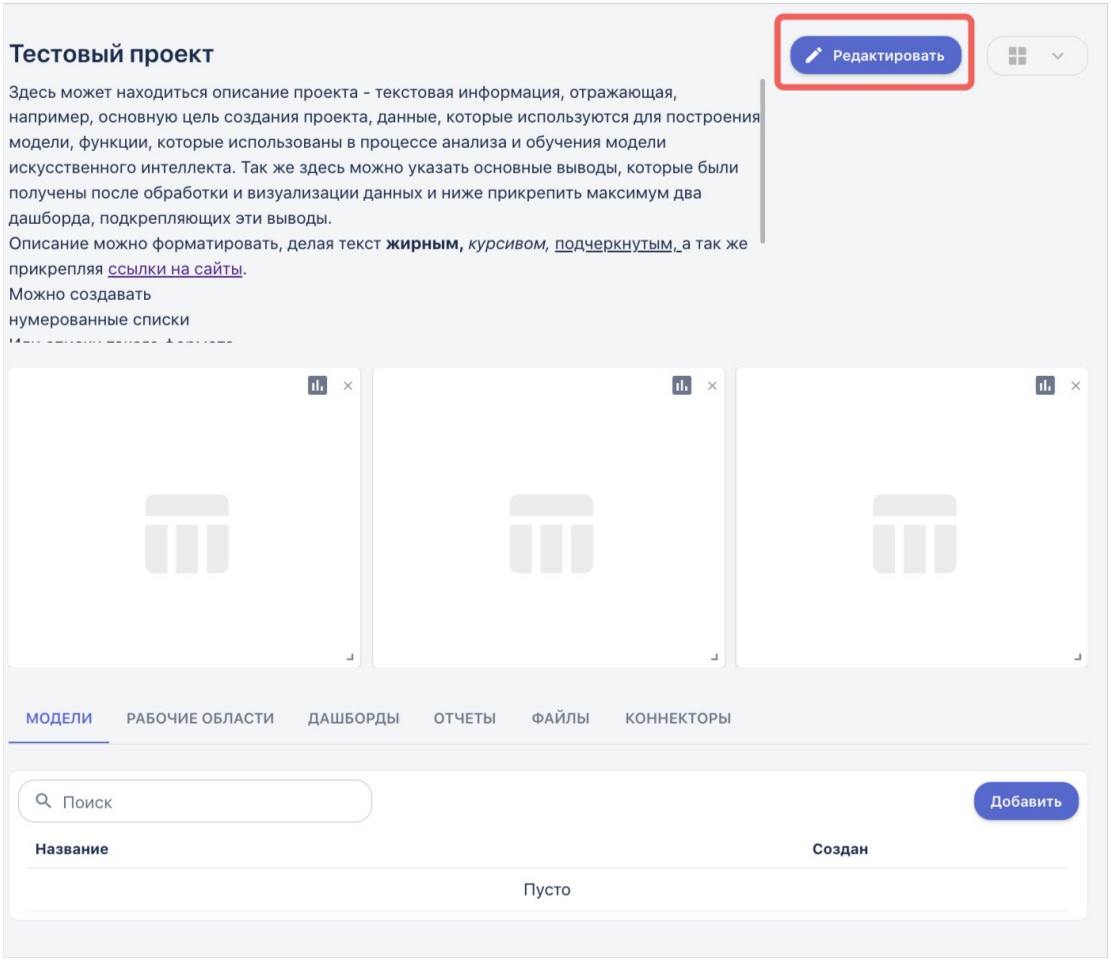


Рисунок 13.4 – Переход к редактированию существующего проекта

3. Задайте новое название проекта и его описание, затем нажмите кнопку «Сохранить»

4. Система вернется на страницу со списком проектов, для того чтобы посмотреть обновленное описание, перейдите в проект, нажав на его наименование.

13.3. Наполнение проекта

После того, как проект создан, его можно наполнить *содержимым* – теми сущностями, над которыми предстоит совместно работать группе пользователей. Для этого:

1. На странице «Проекты» перейдите по ссылке с названием созданного проекта, кликнув на его название:

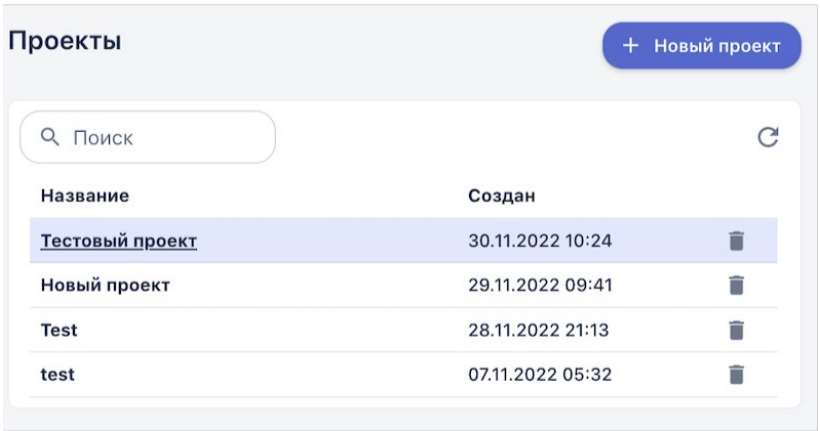


Рисунок 13.5 – Переход на страницу проекта для его просмотра и редактирования

2. Откроется *страница проекта* на первой вкладке «Модели»:

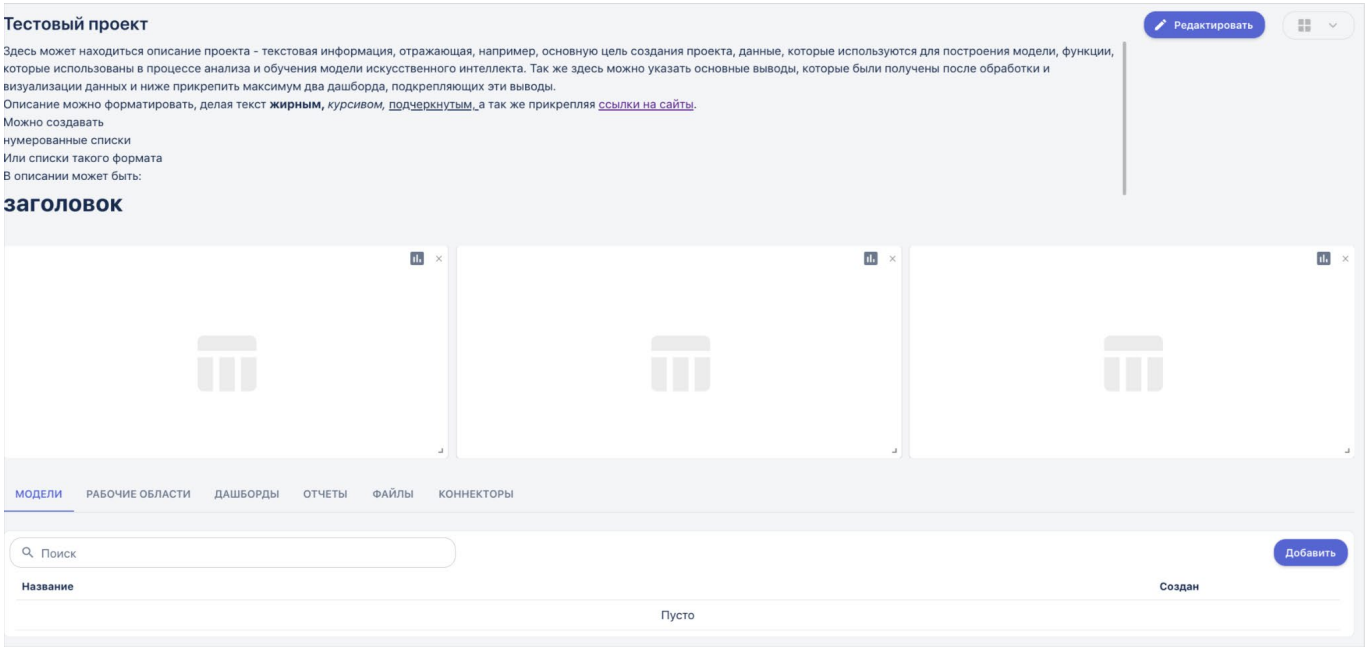
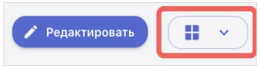


Рисунок 13.6 – Страница проекта

Целиком страница проекта состоит из вкладок с одноименными сущностями: модели, рабочие области, дашборды, отчеты, файлы, коннекторы.

3. Работа с дашбордами. Под описанием по умолчанию находятся три пустых дашборда. Вы можете сделать следующее в данном разделе:

a. Удалить ненужные дашборды, для этого нажмите “x” в правом верхнем углу дашборда.

   Обратите внимание! Вы можете добавить максимум три дашборда в данный раздел проекта. После удаления хотя бы одного Дашборда активируется кнопка добавления новой визуализации: 

b. Добавить новый дашборд. Для этого нажмите на кнопку «Добавить дашборд» и в выпадающем списке выберите нужный тип визуализации:

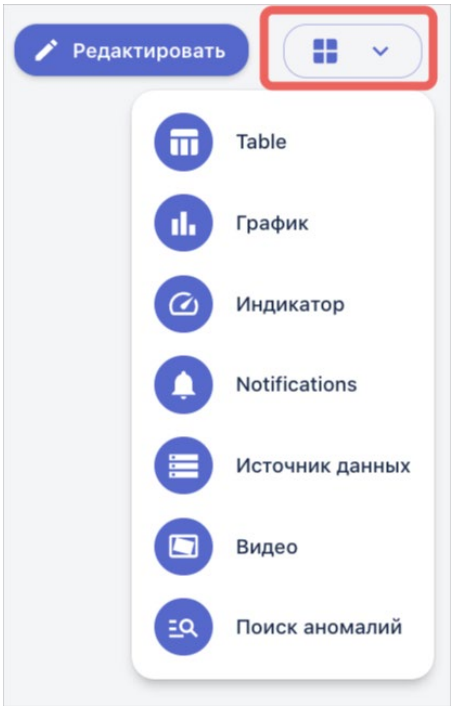


Рисунок 13.7 – Выбор типа визуализации

После этого под описание проекта добавится новый пустой дашборд выбранного типа

c. Выбрать коннектор на отображения информации на дашборде. Для этого нажмите на кнопку коннектор в правом углу дашборда ( ). Отобразится список всех созданных коннекторов в системе соответствующего типа. Выберите необходимый коннектор из списка, кликнув на его наименование:

Рисунок 13.8 – Выбор коннектора

После этого визуализация отобразится на дашборде.

4. **Добавление сущности «Модель»**. На вкладке «Модели» пользователь нажимает кнопку «Добавить», и открывается выпадающий список со всеми моделями ИИ, созданными в Системе. Выбирается модель и добавляется в проект.

Модель доступна для скачивания по кнопке :

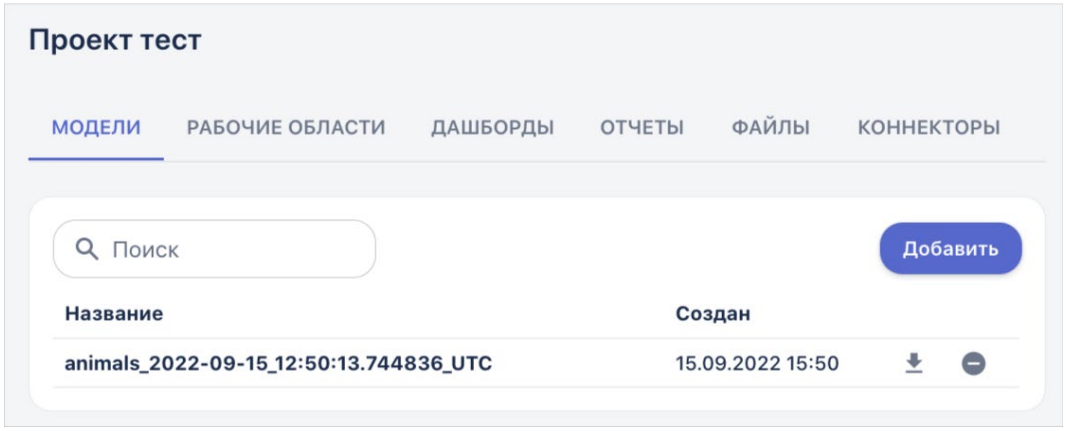


Рисунок 13.9 – Загруженная в проект модель

5. **Добавление сущности «Рабочая область»**. Перейдите на вкладку «Рабочие области» и нажмите кнопку «Добавить». Из выпадающего списка выберите рабочую область для добавления в проект:

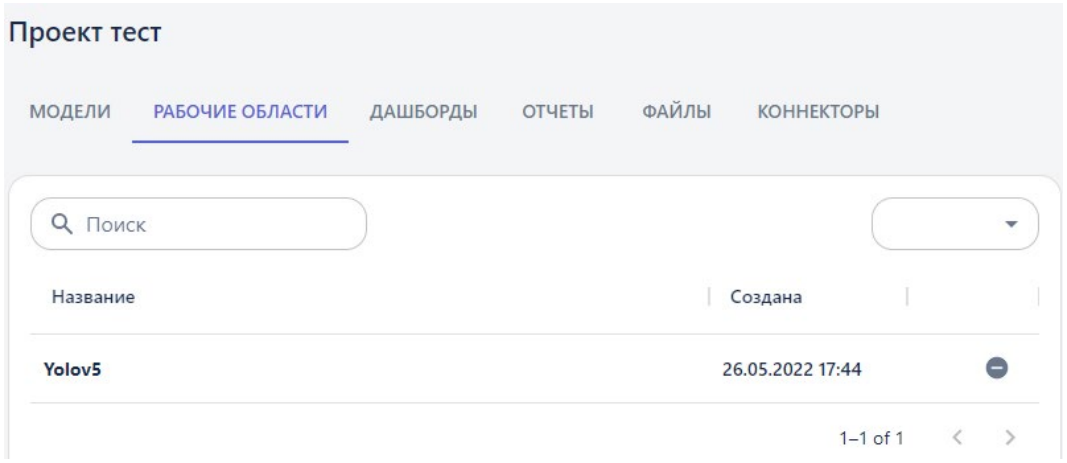


Рисунок 13.10 – Вкладка «Рабочие области»

6. **Добавление сущности «Дашборд»**. Перейдите на вкладку «Дашборды» и нажмите кнопку «Добавить», после чего выберите дашборд из выпадающего списка:

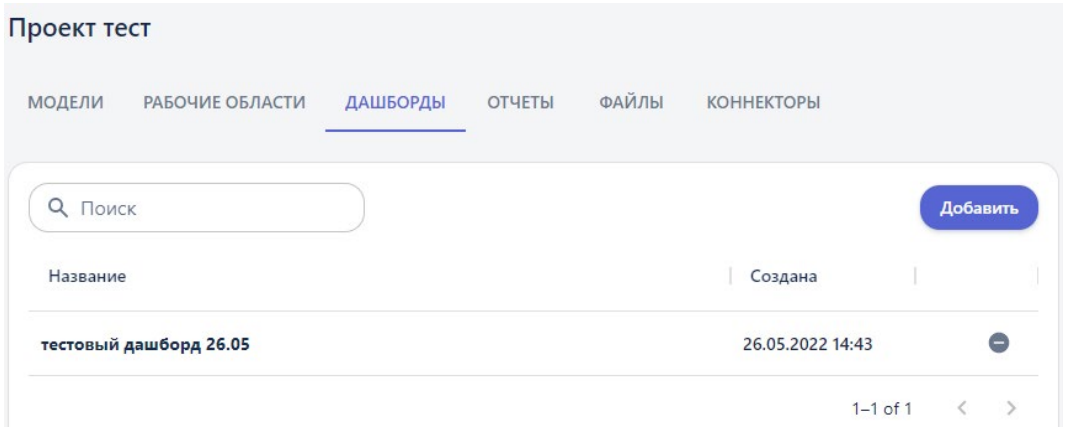


Рисунок 13.11 – Вкладка «Дашборды»

7. **Добавление сущности «Отчет»**. Перейдите на вкладку «Отчеты» и нажмите кнопку «Добавить». Из выпадающего списка выберите отчет для добавления в проект.

8. **Добавление сущности «Файл»**. Перейдите на вкладку «Файлы» и нажмите кнопку «Добавить». Из выпадающего списка выберите файл для добавления в проект:

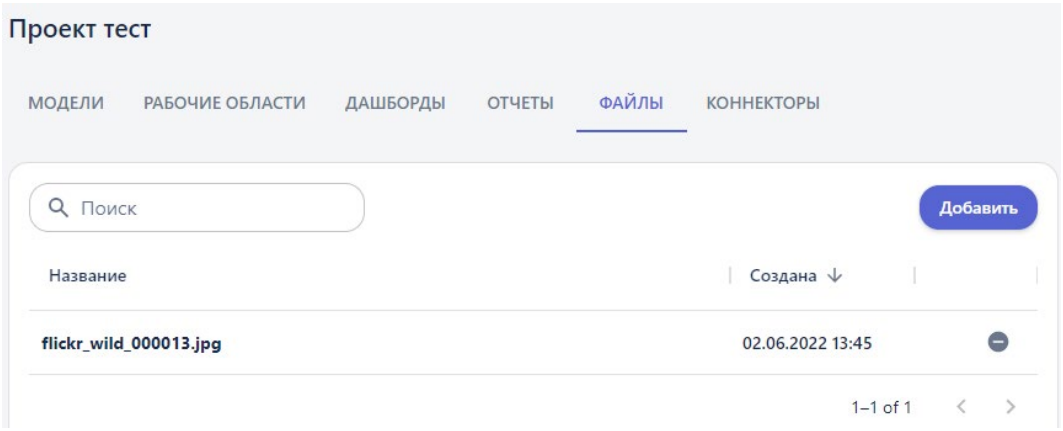


Рисунок 13.12 – Вкладка «Файлы»

9. **Добавление сущности «Коннектор»**. Перейдите на вкладку «Коннекторы» и нажмите кнопку «Добавить». Из выпадающего списка выберите коннектор для добавления в проект:

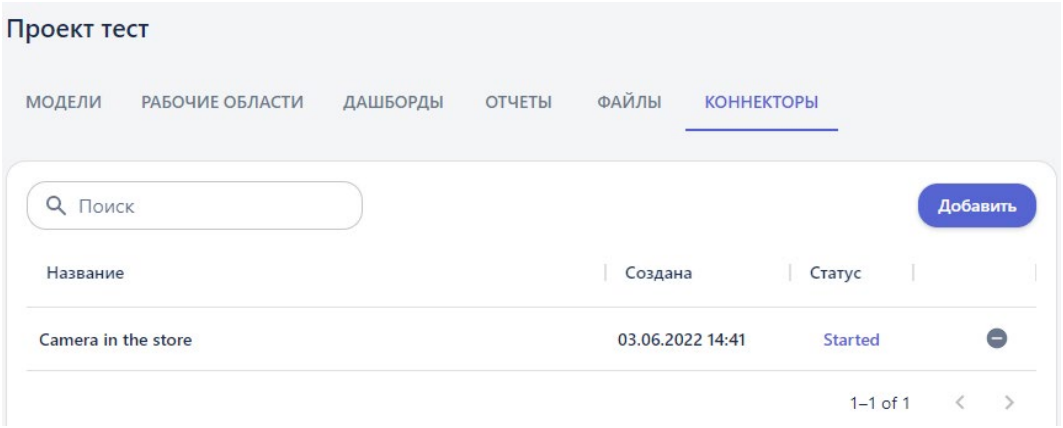


Рисунок 13.13 – Вкладка «Коннекторы»

13.4. Автоматическая сборка и тестирование проектов

На платформе существует сервис автоматической сборки и тестирования проектов. Данный сервис разработ на Python 3.10. Для работы сервиса рекомендуется настроить отдельную виртуальную среду.

Сервис использует библиотеки Selenium и Pytest.

Для запуска тестов необходимо установить указанные выше библиотеки.

13.4.1 Порядок работы

- Скачайте в папку `drivers` нужный драйвер: для браузера Chrome.
- При необходимости установите свойства исполняемых файлов `chmod -x`.

13.4.2 Структура тестов

- drivers - драйверы для управления браузерами
- pages - описания объектов на страницах
- basic_tests - базовые элементарные тесты используемые в `smoke_test.py`
- constconstructor - пакет для конструирования проектов
- project_configs - конфигурации проектов

- logs - автоматически создающаяся папка при запуске проектов, которая содержит логи тестируемых проектов
- smoke_test.py - смоук-тест из элементарных тестов
- 01_create_fire_project.py - тесты проектов (создание сущностей: проект, рабочая область с пайплайнами, дашборд, коннекторы)
- conftest.py - фикстура, в которой настраиваются параметры браузера
- config.json - адрес тестируемой платформы
- configs.py - константы тестируемой платформы

13.4.3 Конструктор проектов

Описание

Конструктор проектов способен создавать проект со всеми необходимыми сущностями внутри:

- Данные
- Источники данных
- ETL
- Коннекторы
- Рабочие области
- Дашборды
- Проекты
    - Прикрепление модели
    - Прикрепление рабочих областей
    - Прикрепление дашбордов
    - Прикрепление файлов
    - Прикрепление отчетов
    - Прикрепление коннекторов

Использование

Для создания нового проекта используйте в качестве примера любой существующий проект:

- `01_create_fire_project.py`
- `02_create_mei_project.py`
- `03_create_series_analysis_project.py`
- `04_create_series_research_project.py`
- `05_create_traffic_pipeline_project.py`
- `06_create_image_classification_project.py`
- `07_create_image_classification_pretrained_project.py`
- `08_create_object_detection_project.py`
- `09_create_shd_project.py`
- `10_create_graph_project.py`
- `11_create_text_classification_project.py`
- `12_create_text_clasterisation_project.py`

В исполняемом файле проекта необходимо указать название проекта и файл конфигурации проекта.

Файл конфигурациии проекта `*_project.json`

Состоит из набора тестовых случаев: **case**

В каждом тестовом случае можно выполнять несколько действий: **action**

Список возможных действий:

- add_file - добавить файл в данные
- delete_file - удалить файл из данных
- check_file - проверить наличие файла в данных
- add_datasource - создать источник данных
- delete_datasource - удалить источник данных
- check_datasource - проверить наличие источника данных
- add_etl - создать ETL
- delete_etl - удалить ETL
- check_etl - проверить наличие ETL
- add_connector - создать коннектор
- delete_connector - удалить коннектор
- check_connector - проверить наличие коннектора
- add_workspace - создать рабочую область
- delete_workspace - удалить рабочую область
- open_workspace - открыть рабочую область
- check_workspace - проверить наличие рабочей области
- add_pipeline - создать пайплайн на рабочей области(указать файл конфигурации пайплайна)
- add_dashboard - создать дашборд
- delete_dashboard - удалить дашборд
- open_dashboard - открыть дашборд
- check_dashboard - проверить наличие дашборда
- add_blocks - _[не реализовано]_ создать блоки на дашборде (указать файл конфигурации пайплайна)
- create_application - создать приложение из модели
- check_application - проверить наличие приложения
- add_project - создать проект
- delete_project - удалить проект
- open_project - открыть проект
- check_project - проверить наличие проекта
- add_project_workspace - добавить рабочую область в проект
- delete_project_workspace - удалить рабочую область из проекта
- check_project_workspace - проверить наличие рабочей области в проекте
- add_project_model - добавить модель в проект
- delete_project_model - удалить модель из проекта
- check_project_model - проверить наличие модели в проекте
- add_project_dashboard - добавить дашборд в проект
- delete_project_dashboard - удалить дашборд из проекта
- check_project_dashboard - проверить наличие дашборда в проекте

- add_project_report - добавить отчёт в проект
- delete_project_report - удалить отчёт из проекта
- check_project_report - проверить наличие отчёта в проекте
- add_project_file - добавить файл в проект
- delete_project_file - удалить файл из проекта
- check_project_file - проверить наличие файла в проекте
- add_project_connector - добавить коннектор в проект
- delete_project_connector - удалить коннектор из проекта
- check_project_connector - проверить наличие коннектора в проекте

Файл конфигурации пайплайна `*_pipeline.json`

Состоит из набора действий: **action**

Список возможных действий (R:required, O:optional):
* **add_block** - добавить блок.
* **block_id** - [R] идентификатор блока. Должен быть уникальным для каждой конфигурации пайплана. Используется при создании соединений и запуске
* **block_type** - [R] тип создаваемого блока ("start", "data_source", "process", "gateway", "info")
* **name** - [R] имя функции используемой в блоке(название блока)
* **check** - [R] параметр включащий проверку наличия блока после создания. . ("1":вкл/"0":вылк)
* **report** - [O] параметр включащий отчет выполнения пайплайна. ("1":вкл/"0":вылк)
* **coordinates** - [R] координаты на которые необходимо создать. Координаты от центра(+X:вправо, -X:влево, -Y:вверх, +Y:вниз)
* **parameters** - [O] дополнительные параметры блока(функции)

На данный момент существует несколько типов полей **parameter_type** в настройках функции:
* **text** - текстовое поле
* **multivalue** - множественный ввод через enter
* **select** - селектор одиночный
* **multiselect** - селектор множественный
* **checkbox** - переключетель
* **file** - выбор файла (указать путь через "/")
* **connector** - выбор коннектора
* **add** - добавить слои(кнопка "+")

* **block_id** - переместить блок от текущего положения(+X:вправоб, -X:влево, -Y:вверх, +Y:вниз)
* **move_workspace** - передвинуть рабочую область (+X:влево, -X:вправо, -Y:вниз, +Y:вверх)
* **zoom_workplace** - приблизить/отдалить рабочую область (-value/+value)

* **add_connections** - создать соединения блоков блок(указать файл конфигурации соединений)

* **play** - запустить блок Запуск(указать какие блоки должны проверяться во время выполнения)

Файл конфигурации соединений `*_connections.json`

Состоит из наборов соединений.

Каждое соединение содержит 4 параметра
* **from_block** - блок, из которого создаем соединение. Указываем **block_id**
* **from_socket** - индекс выходящего сокета на блоке
* **to_block** - блок, в который создаем соединение. Указываем **block_id**
* **to_socket** - индекс входящего сокета на блоке

Файл конфигурации дашборда `*_dashboard.json` [в процессе разработки]

Структура логирования:
* **INFO** - обязательные уровни логирования (для последующего отчета)
  * **PROJECT** - корневой уровень проекта
  * **TESTCASE** - Тестовый случай (в соответствии с конфигурацией `*_project.json`)
  * **ACTION** - Действия в тестовом случае
  * **PARAMETER** - параметры действий в тестовых случаях
  * **CHECK** - тестовые проверки
* **WARNING** - Некорректные данные, введенные пользователем
* **ERROR** - упавшие тесты

14. Настройка подключения к источникам данных

В платформе BASIS AI реализована возможность подключения к *внешним системам*, выступающим в качестве *источников данных* для Платформы. При этом данные в режиме реального времени могут поступать из следующих источников: БД (ClickHouse, PostgreSQL, MongoDB) и камеры видеонаблюдения. В данном разделе рассказывается, как настроить в Системе такие подключения. Подробно о визуализации информации из коннекторов на дашбордах написано в разделе «Работа с дашбордами»

Все подключения настраиваются в пункте меню «Соединения», где создаются сущности «Коннектор». *Коннекторы* объединяют в себе источник подключения и запрос на получение данных из него. В данном разделе описаны все типы коннекторов на платформе и сценарии работы с ними на примерах с тестовыми данными.

14.1 Типы коннекторов

1. **«ClickHouse»**

Данный коннектор предназначен для подключения к БД «ClickHouse». Настройка подключения описана выше в разделе **Настройка подключения на примере ClickHouse**.

2. **«PostgreSQL»**

Данный коннектор предназначен для подключения к БД «PostgreSQL». Настраивается по аналогии с коннектором «ClickHouse».

3. **«Mongo»**

Данный коннектор предназначен для подключения к БД «MongoDB». Настраивается по аналогии с коннектором «ClickHouse».

4. **«Table_app»**

На вход коннектора поступают *табличные данные* в онлайн режиме, например из БД «PostgreSQL»**.** Чтобы анализировать входные данные используется обученная в системе *модель*. Рассмотрим коннектор для задачи прогнозирования лесных пожаров, которая на вход принимает данные о погодных условиях в онлайн-режиме (эти табличные данные аналогичны тем, на которых обучалась модель):

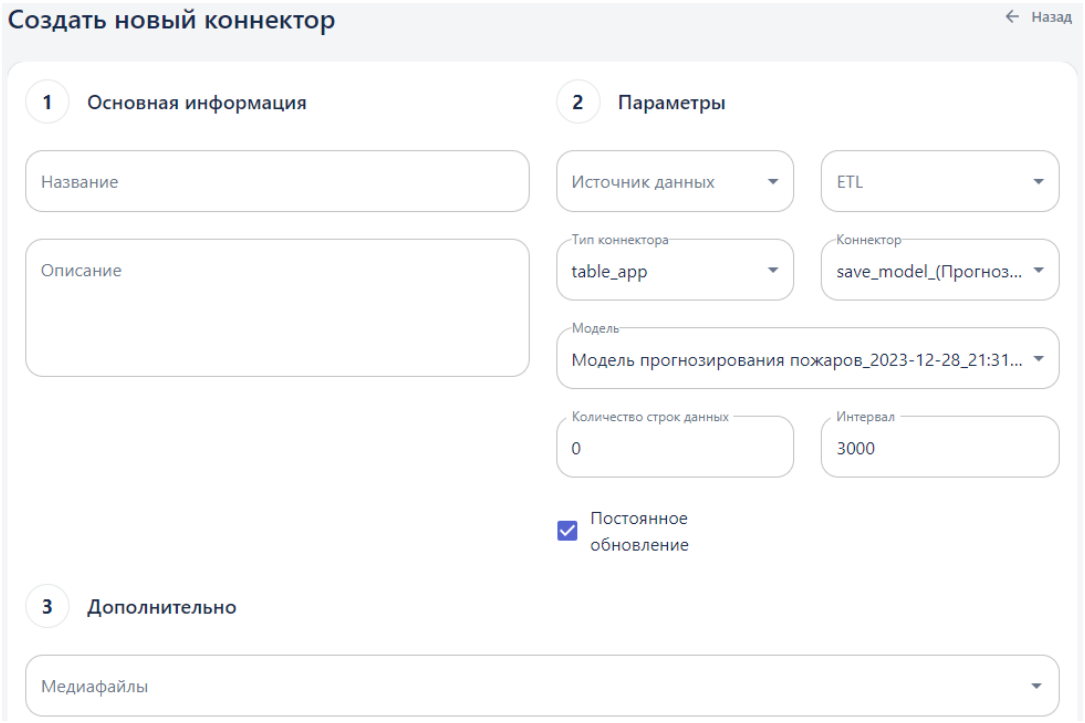


Рисунок 14.1 – Настройка коннектора «Table_app»

В поле «Коннектор» выбирается *коннектор*, который подключен к таблице с погодными условиями. А в поле «Модель» выбирается обученная модель прогнозирования лесных пожаров. Устанавливается галочка в поле «Постоянное обновление» – активирование режима ожидания новых данных на входе настраиваемого коннектора.

5. «Save_table»

Данный коннектор предназначен для сохранения в Системе в виде файлов (на данный момент реализовано сохранение файлов в формате csv) табличных данных, поступающих из сторонних систем. Директория для сохранения файлов в Системе – это раздел «Данные».

В поле «Коннектор» указывается коннектор для подключения к таблице внешнего источника. Устанавливается галочка в поле «Постоянное обновление».

Для такого типа коннектора обязательно указать на выбор:

- Количество строк данных - количество строк из таблицы, которые будут сохранены.
- Интервал - промежуток времени в секундах, в течение которого коннектор должен собирать информацию из таблицы, по истечению этого времени загрузка прекратится и файл будет сохранен.

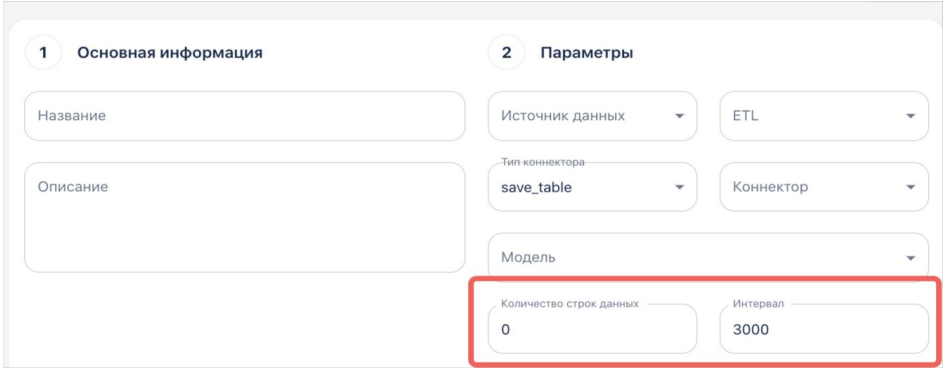


Рисунок 14.2 – Настройка коннектора «save_table»

6. «video_detection_app»

Данный коннектор используется в задаче распознавания объектов (Object detection) на видеопотоке данных в онлайн-режиме. Пример настройки коннектора:

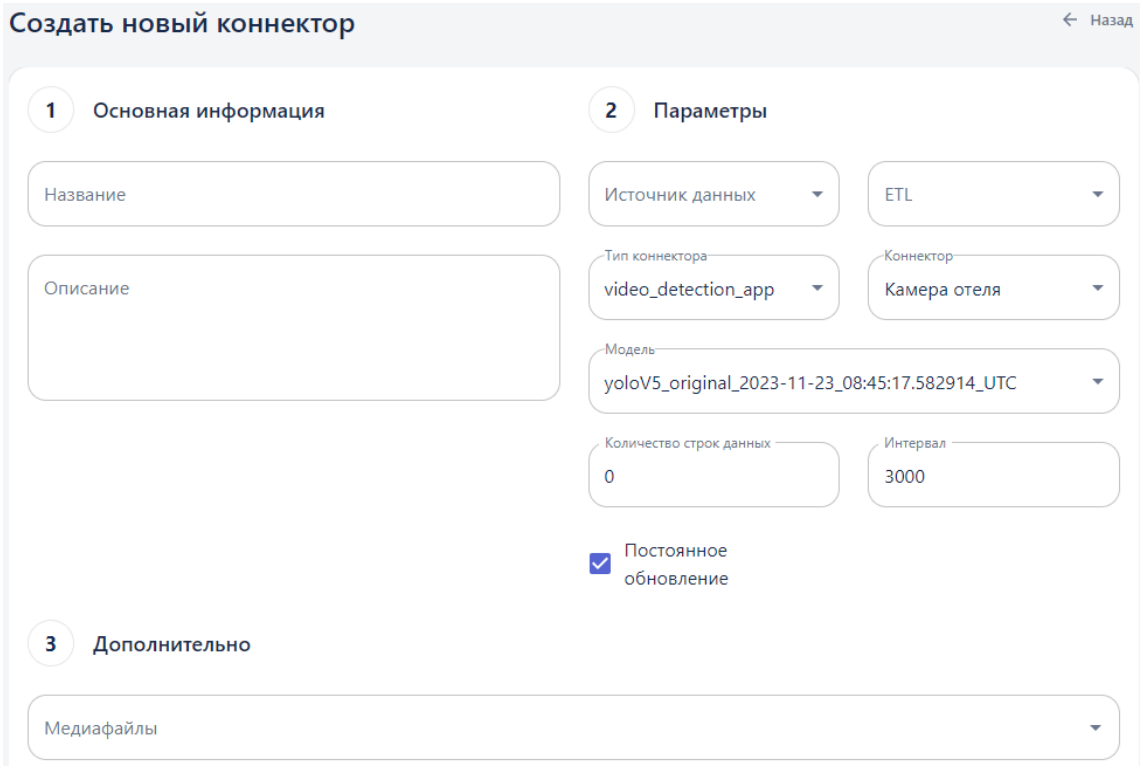


Рисунок 14.2 – Настройка коннектора «video_detection_app»

В поле «Коннектор» выбирается камера для подключения. А в поле «Модель» выбирается обученная модель распознавания изображений, которая умеет распознавать определенные объекты на изображениях/видео. При этом модель умеет определять именно те объекты, распознавать которые она обучалась, и предполагается, что на видео с камеры также будут эти объекты. Устанавливается галочка в поле «Постоянное обновление», для получения видео с камеры в режиме реального времени.

Если не устанавливать галочку в поле «Постоянное обновление», то выполняется автоматическое разбиение полученного видеопотока на эпизоды (*раскадровка*). Такие эпизоды партициями передаются модулю «Apache kafka», который в Системе отвечает за передачу данных. Внешние системы могут подключиться к брокеру сообщений, и получить доступ к раскадрованному видео – на просмотр и обработку.

7. «Images_detection_app»

Данный коннектор также используется в задаче распознавания объектов, только входными данными для анализа являются на выбор – готовый видеофайл, или серия из нескольких изображений. Пример настройки коннектора:

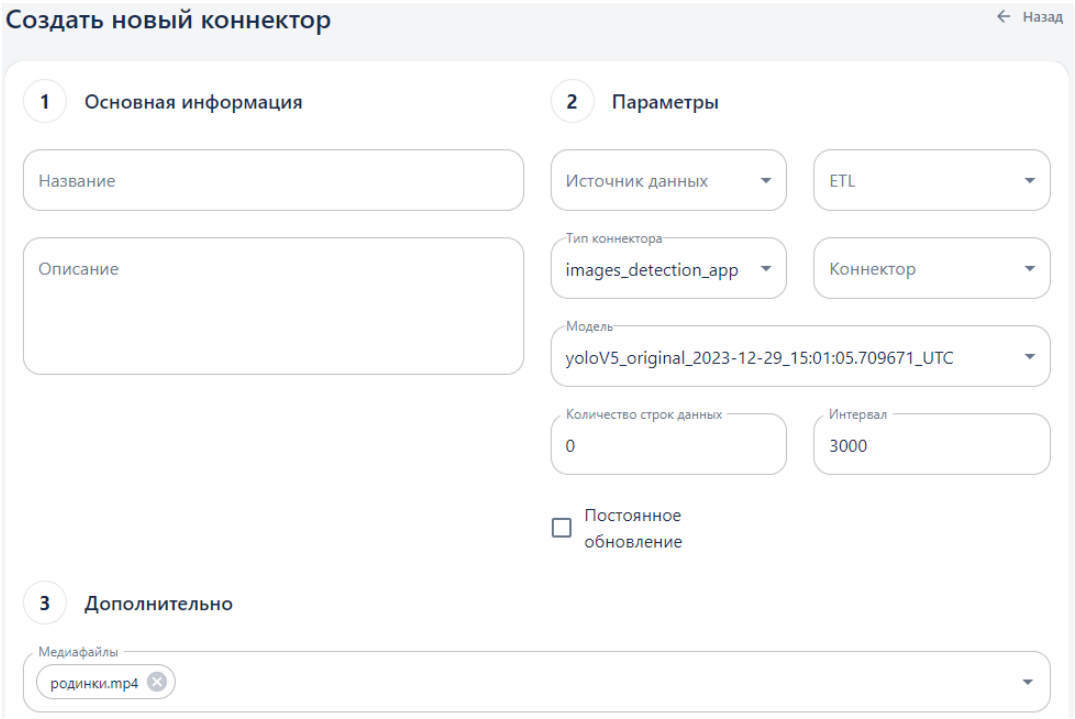


Рисунок 14.3 – Настройка коннектора «Images_detection_app»

В поле «Модель» выбирается обученная модель «Yolov5». В поле «Медиафайлы» выбирается либо один видео файл, загруженный в раздел «Данные», либо несколько изображений. В поле «Постоянное обновление» галочка не устанавливается. После запуска коннектора с данным типом автоматически выполняется сохранение размеченного видеофайла/серии изображений в разделе «Данные».

***Важно**: **чтобы модель могла распознавать объекты, обучение должно пройти минимум на трехстах эпохах**.

8. «Classification_app»

Данный коннектор используется в задаче *классификации изображений*, где на вход коннектора для анализа подается серия из нескольких изображений. Пример настройки коннектора:

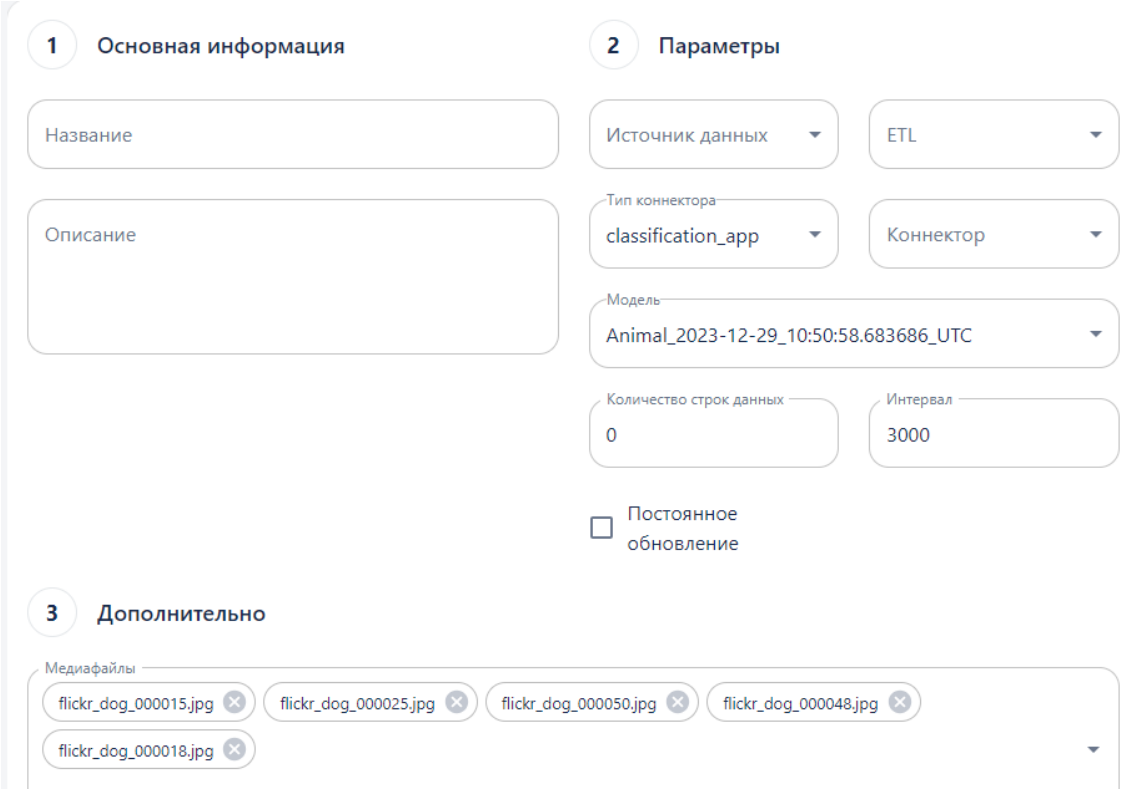


Рисунок 14.4 – Настройка коннектора «Classification_app»

В поле «Модель» выбирается обученная модель классификации изображений. В поле «Медиафайлы» – серия изображений, которые необходимо классифицировать с использованием обученной модели.
*Для данного типа коннектора не нужно устанавливать признак «Постоянное обновление», так как анализируются данные, загружаемые с локального устройства.

9. «Constructor» (автоматически создаваемый)

Условием создания коннектора является следующее: на Платформе создается блок-схема, где один из элементов имеет на выходе *визуализацию* – выходным параметром элемента является *таблица*, *график* и т.д. Пользователь запускает такую блок-схему, и после успешной отработки элемента с визуализацией в Системе создается коннектор. Число создаваемых коннекторов при запуске блок-схемы соответствует числу элементов с визуализацией на этой блок-схеме.

Название коннектора формируется из названия элемента и названия рабочей области, и должно являться уникальным в рамках Системы. Пример – *yolov5_train_(yolov5_noses_eyes)*, где название рабочей области указано в скобках.

Такой коннектор автоматически создается в статусе «Started» – пользователь не должен запускать коннектор, и может сразу же перейти к просмотру данных коннектора в окне дашборда.

14.2 Порядок работы с коннекторами

Предварительно коннектор должен быть создан и запущен (за исключением служебных коннекторов, которые создаются и запускаются автоматически). Только после этого выполняется подключение к нему через окно дашборда.

14.2.1 Создание коннектора

Создание коннекторов осуществляется в разделе меню «Соединения». Для коннекторов, в которых настраивается подключение к внешним источникам данных (к внешним базам данных, к камере видеонаблюдения), дополнительно создаются сущности – *источник данных*, и *ETL*. Для остальных типов, эти сущности не создаются, а сразу создается сущность «Коннектор».

Для того чтобы создать новый коннектор, перейдите на вкладку «Коннекторы» и нажмите на кнопку «Создать коннектор»:

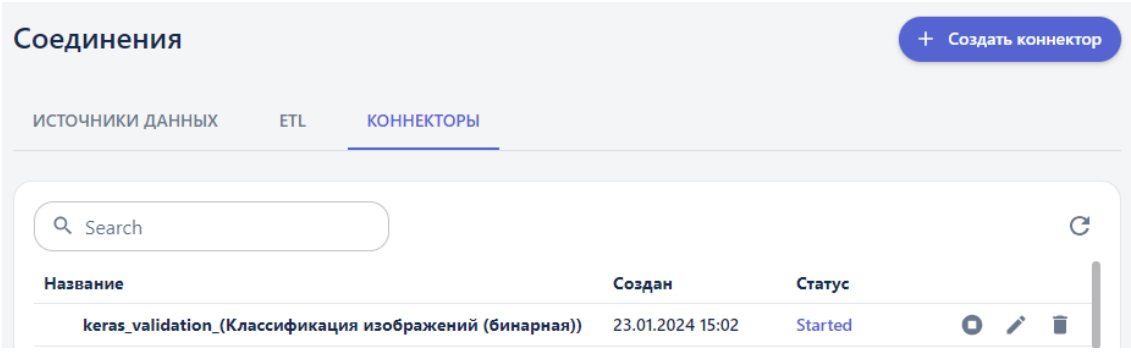


Рисунок 14.5 – Переход к созданию коннектора на вкладке «Коннекторы»

В открывшейся форме выберите тип коннектора в одноименном поле «Тип коннектора», и заполните поля:

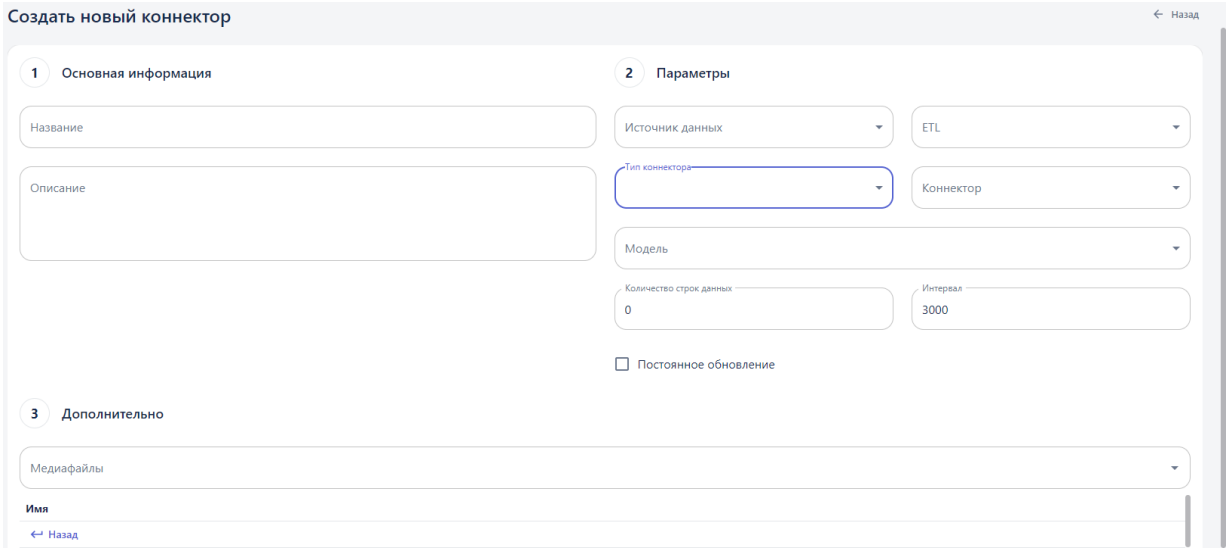


Рисунок 14.6 – Форма создания нового коннектора

После заполнения формы нажмите кнопку «Создать».

14.2.2 Запуск коннектора

Сразу после создания коннектору присваивается статус «Stopped». Далее коннектор запускается для того, чтобы активировать его работу (начать получать данные из внешних

источников, запустить алгоритм обработки данных). Для этого нажмите кнопку « » в строке с коннектором:

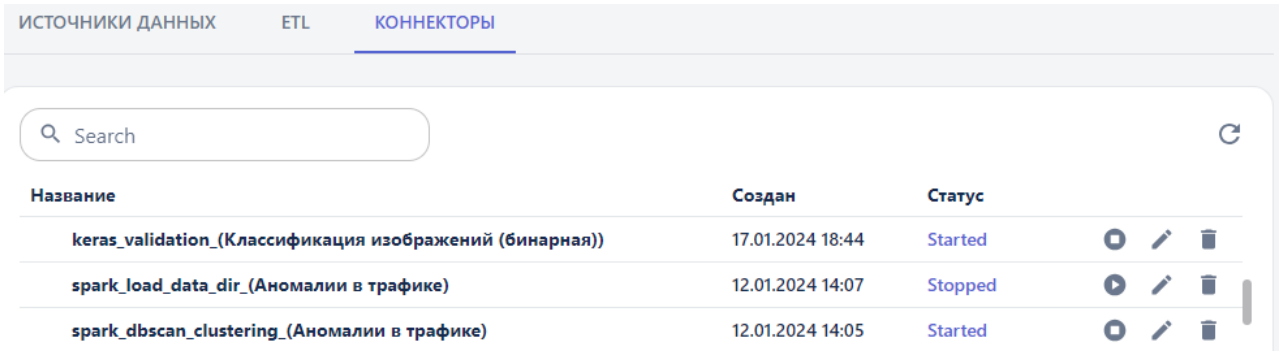


Рисунок 14.7 – Запуск коннектора

В результате коннектору присваивается статус «Started», и он готов к визуализации на дашборде.

14.2.3 Подключение к коннектору на дашборде

Для подключения коннектора на дашборде нажимается кнопка DASH, чтобы добавить интерактивный блок на рабочую область. После добавления на интерактивного блоке нужно нажать в правом углу. Откроется модальное окно со списком коннекторов для подключения.

14.3 Настройка подключения на примере ClickHouse

Действия:

1. Перейдите в пункт меню «Соединения»:

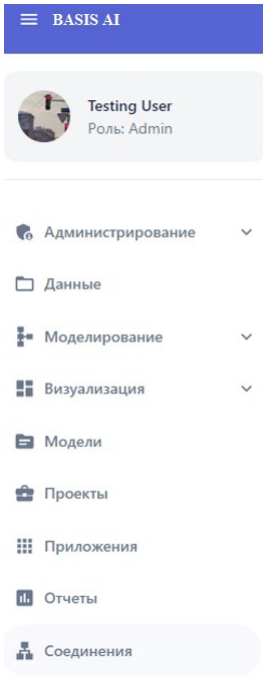


Рисунок 14.8 – Пункт меню Соединения

Откроется страница «Соединения» на первой вкладке «Источники данных», на которой отображаются все ранее созданные источники:

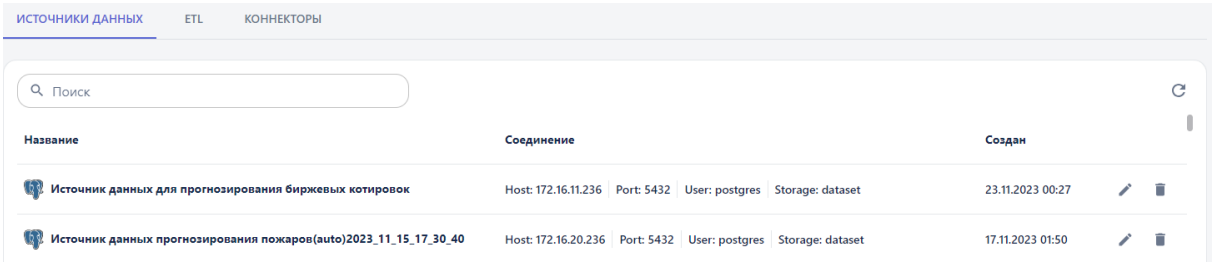


Рисунок 14.9 – Вкладка «Источники данных»

2. **Создание нового источника**. Нажмите на кнопку «Создать источник данных». Откроется окно «Создание нового источника данных»:

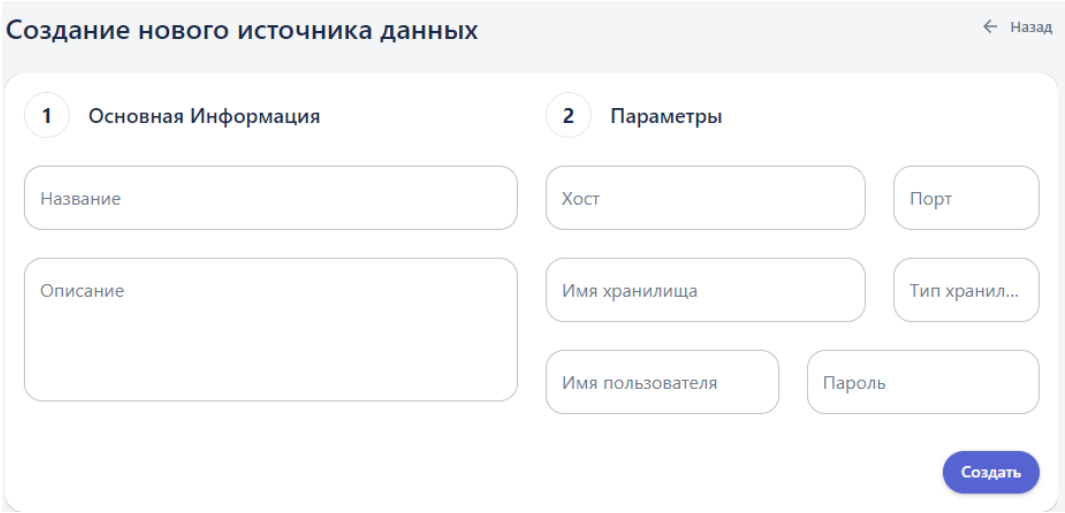


Рисунок 14.10 – Окно настройки источника данных

Заполните поля:

- ➔ *Название*. Пользователь задает название источника «TEST DATA SOURCE (clickhouse)», к которому будет настраиваться подключение.
- ➔ *Хост*. Указывается хост протокола TCP/IP, т.е. сетевой интерфейс устройства, предоставляющего сервис формата «клиент-сервер», где сервером выступает БД **ClickHouse**, а клиентом – платформа BASIS AI. По сути это IP-адрес подключаемой БД. Необходимо указать «172.16.11.116».
- ➔ *Порт* – номер порта, по которому устанавливается соединение с сервером, на котором установлена БД **ClickHouse**. Указать «12366».
- ➔ *Имя хранилища* – название базы данных, которое указано на подключаемом сервере. Указать «default».
- ➔ *Тип хранилища*. Из выпадающего списка необходимо выбрать тип «clickhouse»:

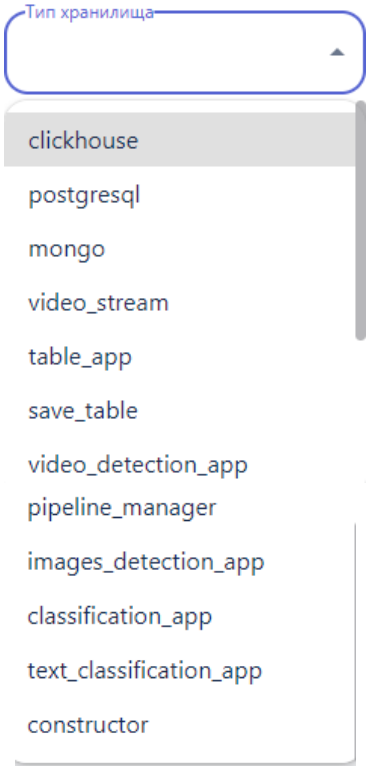


Рисунок 14.11 – Список возможных типов хранилища

Описание всех типов коннекторов представлено в разделе **Классификация коннекторов**.

➔ *Имя пользователя, пароль* – параметры учетной записи администратора внешнего сервера для разрешения доступа к данным. Указать пользователя «clickhouse_operator», и пароль для него «clickhouse_operator_password».

➔ *Описание.* Вводится дополнительная информация по источнику, необязательное поле.

Для регистрации в Системе источника нажмите кнопку «Создать».

3. **Создание нового ETL**. Сущность «ETL» (дословно Extract, Transform, Load – с англ. извлечение, преобразование загрузка) содержит в себе sql запрос для извлечения данных из источника. То есть в шаге 2 создается источник и прописывается запрос для извлечения данных из него.

   3.1. Перейдите на вкладку «ETL»:

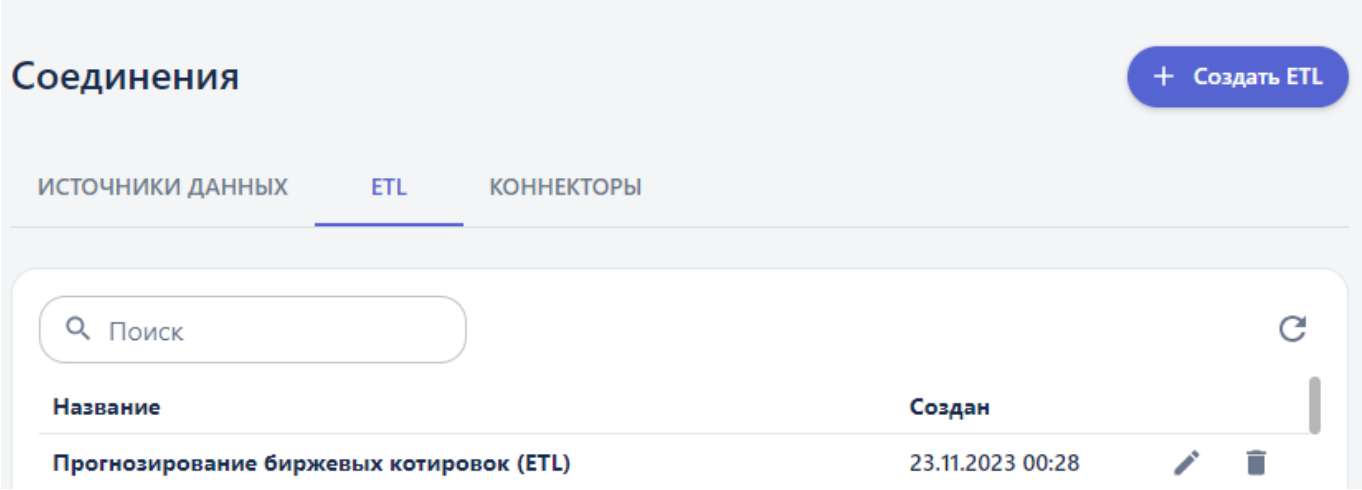


Рисунок 14.12 – Вкладка ETL

3.2. Нажмите на кнопку «Создать ETL». Откроется окно «Создать новый ETL»:

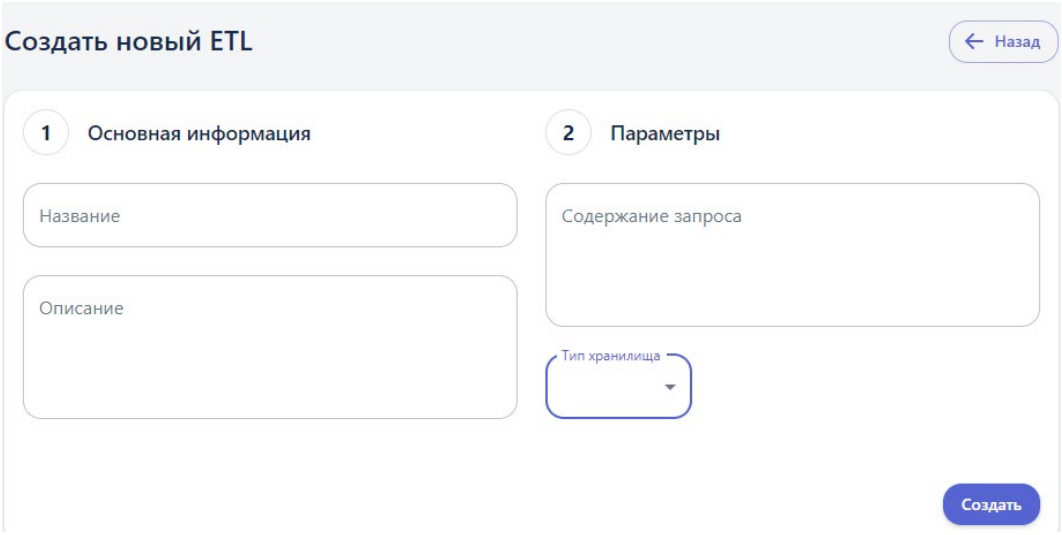


Рисунок 14.13 – Окно настройки ETL

3.3. Заполните поля:

- ➔ *Название. Пользователь вручную задает название ETL «TEST ETL (clickhouse)» – запрос на извлечение данных.*
- ➔ *Содержание запроса. Прописывается непосредственно sql запрос для извлечения данных из внешнего сервера. При этом указывается название таблицы, из которой данные извлекаются, в запросе «SELECT * FROM stock». Чтобы извлечь данные только из первых ста строк этой таблицы используется запрос «SELECT * FROM stock LIMIT 100».*
- ➔ *Тип хранилища. Выбирается тип «clickhouse».*
- ➔ *Описание (необязательное поле).*

3.4. Нажмите кнопку «Создать».

4. **Создание нового коннектора**:

4.1. Перейдите на вкладку «Коннекторы»:

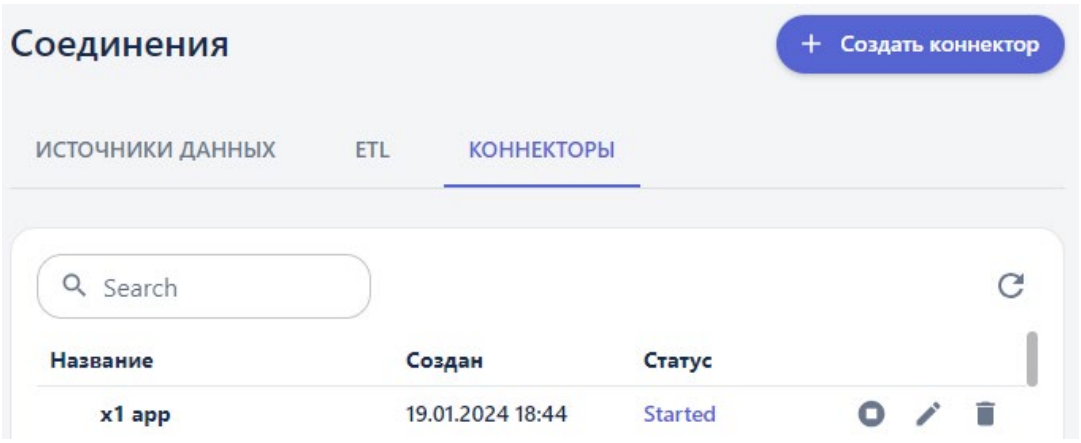


Рисунок 14.14 – Вкладка коннекторов

4.2. Нажмите на кнопку «Создать коннектор». Откроется окно «Создать новый коннектор»:

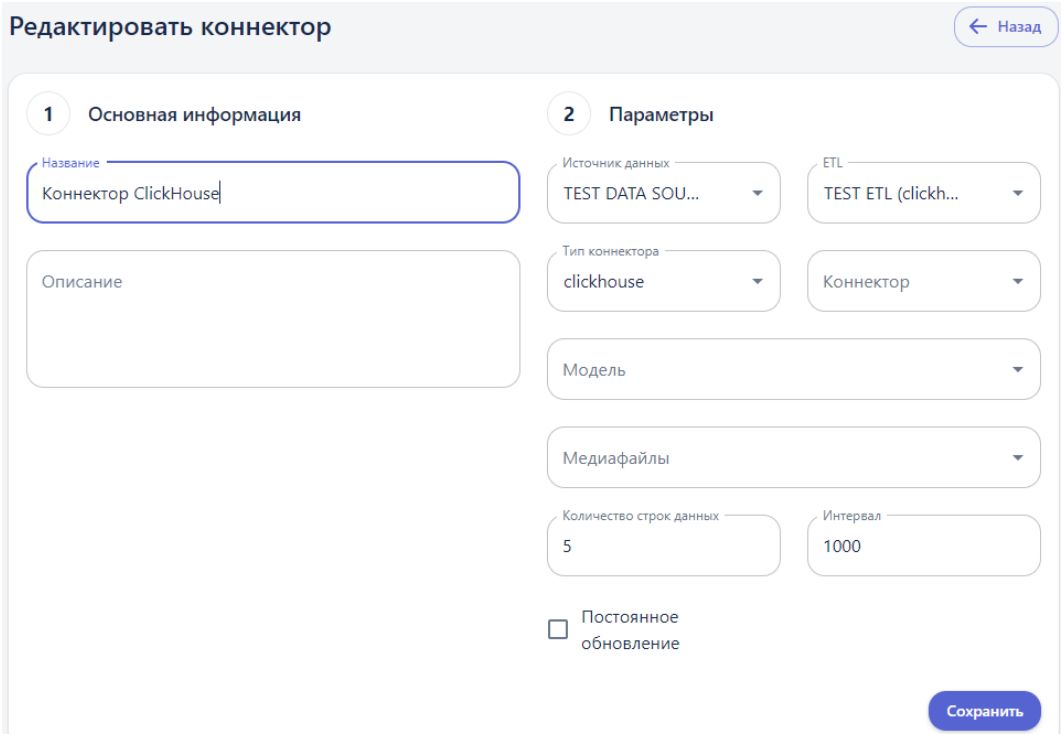


Рисунок 14.15 – Окно настройки коннектора

4.3. Заполните поля:

➔ *Название.* Пользователь вручную задает название создаваемого коннектора «Коннектор ClickHouse».

➔ *Источник данных.* Из списка выбирается источник «TEST DATA SOURCE (clickhouse)», созданный в шаге 2.

➔ *ETL.* Из списка выбирается ETL «TEST ETL (clickhouse)», созданный в шаге 3.

➔ *Тип коннектора.* По умолчанию выбирается первый тип из списка – clickhouse, оставить выбранное значение.

➔ *Количество строк данных.* Данные из внешней БД поступают порциями, по указанному или меньшему количеству строк за раз.

➔ *Интервал* – периодичность, с которой выполняются запросы во внешнюю БД. Заполняется числовым значением (в миллисекундах), или значением в формате «Дата» (*второй вариант не реализован в текущей версии*). Если задать интервал 1000 мс, и указать количество строк пять, каждую секунду будет запрашиваться пять записей.

➔ *Постоянное обновление*. Признак устанавливается, когда данные ожидаются бесконечно. Если признак не установить, запрос данных завершится, при получении их в полном объеме.

➔ *Описание.*

Остальные поля на форме создания коннектора для текущего сценария не заполняются, они используются при создании других типов коннекторов. Нажмите кнопку «Создать».

4.4. Сразу после создания коннектору присваивается статус «Stopped»:

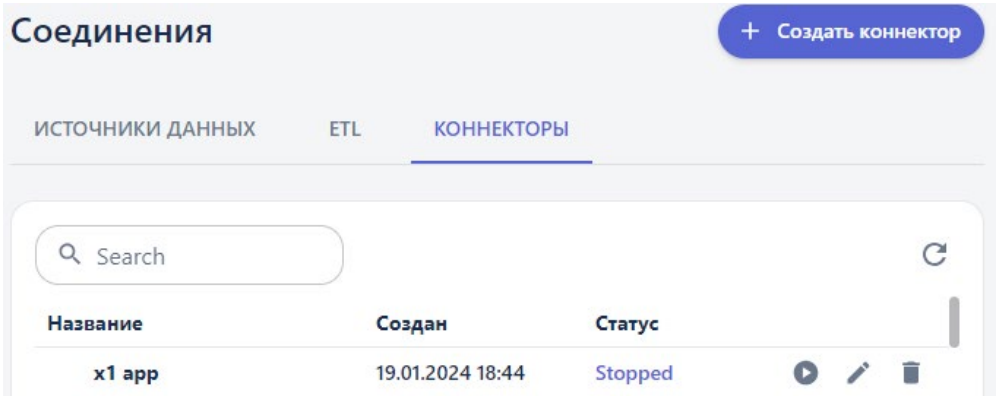


Рисунок 14.16 – Статус запуска коннектора

5. **Запуск коннектора**. Чтобы данные из источника начали поступать в Систему необходимо запустить коннектор. Для этого нажмите на кнопку « ▶ » в строке с коннектором. В результате статус меняется на значение «Started».

   Вы так же можете отображать табличные данные ClickHouse в режиме реального времени, используя для визуализации сущность «Дашборд» и подключаясь к запущенному коннектору.

14.4 Получение данных с камеры видеонаблюдения

Подключение к камере наблюдения настраивается по аналогии с подключением к БД **ClickHouse**. Отличие заключается в выборе *канала подключения*, а сам порядок действий повторяется:

1. **Создание нового источника**. На вкладке «Источники данных» нажмите кнопку «Создать источник данных». Заполните поля:

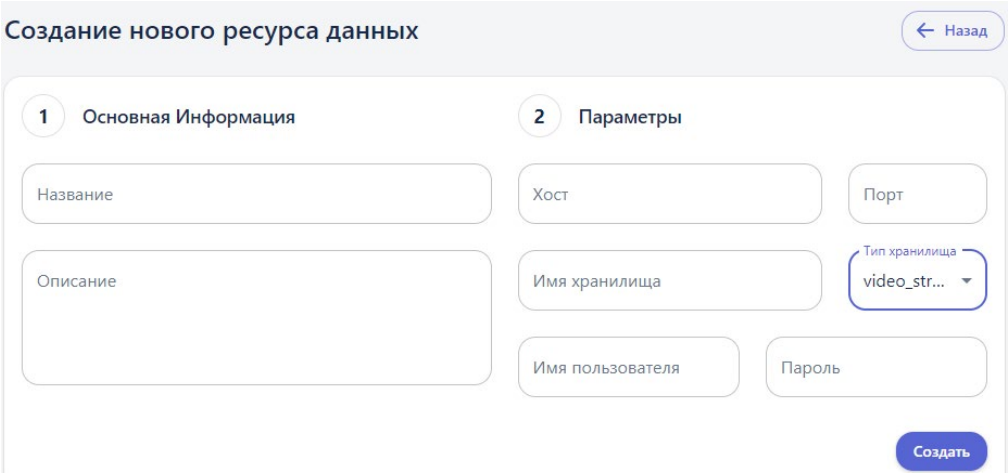


Рисунок 14.17 – Создание источника данных типа video_stream

- ➔ *Название. Пользователь задает название источника «Камера магазина».*
- ➔ *Хост. Пользователь вводит сетевой адрес камеры (в данном примере: «http://158.58.130.148:80/mjpg/video.mjpg»). Все сведения, вводимые начиная с этого поля, необходимо уточнять непосредственно у владельцев смежной системы, с которой настраивается соединение.*
- ➔ *Порт. Пользователь вводит номер порта для подключения к камере «0» (обязательно числовое значение).*

- ➔ *Имя хранилища.Пользователь вводит название камеры «pass».*
- ➔ *Тип хранилища. Из выпадающего списка выбрать тип «video_stream».*
- ➔ *Имя пользователя, пароль. Пользователь вводит параметры учетной записи администратора, имеющего доступ к видеокамере. Указываются соответственно пользователь «pass» и пароль «pass».*
- ➔ *Описание.*

Для регистрации в Системе источника нажмите кнопку «Создать» (далее действие по сохранению введенных настроек предполагаются по умолчанию).

2. **Создание ETL.** На вкладке ETL нажмите на кнопку «Создать ETL» и заполните поля следующим образом:

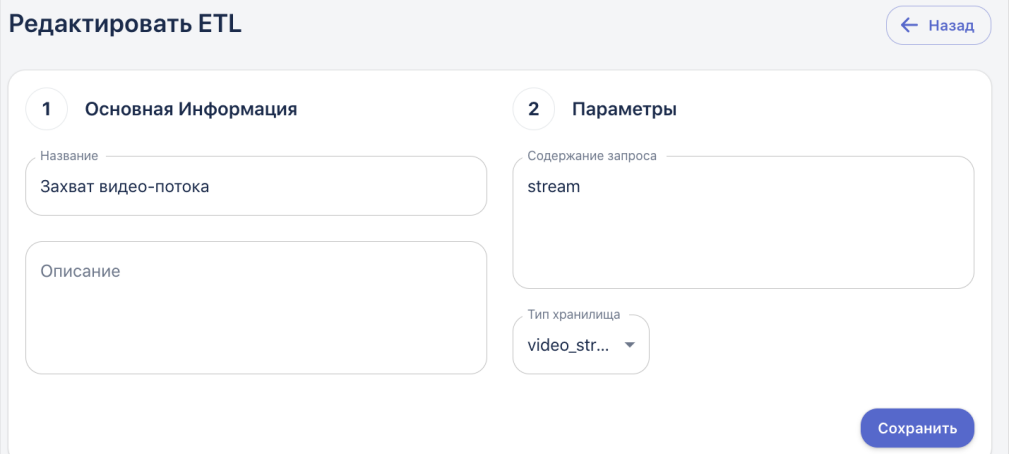


Рисунок 14.18 – Параметры ETL типа video stream

В случае если коннектор создаётся для захвата видеопотока (его отображения в режиме онлайн на дашборде), тогда в поле «Содержание запроса» укажите «stream». Для коннекторов, которые подразумевают сохранение видео, запрос должен содержать «save».

3. **Создание нового коннектора**. Перейдите на вкладку «Коннекторы», нажать на кнопку «Создать коннектор» и в открывшемся окне заполните поля:

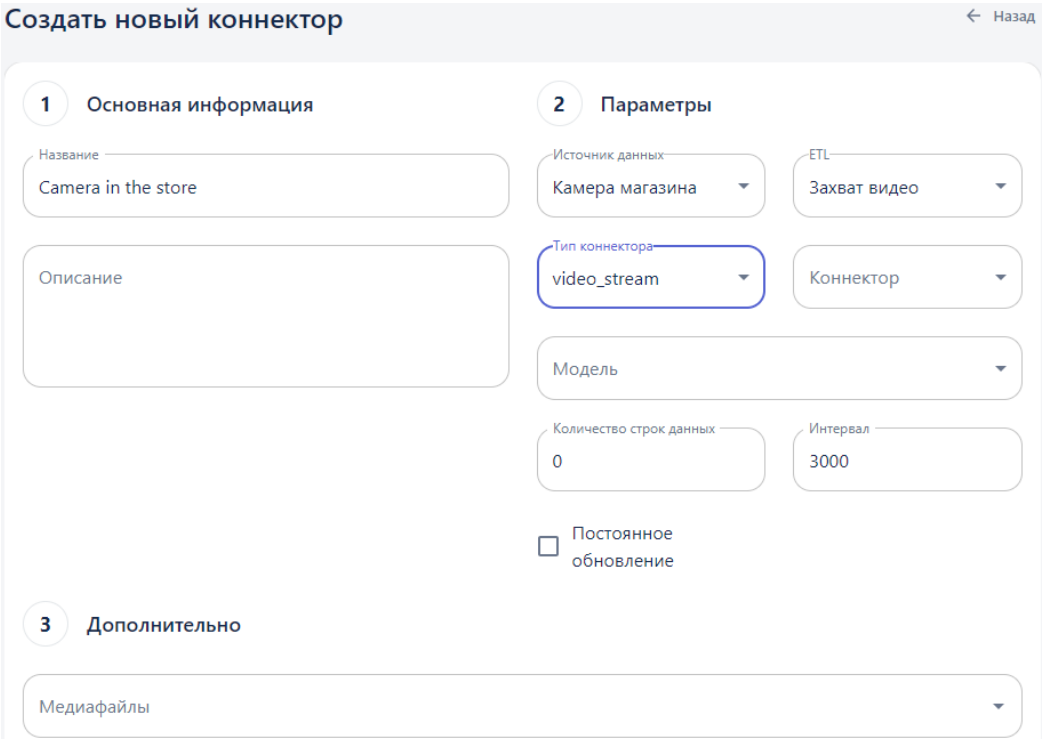


Рисунок 14.12 – Параметры коннектора типа video stream

- ➔ *Название. Пользователь задает название создаваемого коннектора «Camera in the store».*
- ➔ *Источник данных. Из списка выбирается источник, созданный в шаге 1.*
- ➔ *ETL. Из списка выбирается ETL «Захват видео-потока» – запрос на получение онлайн видео потока данных с камеры (запрос «stream»).*
- ➔ *Для коннектора с типом «video_stream» доступен для выбора еще один ETL «Сохранение видеопотока» – запрос на сохранение полученных данных с камеры в раздел «Данные» → «Видео» (запрос «save»). Убедиться, что данный тип ETL доступен для выбора. Продолжение сценария, как сохранить видеопоток, см. в разделе 4.13.*
- ➔ *Тип коннектора. Выбирается тип «video_stream».*
- ➔ *Описание (необязательное поле).*

Остальные поля на форме для данного типа коннектора остаются незаполненными.

4. **Запуск коннектора**. На вкладке «Коннекторы» в строке с коннектором «Camera in the store» нажмите кнопку запуска « ». Коннектору присваивается статус «Started» – с этого момента в Систему начинают поступать данные из внешнего источника:

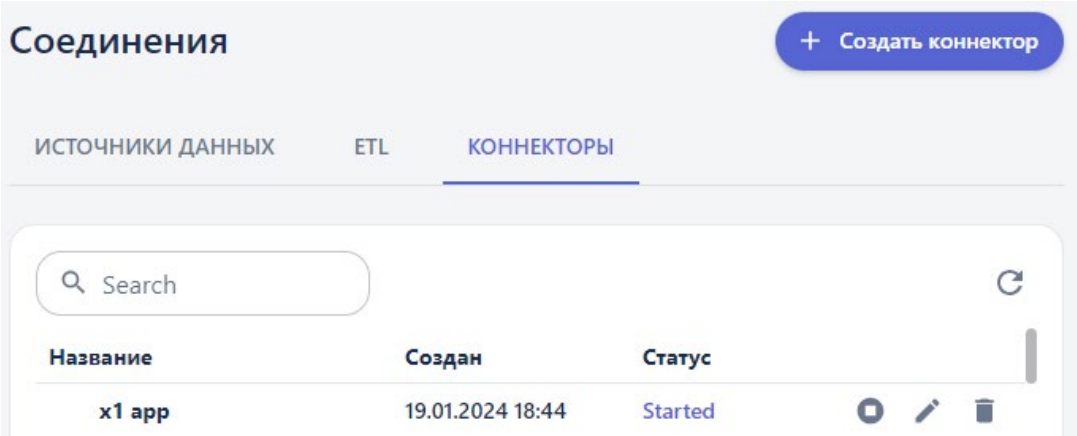


Рисунок 14.19 – Статус запуска коннектора

Для *остановки коннектора* в строке с ним нажмите кнопку « ».

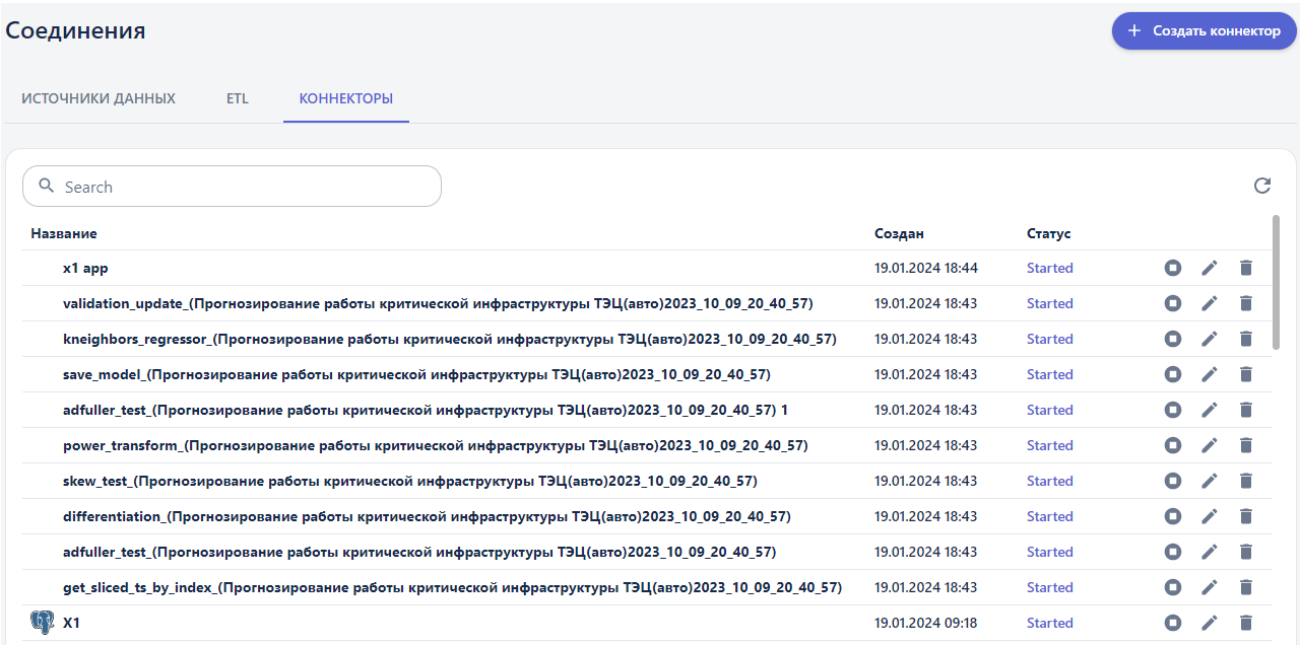


Рисунок 14.20 – Выбор коннектора для отображения в окне дашборда

Для выбора и подключения коннектора он выбирается из списка. Данные, получаемые от коннектора, отобразятся на интерактивном блоке после подключения:

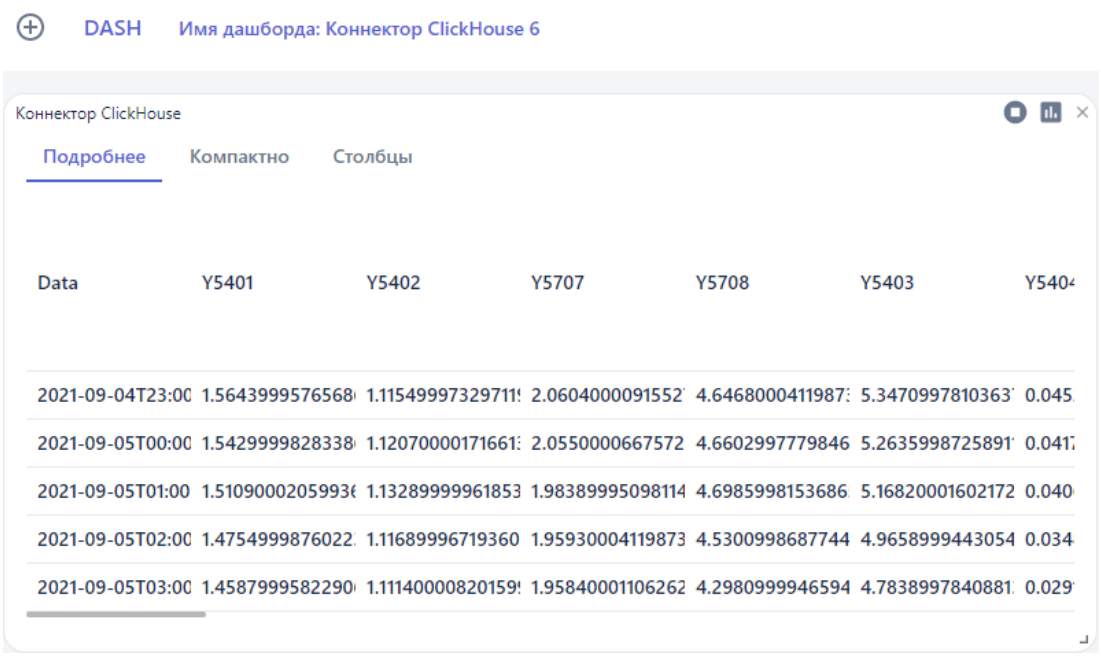


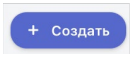
Рисунок 14.21 – Дашборд с запущенным коннектором

15. Примеры работы с Платформой

15.1 Обучение модели прогнозирования температуры воды и газов в котле

16. **Загрузка входных данных:**

16.1. В левой части главного окна на панели вкладок Системы откройте вкладку «Данные».

16.2. На открывшейся странице нажмите кнопку 

16.3. В открывшемся окне в качестве Типа выберите «Категория», в поле Название ввести вручную «МЭИ» и нажмите на кнопку «Создать»:

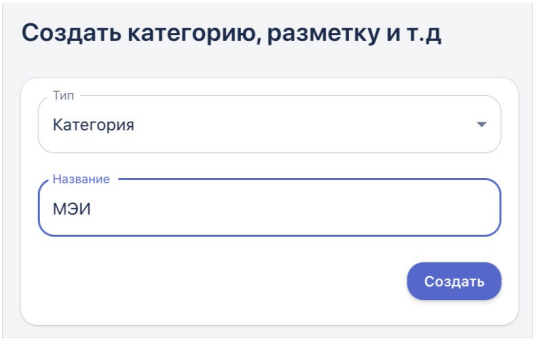


Рисунок 15.1.1 - Создание новой папки в разделе Данные

16.4. В разделе «Данные» появится папка «МЭИ, для загрузки файлов перейдите в неё и нажмите на кнопку «Загрузить» на верхней панели:

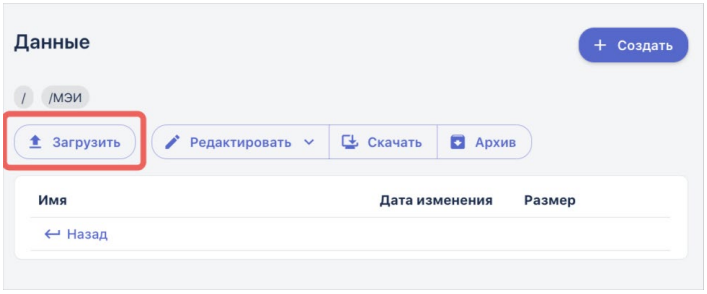


Рисунок 15.1.2 - Загрузка файлов в папку

16.5. В открывшемся окне нажмите на кнопку «Выбрать файлы» и укажите путь к заранее подготовленному файлу **med1d.csv**, в котором содержатся данные о температуре газов и воды в котле за определенный промежуток времени. Второй вариант – перенести файлы в этот раздел по технологии «drag n drop».

Выбранные файлы отобразятся в нижней части окна загрузки:

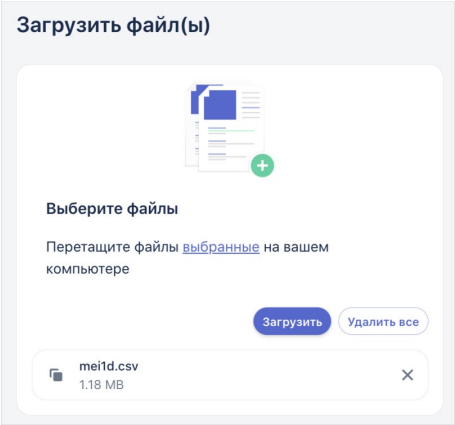


Рисунок 15.1.3 – Отображение выбранного файла

16.6. Нажмите на кнопку «Загрузить». Файл с входными данными отобразится в папке:

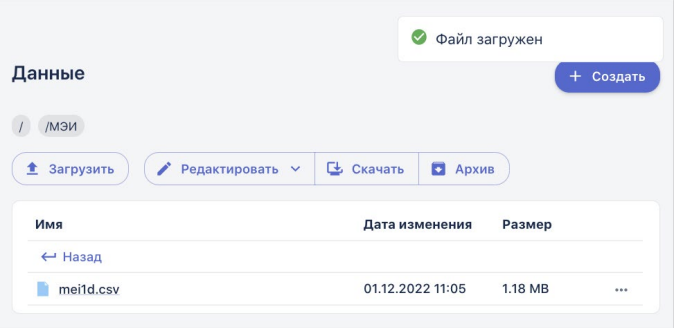


Рисунок 15.1.4 – Загрузка выбранного файла

17. **Создание новой рабочей области**

Полностью блок схема представлена в Таблице 16.7.

17.1. Перейдите в пункт меню системы **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка ⊕):

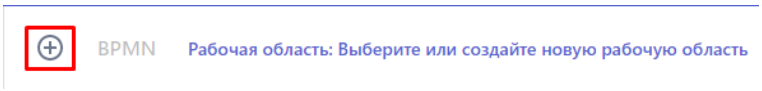


Рисунок 15.1.5 - Создание новой рабочей области

17.2. В открывшейся форме введите название новой рабочей области «МЭИ» и нажмите кнопку «Создать»:

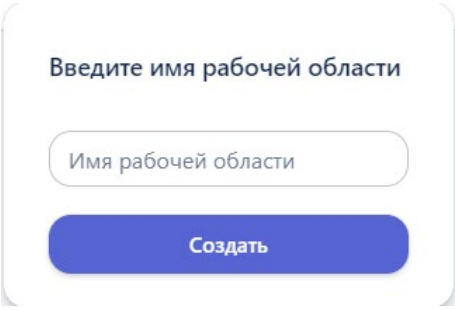


Рисунок 15.1.6 - Ввод имени рабочей области

17.3. На панели инструментов отобразится название созданной рабочей области.

18. **Добавление элемента «Запуск»:**

18.1. На панели инструментов блок-схемы нажмите кнопку «Добавить элемент» (кнопка BPMN)

18.2. В открывшейся библиотеке графических элементов выберите элемент «Запуск» :

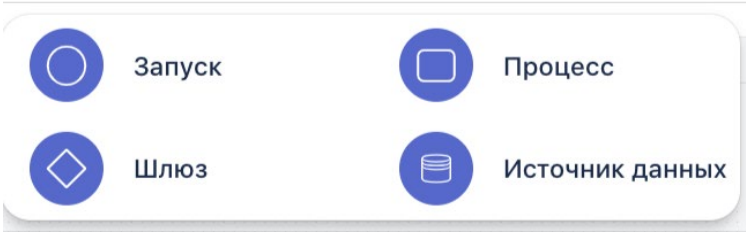


Рисунок 15.1.7 - Возможные элементы блок схемы

18.3. На рабочую область добавится элемент «Запуск»:

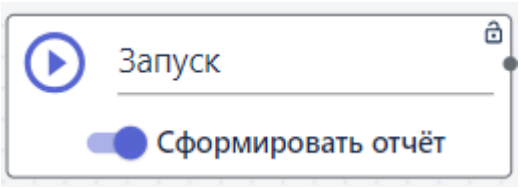


Рисунок 15.1.8 - Блок Запуск

19. **Добавление и настройка элемента «Источник данных»**.

19.1. Добавьте на рабочую область элемент «Источник данных»:

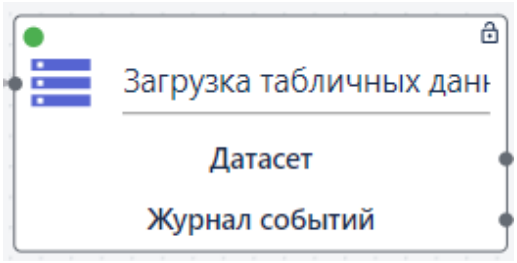


Рисунок 15.1.9 - Блок Источник данных

19.2. **Открытие настроек элемента**. На элементе «Источник данных» нажмите на значок . Справа откроется панель настроек элемента, где будут отображаться созданные в разделе папки и файлы с табличными данными.

19.3. **Выбор данных для загрузки в блок-схему**. Для того чтобы найти нужный файл, кликните на папку и перейдите в нее, выберите из списка файл, загруженный в Систему в шаге 1 «mei1d_dublicate_2.csv», нажмите на три точки в строке с названием файла и кликните «Выбрать». Внизу отобразится название выбранного файла:

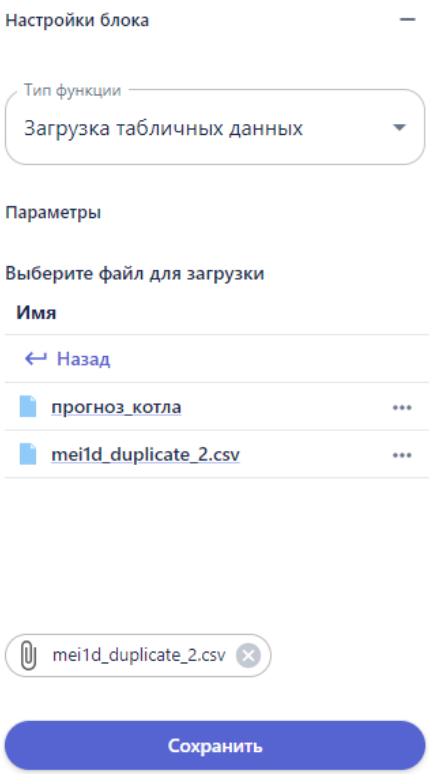
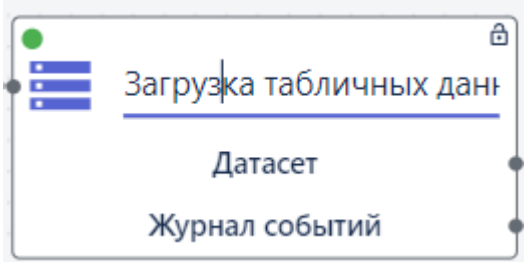


Рисунок 15.1.10 - Выбор файла

19.4. **Сохранение настроек элемента**. На панели настроек элемента нажмите на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

19.5. **Ввод названия элемента**. Чтобы задать название элемента нужно дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области, и ввести нужное название в поле с названием, доступным для редактирования:



19.6. **Установка соединений**. Соедините выходную точку элемента «Запуск» с входной точкой элемента «Источник данных» с помощью левой кнопка мыши:

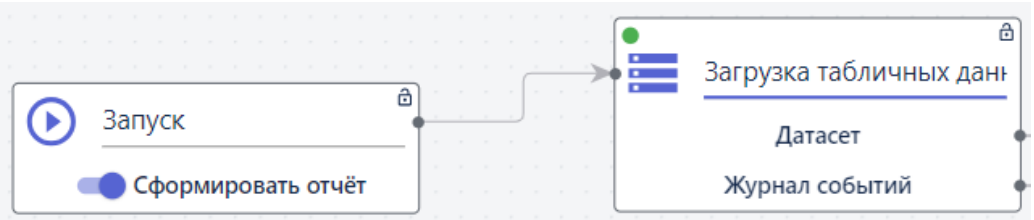


Рисунок 15.1.11 - Соединение элементов Запуск и Источник данных

20. **Добавление и настройка элемента «Процесс».** Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

20.1. На панели свойств элемента выбрать из списка функцию: тип функции «Загрузка данных» -> функция «Преобразование данных во временной ряд».

В разделе «Параметры» отобразятся поля:

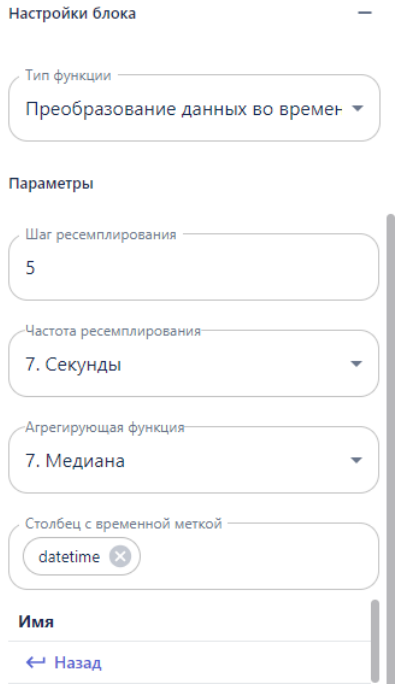


Рисунок 15.1.12 - Параметры функции «Преобразование данных во временной ряд»

20.2. В поле «Шаг ресемплирования» введите 5

20.3. В поле «Частота ресемплирования» выберите 7.Секунды

20.4. В поле «Агрегирующая функция» выберите 7.Медиана

20.5. В поле «Столбец с временной меткой» введите datetime

При помощи функции «Преобразование данных во временной ряд» можно привести данные к другой дискретности/периодичности, например в нашем случае - 5 секунд, а в качестве значения выбирается медиана - значение посередине.

20.6. На панели настроек элемента нажмите на кнопку «Сохранить».

20.7. Измените название элемента на «во временной ряд».

20.8. Соедините с элементы:

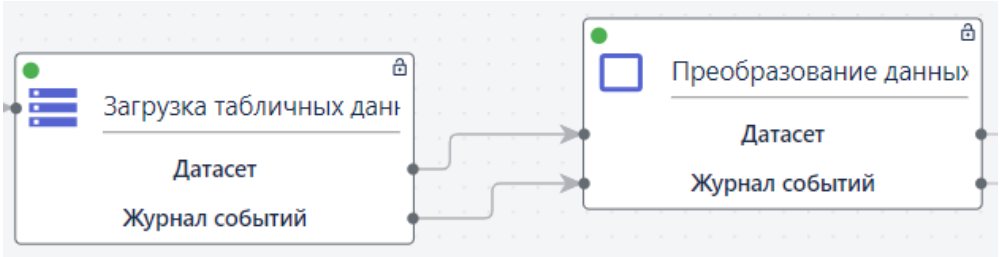


Рисунок 15.1.13 - Соединение элементов Источник данных и Преобразование во временной ряд

21. **Срез временного ряда.** Добавьте на рабочую область элемент «Процесс» и настроить его:

21.1. Выберите из списка функцию: раздел «Предобработка данных» -> функция «Срез временного ряда по индексу»:

21.2. В разделе «Параметры» в поле «Дата начала» введите значение *2020-12-16 11:00:00,* а в поле «Дата окончания» - *2020-12-16 15:00.*

21.3. Соедините элементы:

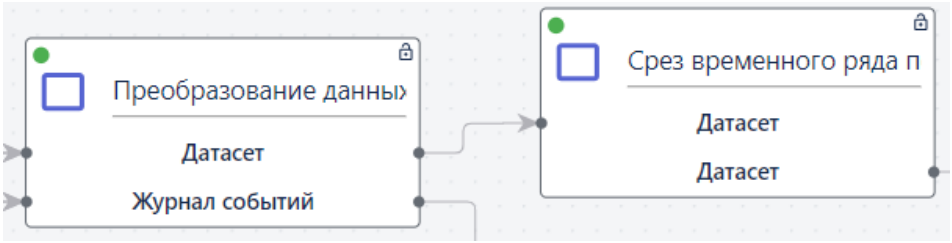


Рисунок 15.1.14 - Соединение элементов Преобразование во временной ряд и Срез временного ряда

22. **Выбор признаков и целевых показателей**. Добавьте на рабочую область и настройте элемент «Процесс»:

22.1. Выберите из списка функцию: раздел «Анализ данных» -> функция «Выбор признаков и целевых признаков»

22.2. В разделе «Параметры» -> в поле «Признаки» оставьте пустым, а в поле «Целевые признаки» вы можете либо поочередно ввести сначала признак Tq и нажать Enter, затем Tw и нажать Enter или вы можете найти в списке файлов mei1d.csv, нажать на три точки рядом с его названием и выбрать «Выгрузить признаки», тогда в поле автоматически подтянутся все признаки, который есть в файле. Вам необходимо удалить ненужные и оставить только Tq и Tw.

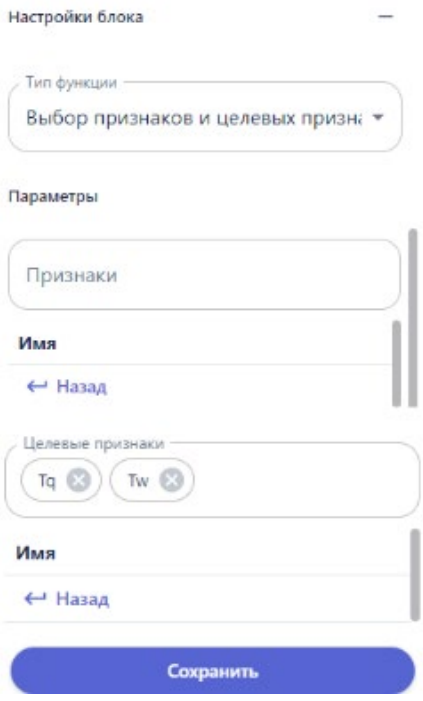


Рисунок 15.1.15 - Параметры функции «Выбор признаков и целевых признаков»

22.3. Соедините элементы:

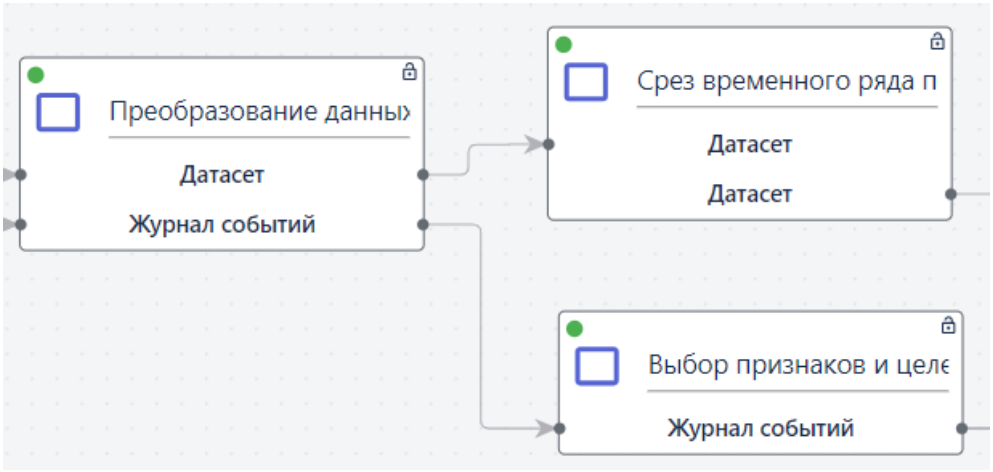


Рисунок 15.1.16 - Соединение элементов Преобразование данных во временной ряд Выбор признаков и целевых признаков

23. **Разделение датасета на обучающую и тестовую выборки**. Добавьте на рабочую область и настройте элемент «Процесс»:
    - 23.1. Выберите из списка функцию: раздел «Машинное обучение» -> функция «Разделение датасета на обучающую и тестовую выборки».
    - 23.2. Укажите долю тестовой выборки – 0.2. Так 80% данных будут использованы для обучения модели, и 20% – для тестирования.
    - 23.3. Не нужно ставить галочки в полях «Перемешивать наблюдения перед разделением» и «Разделять с учетом меток классов» (параметр активируется только для решения задач классификации)
    - 23.4. Измените название элемента на «Сплит».
    - 23.5. Соедините элементы:

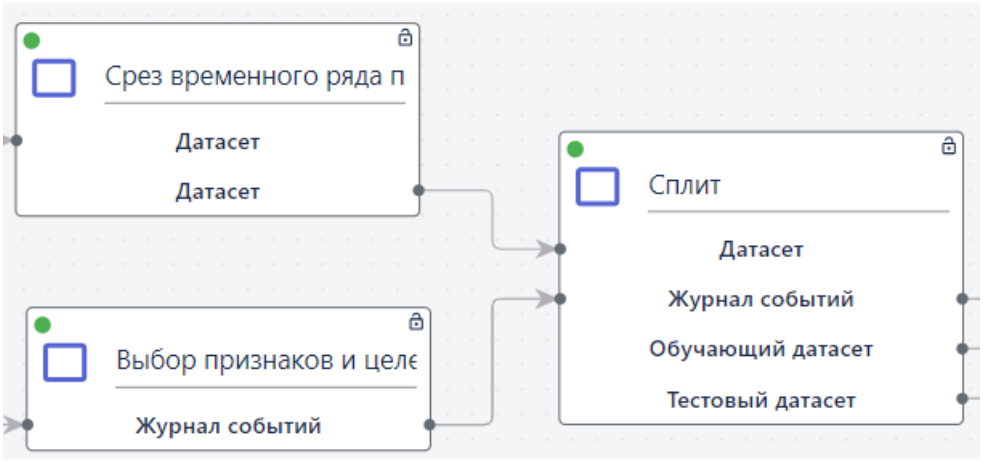


Рисунок 15.1.17 - Соединение элементов «Установить признаки» и «Срез временного ряда с блоком Сплит»

24. **Стабилизация дисперсии**. Добавьте на рабочую область и настройте элемент «Процесс»:
    - 24.1. Выберите из списка функцию: раздел «Препроцессинг» -> функция «Стабилизация дисперсии».
    - 24.2. В качестве метода выберите 2. yeo-johnson
    - 24.3. Установите галочку в поле «Замена значений столбцов».
    - 24.4. Установите галочку в поле «Стандартизация».
    - 24.5. В поле «Флаг признака» выбрать из списка значение «1. Признаки»

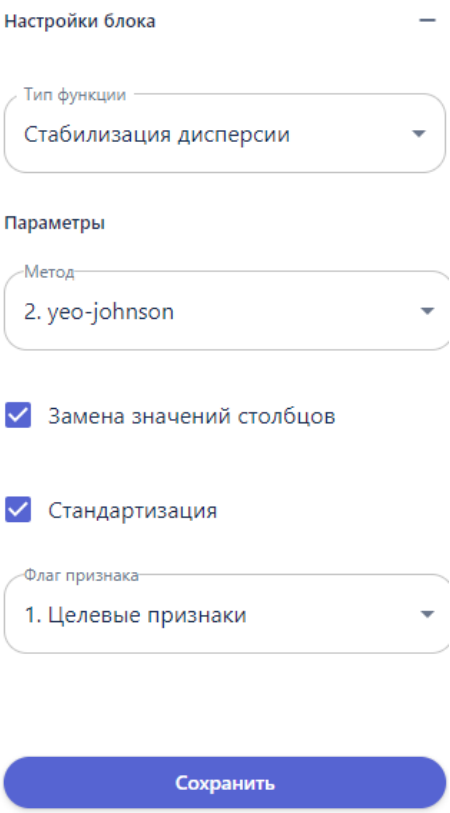


Рисунок 15.1.18 - Параметры функции «Стабилизация дисперсии»

24.6. Соедините элементы:

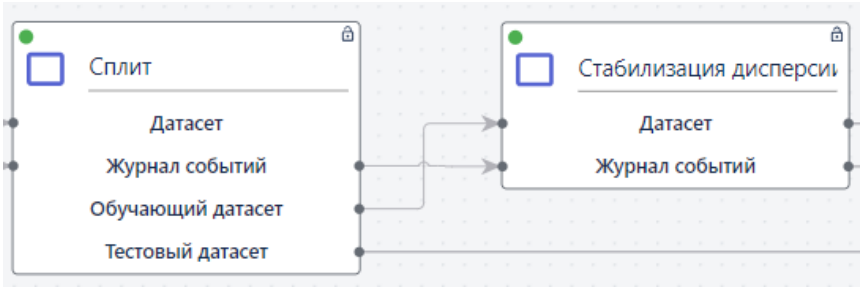


Рисунок 15.1.19 - Соединение элементов Сплит и Стабилизация дисперсии

25. **Skew и Дики-Фуллер для Стабилизации.** Данный этап не является обязательным, но позволяет оценить стал ли ряд стационарным после всех преобразований.
    25.1. Добавьте на рабочую область и настроить два элемента «Процесс»:
    25.2. Для первого блока в карточке элемента выберите из списка функцию: раздел «Тест на нормальность распределения» -> функция «Коэффициент асимметрии Skewness». В поле признаки введите ['Tq', 'Tw'] и сохраните.
    25.3. Для второго блока в карточке элемента выберите из списка функцию: «Тест на стационарность временного ряда» -> функция «Тест Дики-Фуллера». В поле «пороговое значение alpha» введите [0.05].
    25.4. Соедините элементы:

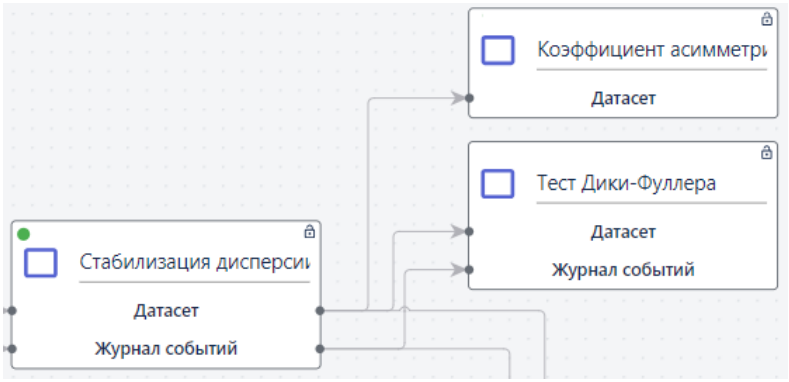


Рисунок 15.1.20 - Соединение элементов Стабилизация дисперсии с Коэффициент асиметрии Skew и Тест Дики-Фуллера

26. **Дифференцирование временного ряда.** Добавьте на рабочую область и настройте элемент «Процесс»:
    26.1. Выберите из списка функцию: раздел «Препроцессинг» -> функция «Дифференцирование временного ряда».
    26.2. В поле «Шаг дифференцирования введите: [1,1] (вместе с квадратными скобками). Шаг сдвига - параметр, который определяет, на сколько наблюдений мы сдвигаем временной ряд, чтобы получить разницу в значениях.
    26.3. Соедините элементы:

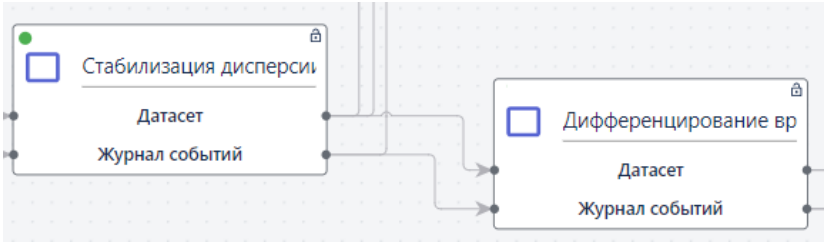


Рисунок 15.1.21 - Соединение элементов Стабилизация дисперсии и Дифференцирование временного ряда

27. Для блока «Дифференцирование временного ряда» опционально можно провести **тест Дики-Фуллера**. Для этого необходимо:
    27.1. Добавить на рабочую область и настроить элемент «Процесс» и выбрать из списка функцию: «Тест на стационарность временного ряда» -> функция «Тест Дики-Фуллера». В поле «пороговое значение alpha» вводим [0.05].
    27.2. Соединить элементы:

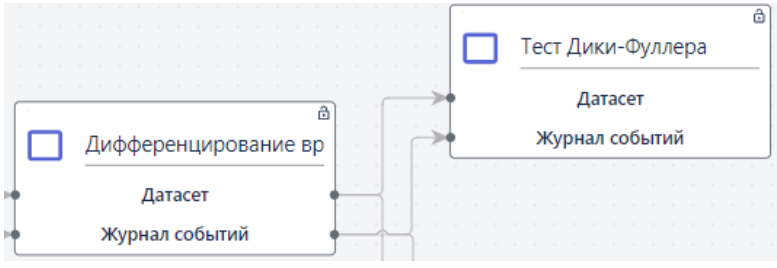


Рисунок 15.1.22 - Соединение элементов Дифференцирование временного ряда и тест Дики-Фуллера

28. **Создание признаков для временного ряда.** Добавьте на рабочую область и настройте элемент «Процесс»:

28.1. Выберите из списка функцию: раздел «Препроцессинг» -> функция «Создание признаков для временного ряда».

28.2. В поле «Максимальное количество лагов» укажите: [1].

28.3. Соедините элементы:

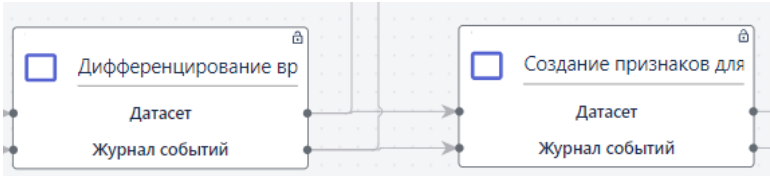


Рисунок 15.1.23 - Соединение элементов Дифференцирование и Создание признаков для временного ряда

29. **Метод k-ближайших соседей для регрессии**. Добавьте на рабочую область и настройте элемент «Процесс»:

29.1. Выберите из списка функцию: раздел «Регрессия» -> функция «Метод k-ближайших соседей для регрессии».

29.2. В поле «Количество ближайших соседей» укажите [2,3,5,10] (вместе с квадратными скобками).

29.3. Установите галочку в поле «Оптимизация гиперпараметров»

29.4. В поле «Тип веса для соседей» выберите и «Единый» и «По расстоянию»

29.5. В поле «Метрика расстояния» выберите и «Евклидово», «Косинусное» и «Манхэттенское»

29.6. В поле «Метрика для оптимизации» выберите и «1. RMSE» и в поле «Количество фолдов для оптимизации» указать [3]

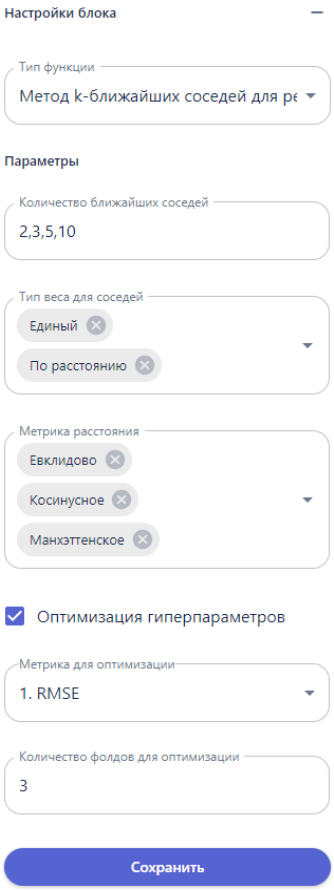


Рисунок 15.1.24 - Параметры функции «Метод k-ближайших соседей для регрессии»

29.7. Соедините элементы:

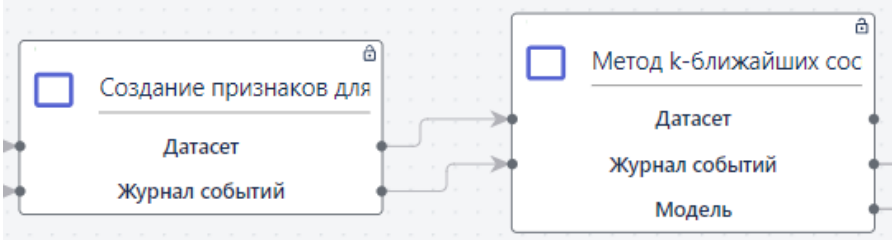


Рисунок 15.1.25 - Соединение элементов для Создание признаков для временного ряда и Метод k-ближайших соседей

30. **Валидация модели**. Добавьте на рабочую область и настройте элемент «Процесс»:

30.1. Выберите из списка функцию: раздел «Машинное обучение» -> функция «Валидация модели»:

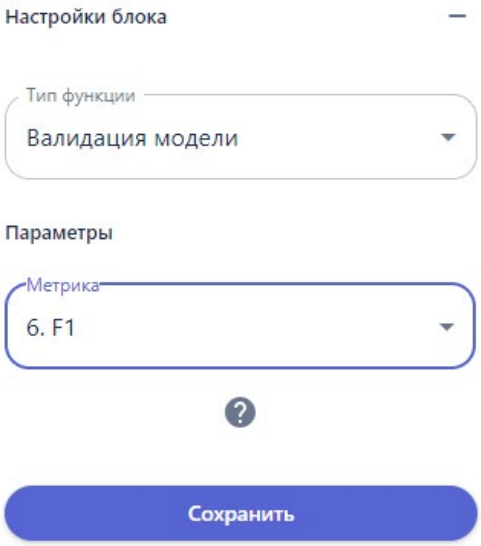


Рисунок 15.1.26 - Параметры функции «Валидация модели»

30.2. В разделе «Параметры» -> в поле «Метрика» выберите значение «6. F1» – метрика для валидации. Анализируется связь между выбранными погодными условиями и целевым признаком. Так оценивается вероятность возникновения лесного пожара по всем показателям. Функция возвращает величину вероятности в виде значения от 0 до 1.

30.3. Установите соединения с предыдущим элементом «Классификация» и элементом «Сплит датасета», как показано в **Приложении 7**.

Обратите внимание! Элементы в блок-схеме могут соединяться не только последовательно.

31. **Сохранение модели**. Добавьте на рабочую область и настройте элемент «Процесс»:

31.1. В карточке элемента выберите из списка функцию: раздел «Управление моделями» -> функция «Сохранение модели».

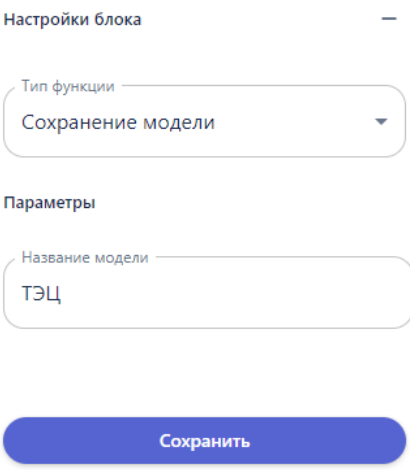


Рисунок 15.1.27 - Параметры функции «Сохранение модели»

31.2. В разделе «Параметры» введите название модели, с которым она будет сохранена, например «ТЭЦ».

31.3. Измените название элемента на «Сохранение модели».

31.4. Соедините элементы:

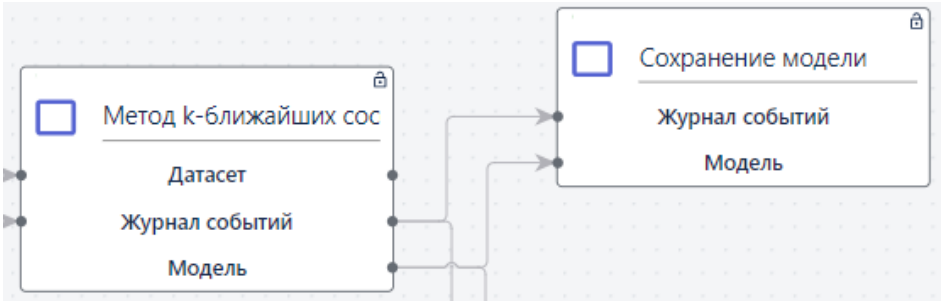


Рисунок 15.1.28 - Соединение элементов Метод k-ближайших соседей и Сохранение модели

32. **Запуск пайплайна**. Чтобы запустить сборку пайплайна нажмите на кнопку на первом элементе «Запуск» собранной блок-схемы. При этом отображение элемента «Запуск» изменится и появится опция «Сформировать отчет»:

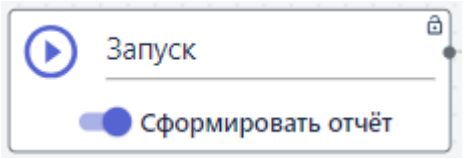
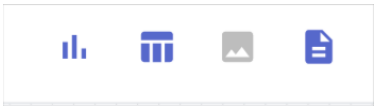


Рисунок 15.1.29 - Запуск пайплайна

Если активировать параметр «Сформировать отчет», в результате запуска пайплайна будет создан отчет.

33. **Визуализация результатов**. После того как все элементы схемы будут успешно обработаны, на панели инструментов появляются кнопки:



15.2 Создание блок схемы для работы с данными в режиме реального времени

Платформа BASIS AI позволяет обрабатывать не только статичную информацию, но также данные, получаемые в режиме реального времени. Возможность прогнозировать значения признаков и линий их трендов создает базу для оперативного и своевременного принятия решений, позволяющих избежать критических ситуаций. Платформа BASIS AI также позволяет создавать визуализации данных в реальном времени при помощи различных графиков, что делает информацию более наглядной и обеспечивает широту ее применения.

В данном разделе будет рассмотрен пример работы с платформой для прогнозирования значений в режиме реального времени. Для выполнения этого сценария, необходимо построить и запустить пайплайн **Обучение модели прогнозирования температуры воды и газов в котле** для обучения модели ИИ на аналогичных данных в виде статичной таблицы. Вторым этапом созданная модель применяется для обработки данных и моментального прогнозирования значений.

Т.о. второй этап работы с данными в режиме реального времени заключается в настройке коннектора и построении второго пайплайна, который будет получать информацию из соединения и обрабатывать её с помощью ранее созданной и обученной модели. В рамках примера также будет показано, как создавать и интерпретировать графики, на которых отображается информация в режиме реального времени.

1. **Создание коннектора.**
   1.1. Перейдите в пункт меню «Соединения». Перейти в пункт меню «Соединения»:

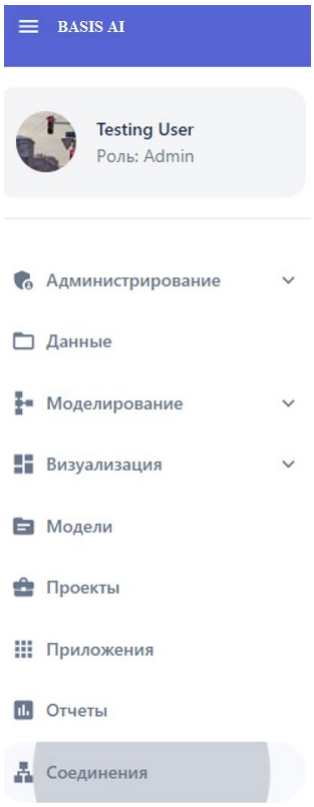


Рисунок 15.2.1 - Пункт меню Соединения

Откроется страница «Соединения» на первой вкладке «Источники данных», на которой отображаются все ранее созданные источники:

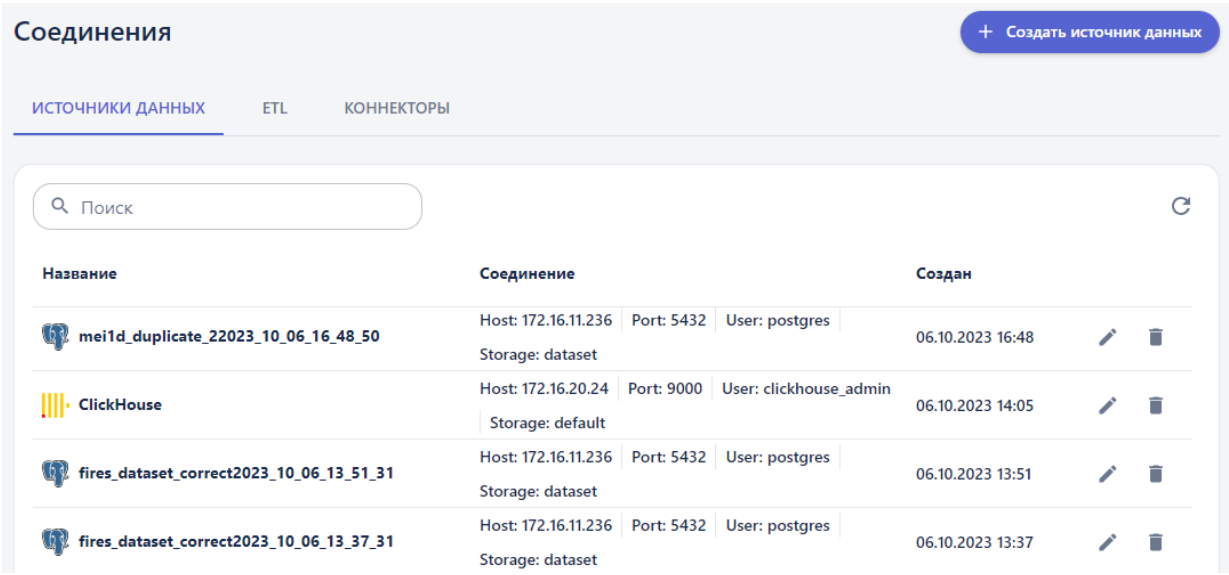


Рисунок 15.2.2 - Список созданных источников данных

1.2. Для создания нового источника, нажмите кнопку «Создать источник данных» в верхнем правом углу. Откроется окно «Создание нового ресурса данных»:

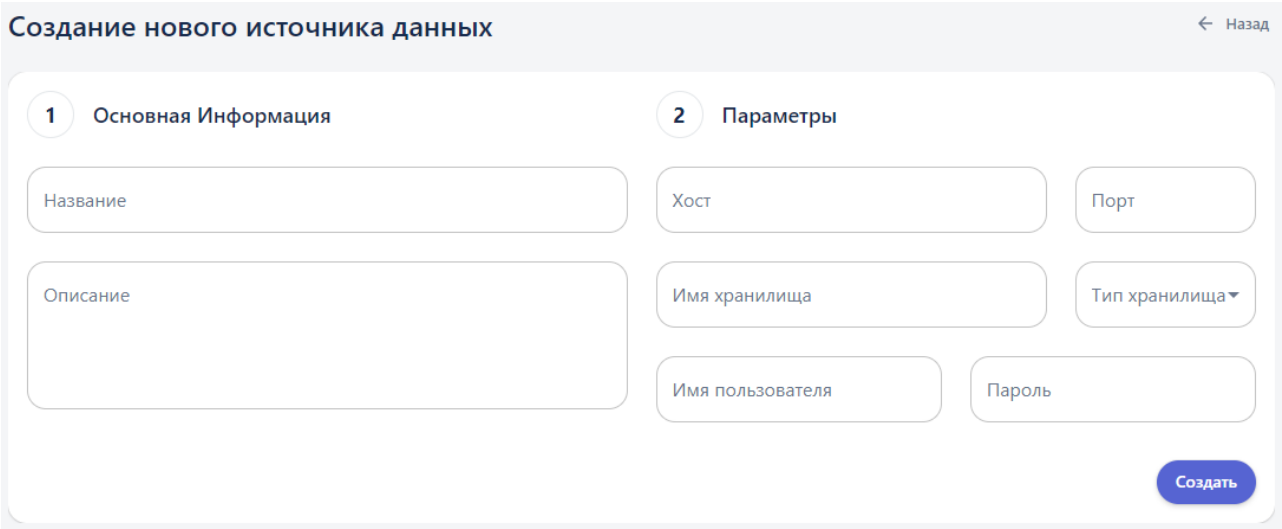


Рисунок 15.2.3 - Создание нового источника данных

1.3. Заполните поля следующим образом:

- ➔ *Название*. Задайте название источника «postgres dataset 116».
- ➔ *Хост*. Указывается хост протокола TCP/IP, т.е IP-адрес подключаемой БД, например: «172.16.11.116».
- ➔ *Порт*. Номер порта, по которому устанавливается соединение с сервером, на котором установлена БД **posgresql**. Указать «9999».
- ➔ *Имя хранилища*. Название базы данных, которое указано на подключаемом сервере. Указать «dataset».
- ➔ *Тип хранилища*. Из выпадающего списка выберите тип «postgresql»:
- ➔ *Имя пользователя*, *пароль*. Параметры учетной записи администратора внешнего сервера для разрешения доступа к данным. Указать пользователя «postgres», и пароль «example».

➔ *Описание*. Вводится дополнительная информация по источнику, необязательное поле.

➔ Для регистрации в Системе источника нажмите кнопку «Создать».

1.4. Созданный коннектор отобразится в общем списке. При необходимости изменить настройки используйте кнопку «Редактировать»:

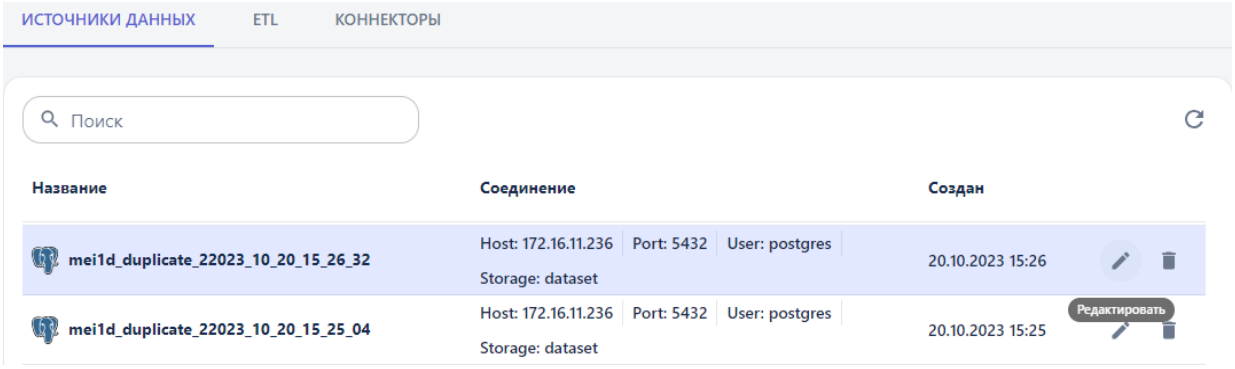


Рисунок 15.2.4 - Отображение созданного источника в списке

1.5. **Создание ETL.** Создание нового ETL осуществляется на вкладке «ETL» в разделе Соединения:

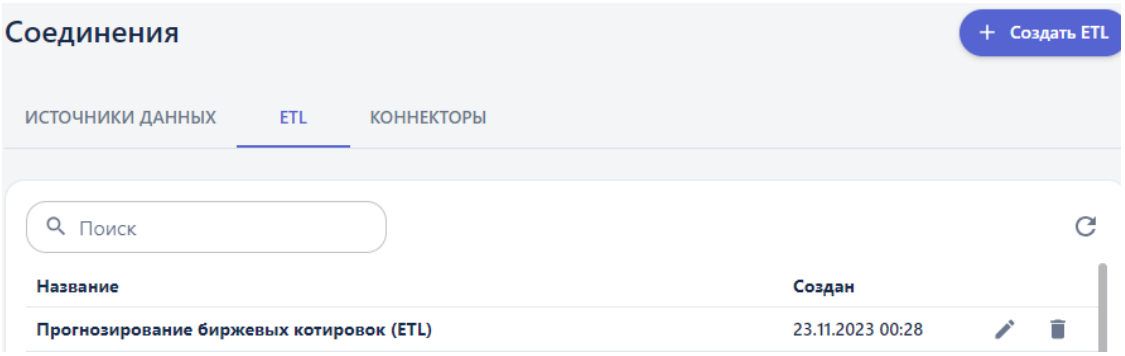


Рисунок 15.2.5 - Список созданных ETL

1.5.1. Нажмите кнопку «Создать ETL». Откроется окно «Создать новый ETL»:

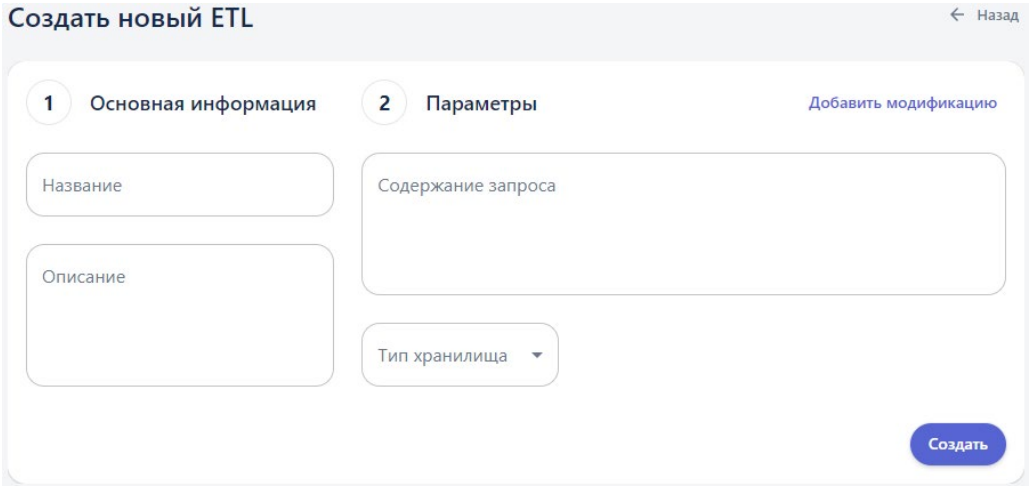


Рисунок 15.2.6 - Создание нового ETL

1.5.2. Заполните поля следующим образом:

➔ *Название*. Пользователь вручную задает название ETL «mei1d_duplicate_2» – запроса на извлечение данных.

➔ *Содержание запроса*. Прописывается sql запрос для извлечения данных из внешнего сервера: select "Tq", "Tw" from mei1d_duplicate_2

➔ *Тип хранилища*. Выбирается тип «postgresql».

➔ Нажмите на кнопку «Создать».

1.5.3. Созданный ETL можно менять и редактировать по аналогии с источником данных.

1.6. **Создание Коннектора.** Перейдите на вкладку «Коннекторы»:

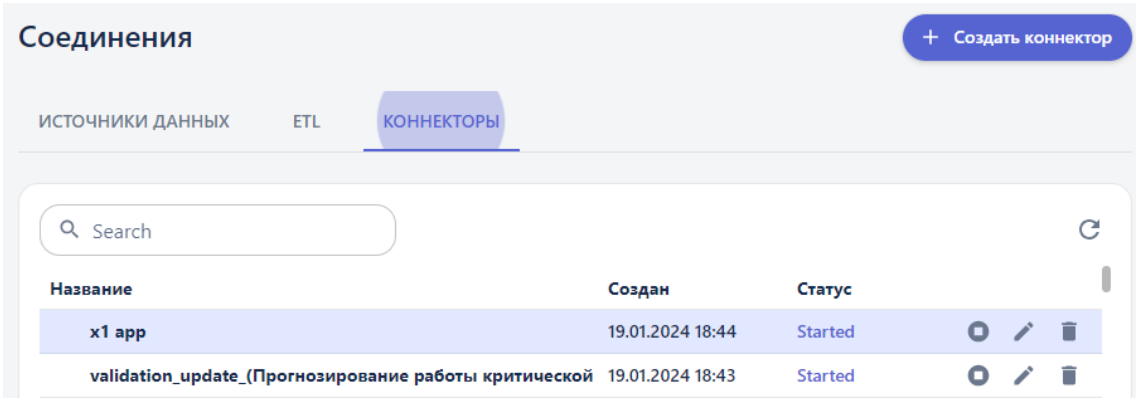


Рисунок 15.2.7 - Список созданных коннекторов

1.6.1. Нажмите на кнопку «Создать коннектор». Откроется окно «Создать новый коннектор»:

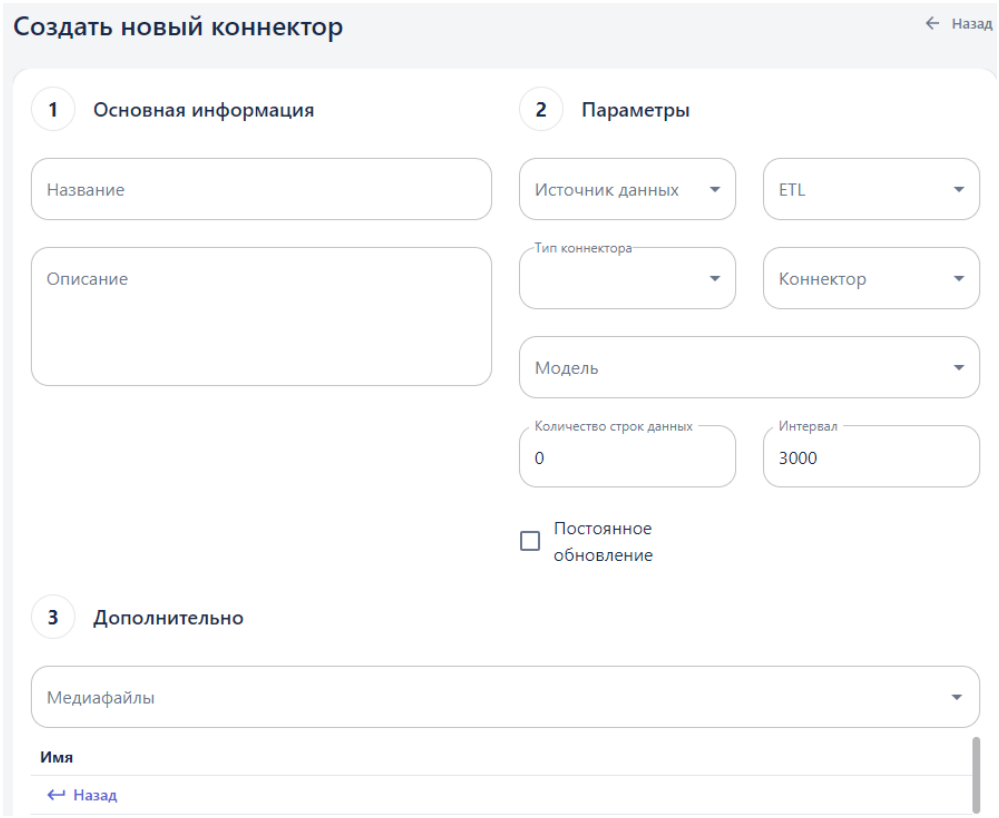


Рисунок 15.2.8 - Создание нового коннектора

1.6.2. Заполните поля:

➔ *Название*. Пользователь вручную задает название создаваемого коннектора, например «mei1d_duplicate_2».

➔ *Ресурсы данных*. Из списка выбирается источник «postgres dataset 116», созданный в шаге 1

➔ *ETL*. Из списка выбирается ETL «mei1d_duplicate_2», созданный в шаге 2.

➔ *Описание*.

1.6.3. Нажмите кнопку «Создать».

1.6.4. Сразу после создания коннектору присваивается статус «Stopped»:

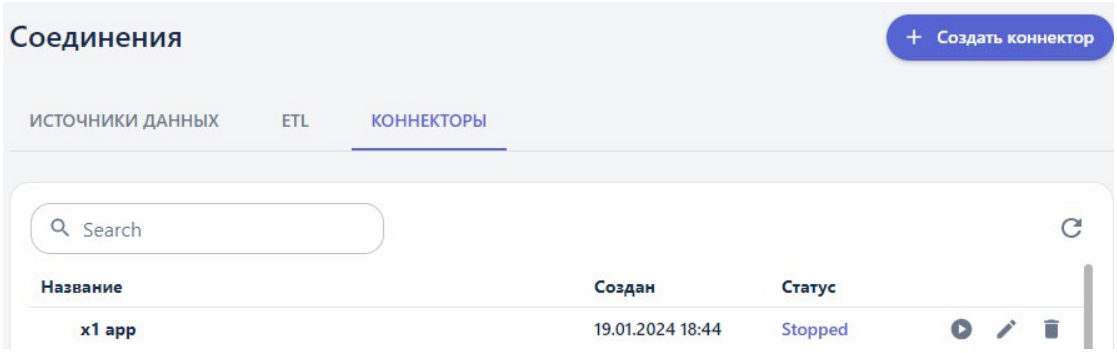


Рисунок 15.2.9 - Отображение нового коннектора в списке

1.7. **Запуск коннектора**. Чтобы данные из источника начали поступать в Систему необходимо запустить коннектор. Для этого нажмите на кнопку « » в строке с коннектором. В результате статус меняется на значение «Started». Теперь коннектор можно использовать в качество источника данных при построении пайплайна.

2. Создание блок схемы.

Следующим этапом мы переходим к созданию второго пайплайна, где будет использована ранее обученная модель и созданный коннектор. Т.о. будет происходить обработка данных, получаемых в режиме реального время для получения прогноза температуры воды и котла.

2.1. **Создание новой рабочей области.**

2.1.1. Перейдите в пункт меню системы **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка)

2.1.2. В открывшейся форме введите название новой рабочей области «МЭИ realtime» и нажмите кнопку «Создать»:

2.1.3. На панели инструментов отобразится название созданной рабочей области.

2.2. **Добавление первого элемента «Источник данных»**. В данном примере мы будем использовать два источника данных: коннектор и модель. Для загрузки в пайплайн данных из коннектора добавьте элемент «Источник данных» на рабочую область и настройте элемент:

2.2.1. На элементе нажмите на кнопку. Откроется панель настроек элемента.

2.2.2. На панели настроек элемента выберите из списка функцию: раздел «Загрузка данных» -> функция «Загрузка табличных данных из коннектора».

2.2.3. В поле «Выберите файл» из списка всех созданных коннекторов выберите «mei1d_duplicate_2».

2.2.4. Установите галочку в поле «Онлайн данные»:

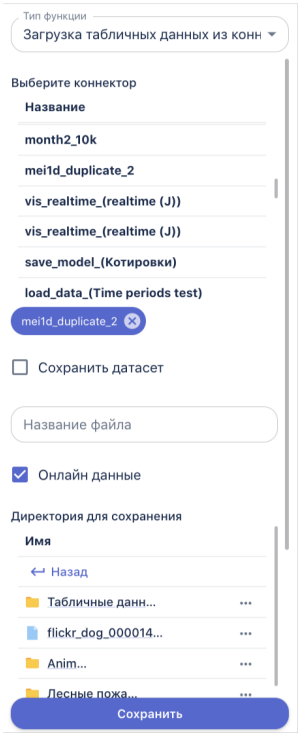


Рисунок 15.2.10 - Выбор файла для загрузки

2.2.5. Нажмите на кнопку «Сохранить».

2.2.6. Переименуйте блок в «Данные МЭИ»

2.2.7. Соедините элементы:

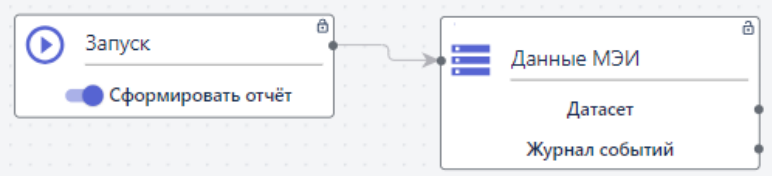


Рисунок 15.2.11 - Соединение элементов Запуск и Данные МЭИ

Полностью схема пайплайна представлена в Таблице 16.8

2.3. **Добавление второго элемента «Источник данных»**. Для того чтобы включить в пайплайн ранее созданную обученную модель, нужно добавить еще один элемент «Источник данных» на рабочую область (кнопка ). Чтобы настроить элемент:

2.3.1. На элементе нажмите на кнопку  . Откроется панель настроек элемента.

2.3.2. На панели настроек элемента выберите из списка функцию: раздел «Машинное обучение» -> функция «Загрузка модели».

2.3.3. В списке моделей выберите «mei»

2.3.4. Переименуйте блок в «Модель»

2.4. **Запись в датасет логирования.** В данном блоке будет осуществляться логирование новой поступающей новой информации в датасет, для этого на рабочую область добавляется элемент «Процесс»:

2.4.1. В карточке элемента выберите из списка функцию: раздел «Анализ данных» -> функция «Запись в датасет логирования».

2.4.2. Переименуйте блок в «Логирование»

2.4.3. Соедините элементы:

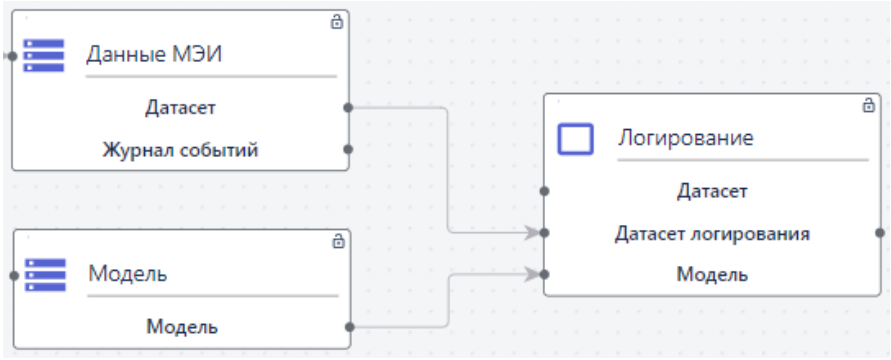


Рисунок 15.2.12 - Соединение элементов «Данные МЭИ», «Модель» и «Логирование»

2.5. **Прогноз.** Добавьте на рабочую область и настройте элемент «Процесс»:

2.5.1. В карточке элемента выбрать из списка функцию: раздел «Машинное обучение» -> функция «Прогноз модели».

2.5.2. Переименуйте блок в «Прогноз»

2.5.3. Соедините элементы:

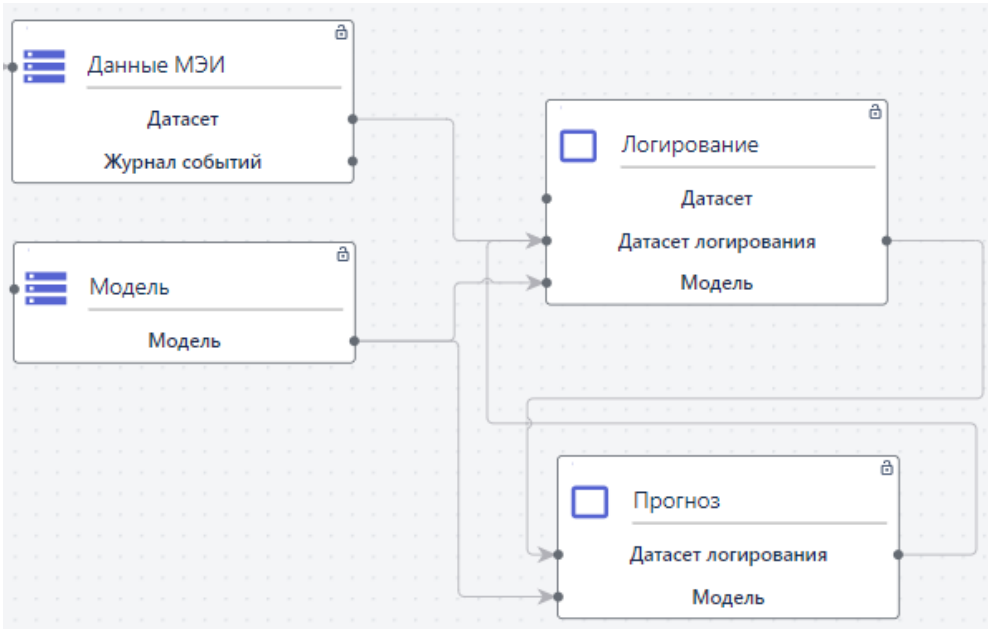


Рисунок 15.2.13 - Соединение элементов «Модель», «Логирование» и «Прогноз»

2.6. **Визуализация.** Добавьте на рабочую область и настройте элемент «Процесс»:

2.6.1. В карточке элемента выберите из списка функцию: раздел «Анализ данных» -> функция «Визуализация Real-Time».

2.6.2. В параметрах блока выберите все графики и установите для них следующие параметры:

- Линейный график: Число периодов в окне: 10; Период окна: 5.Минуты
- Свечной график: Число периодов: 1; Период: 5.Минуты; Число периодов в окне: 10; Период окна: 5.Минуты

2.6.3. Переименуйте блок в «Визуализация»

2.6.4. Соедините элементы:

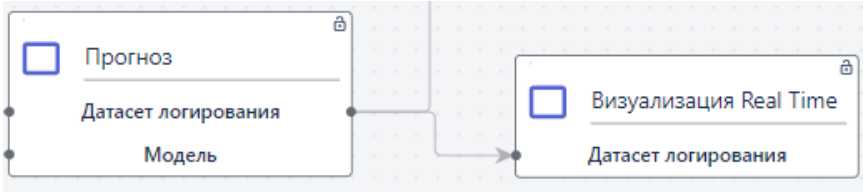


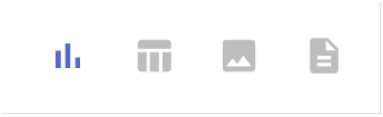
Рисунок 15.2.14 - Соединение элементов «Прогноз» и «Визуализация»

2.7. **Запуск блок-схемы**. Для запуска блок-схемы нажмите на кнопку на первом элементе «Запуск». Все элементы блок-схемы должны отработать с зелеными индикаторами.

При запуске real time блок схем элементы будут отрабатывать снова и снова и индикаторы на пайплайне будут постоянно менять цвета. Если необходимо остановить обработку данных - нажмите кнопку запуск повторно.

3. Визуализация результатов прогнозирования температуры на графиках

1. После того как все элементы схемы будут успешно обработаны, на панели инструментов активизируется кнопка «Графики»:



Графики будут доступны в зависимости от выбранных в рамках блока «Визуализация». В текущем примере это Линейный и Свечной графики:

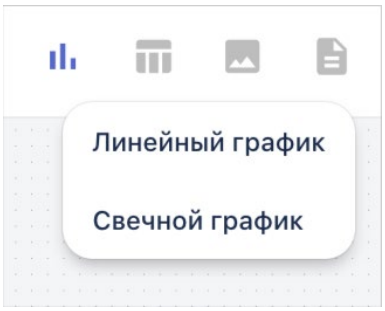


Рисунок 15.2.15 - Список доступных графиков для отображения на рабочей области

Для отображения графиков на рабочей области нужно выбрать необходимые, кликнув на их названия.

Обратите внимание, что блок схема работает в режиме реального времени и данные на графиках будут постоянно обновляться в зависимости от заданного лага в графиках.

1.1. Линейный график:

Рисунок 15.2.16 - Линейный график временного ряда в режиме реального времени

График показывает изменения значений целевых признаков (Tq и Tw) в течение заданного периода времени, который устанавливается в параметрах блока «Визуализация». В нашем примере мы выбрали период равный 10 минутам:

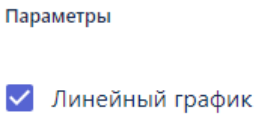


Рисунок 15.2.17 - Параметры настройки линейного графика

Соответственно на графике отображаются все значения признаков за последние 10 минут. Каждый раз, когда блок схема будет отрабатывать - временной отрезок ниже будет сдвигаться вперед, но значения всегда будут в пределах 10 минут. Т.к. на пайплайне присутствует блок прогнозирования, на линейном графике мы также видим прогноз значений параметров. Горизонт планирования равен шагу ресемплирования из пайплайна обучения, это значение записывается моделью.

1.2. Свечной график:

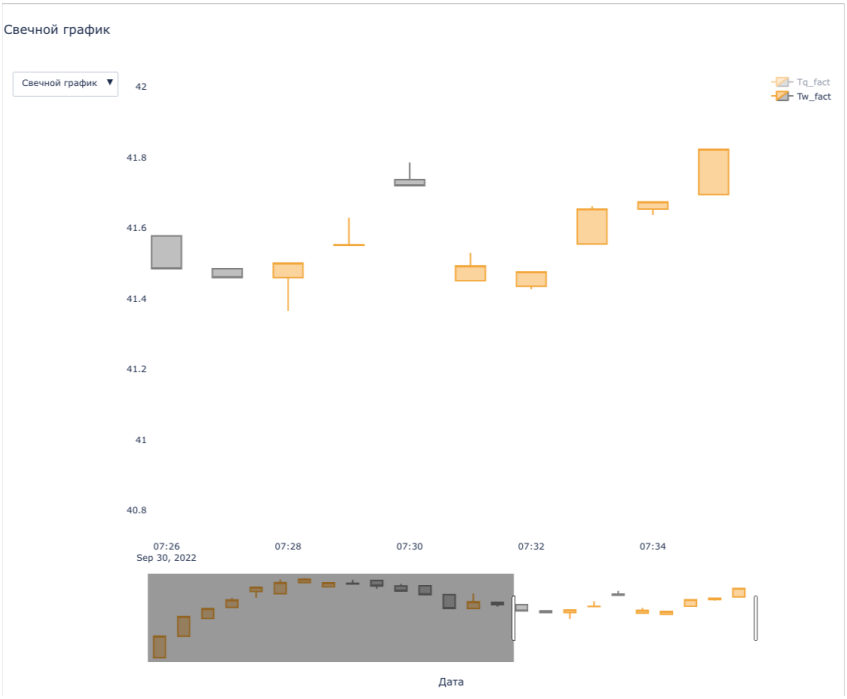


Рисунок 15.2.18 - Свечной график в режиме реального времени

График также показывает изменения значений параметров в течение заданного периода времени. При этом задается период, в рамках которого будет сформирована т.н. свеча (столбик на графике); и число периодов в окне. В нашем примере - это одна минута, и мы можем видеть 10 свечей на графике, т.к. задали число периодов в окне равное 10 минутам:

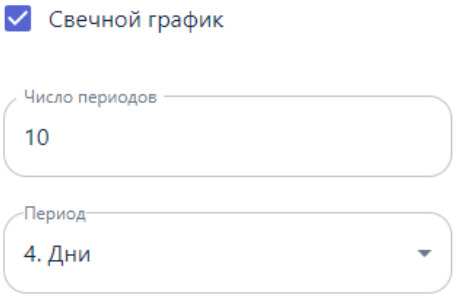


Рисунок 15.2.19 - Параметры настройки свечного графика

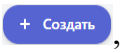
На горизонтальной оси отображается время, для которого было зафиксировано значение. На вертикальной - само значение признака. Прямоугольники (свечи) на графике отображают разницу между значением параметра на начало периода и на конец. Линии, исходящие из свечей, показывают максимальное и минимальное значение параметра за период времени. Если значение параметра на конец периода выше, чем на начало – то свеча окрасится в оранжевый цвет; если значение на конец периода ниже, чем на начало – в серый.

Созданные графики можно использовать для создания индивидуальных дашбордов.

15.3 Классификация изображений

Классификация изображений позволяет отнести изображение к определенному классу. Например, из набора изображений определить какое изображение относится к классу “Кошка” или “Собака”. Сначала в системе создаются папки, которые будут далее пополнены изображениями, классификации которых будет обучаться модель ИИ. Для загрузки в систему могут быть использованы файлы следующих форматов: 'jpg', 'jpeg', 'png'. Для каждого класса будет создана отдельная папка, данных из которой будут использованы при построении пайплайна для обучения модели опознавать определенные объекты на изображении.

15.3.1 Загрузка изображений

1. Первым шагом откройте раздел «Данные»
2. Для создания новой папки нажмите 

, после чего откроется *форма создания нового типа данных*:

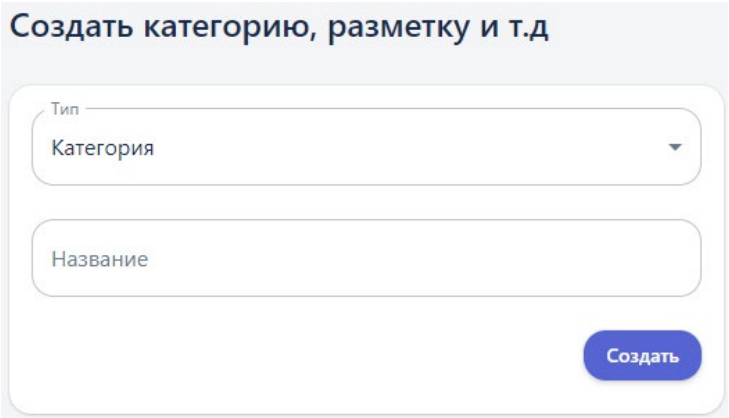


Рисунок 15.3.1 – Создание новой папки для классификации

3. В открывшейся в поле «Тип» выберите значение «Категория», в поле «Название» введите название новой группы, например «Данные для классификации изображений», и нажмите кнопку «Создать». **Примечание** – *Здесь под категорией имеется в виду папка, в которую будут складываться данные для решения задачи.*
4. Перейдите в созданную папку «Данные для классификации изображений» и создайте две новые папки внутри – «Animals Train» и «Animals Test». В группу «Animals Train» будут складываться данные для обучения будущей модели машинного обучения, а в группу «Animals Test» – данные для валидации или проверки ‘качества’ уже обученной модели. При этом для обучения модели необходимо использовать большее количество файлов, в нашем примере пропорция составляет 4 к 1.
5. В папке «Animals Train» создайте еще две папки, которые и будут определять классы, – «Dogs» и «Cats». Количество классов равно двум, так как в данном сценарии решается задача *бинарной классификации* (для многоклассовой классификации создавалось бы больше двух классов). В класс «Dogs» загружаются изображения собак, а в класс «Cats» загружаются изображения кошек.
6. Для того чтобы загрузить файлы, перейдите в нужную папку и нажмите кнопку «Загрузить»:

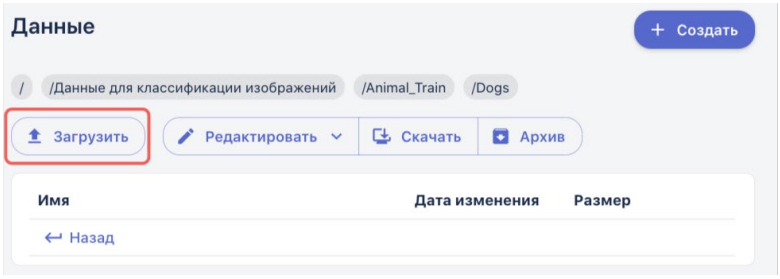


Рисунок 15.3.2 – Кнопка загрузки файлов для классификации

7. Откроется окно загрузки файлов:

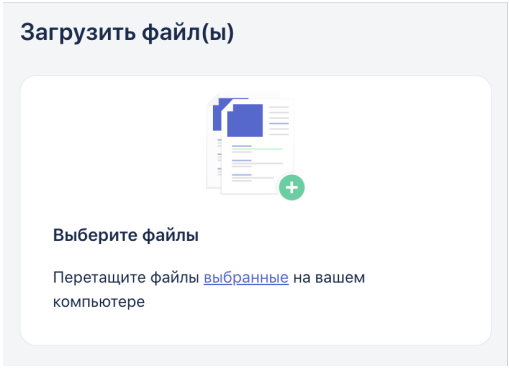


Рисунок 15.3.3 – Окно выбора файлов

8. Для выбора файлов кликните в область окна «Выберите файлы» или перетащите их по технологии drag and drop (из окна папки на вашем ПК в окно браузера).

   Обратите внимание: за раз можно добавить максимум 10 файлов. Соответственно, если нужно загрузить больше файлов, нужно повторить выбор несколько раз.

   После того, как все файлы выбраны, при необходимости вы можете удалить ненужные файлы, нажав на крестик в правой части строки с файлом, или нажать «Удалить все», если это требуется.

9. Когда все файлы выбраны и готовы к загрузке, нажмите на кнопку «Загрузить»:

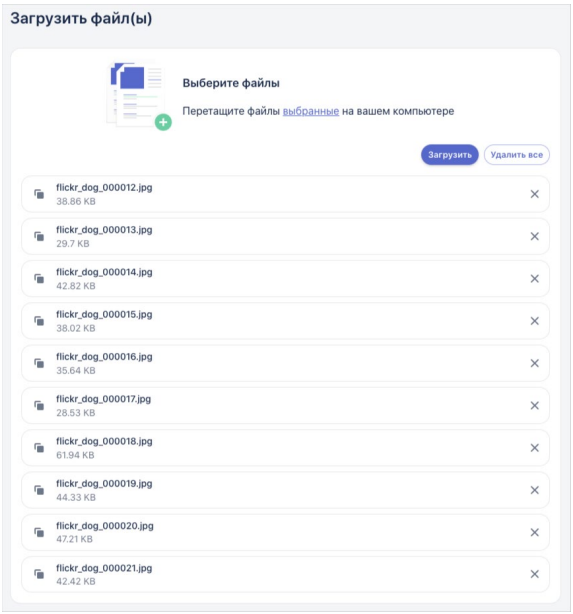


Рисунок 15.3.4 – Список выбранных файлов

10. В результате загруженные файлы отобразятся в папке:

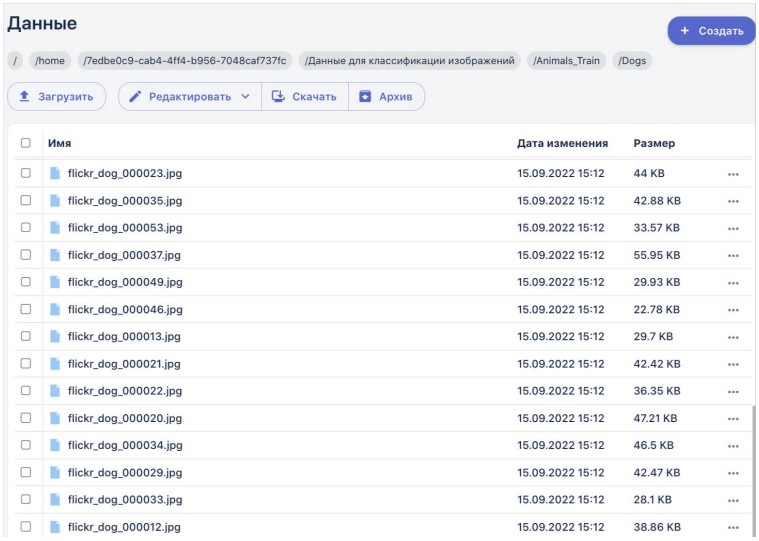


Рисунок 15.3.4 – Загруженные файлы в папке

11. Вышеописанные действия повторяются для папки «Animals Train» -> «Cats».
12. Далее по аналогии создаются и заполняются папки «Animals Test» -> «Dogs» и «Cats», туда загружаются файлы для валидации модели.
13. Чтобы удалить группу/класс достаточно удалить соответствующую папку в разделе Данные, нажав на три точки в строке с этой папкой.
14. После того, как обучающая и валидационная выборки собраны, для папок «Animals Test» и «Animals Train» добавляется параметр классификация. Для этого в строке с папкой нажмите на три точки и кликнуть на кнопку «Классификация», после этого содержимое папки будет готово для использования при построении модели:

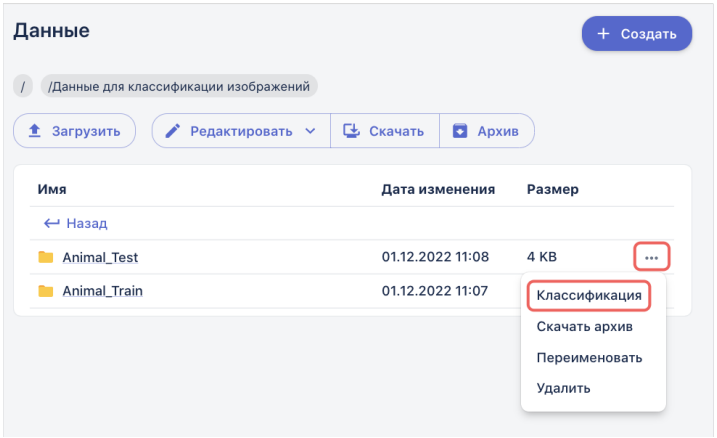


Рисунок 15.3.4 – Кнопка присвоения параметра Классификации папке

Обратите внимание: данное действие необходимо выполнить один раз. Даже если позже в папку будут добавлены новые файлы, они будут учтены при построении или запуске модели классификации.

15.3.2 Создание моделей классификаций

На левой рабочей панели выберите Моделирование-> Рабочая область. Нажмите  , чтобы добавить новую модель (например, “Animals”) и нажмите кнопку «Создать» (см. рис.

13.3.3). Чтобы воспользоваться существующими моделями выберете Моделирование-> Сохраненная рабочие области. Выберем готовую блок-схему из модели “Animals”. Она состоит из 4 блоков: “Загрузка изображений”, “Классификация изображений”, “Валидация модели классификации” и “Сохранение модели классификации” (см. рис. 13.3.4 или приложение 6).

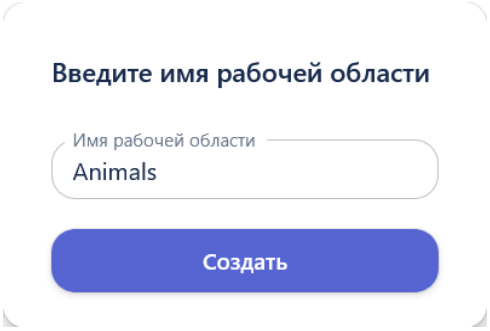


Рисунок 15.3.5 - Модель классификации изображений

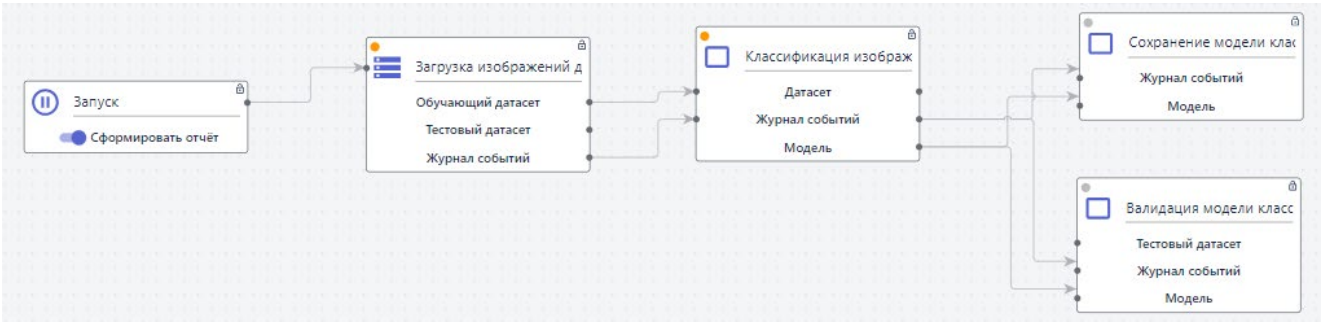


Рисунок 15.3.6 - Модель “Animals” для классификации изображений

1. Настройки блока “Запуск”

На рисунке 14.3.7 представлен блок “Запуск”. Используя ползунок Сформировать отчет можно опционально сформировать отчет после успешного выполнения модели (см. рис. 13.3.5).

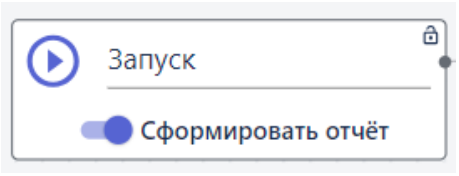


Рисунок 15.3.7 - Блок “Запуск”

2. Настройки блока “Загрузка изображений”

На рисунке 13.3.6 представлен блок источник данных - “Загрузка изображений”. Нажмите , чтобы редактировать настройки блока “Загрузка изображений”. В поле “Тип функции” выберете “Анализ данных”. В поле “Список функций” выберете “Загрузка изображений для классификации”.

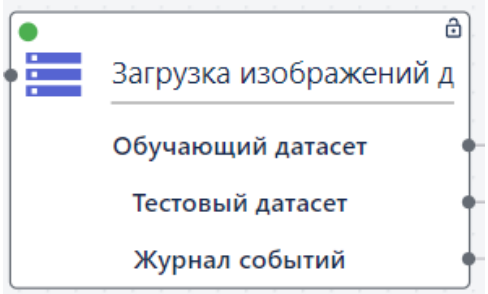


Рисунок 15.3.8 - Блок “Загрузка изображений”

В окне настройки параметров блока необходимо выбрать в Группе обучающих изображений выборку Train, а в Группе тестовых - Test:

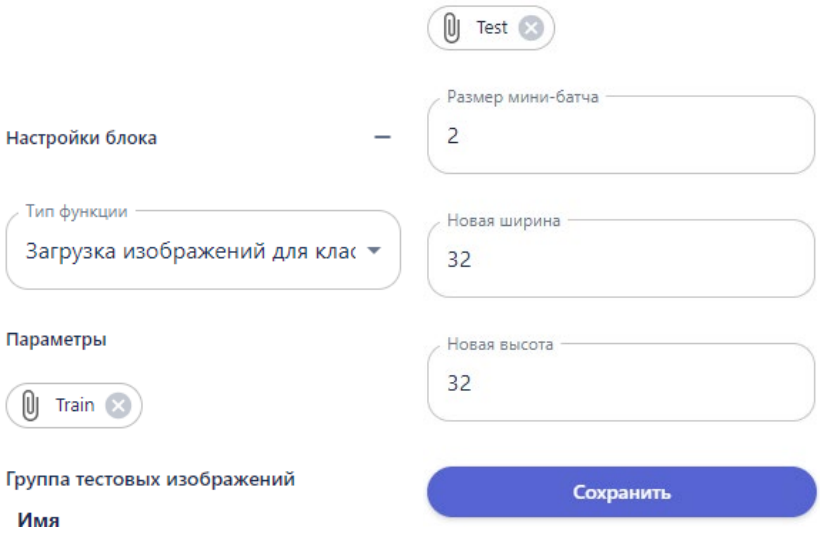


Рисунок 15.3.9 - Выбор обучающих и тестовых изображений

В поле “Размер мини-батча” укажите 2 - размер данных (количество изображений), по которым считается функция потерь при градиентном спуске. То есть при обучении на каждом шаге градиентного спуска из всего датасета берется случайным образом 2 объекта. В поле “Высота” и “Ширина” укажите 32, чтобы масштабировать изображение (уменьшить размер). Нажмите кнопку “Сохранить”, чтобы применить настройки.

3. Настройки блока “Классификация”

На рисунке 13.3.8 представлен блок “Классификация”. Нажмите  , чтобы редактировать настройки блока.

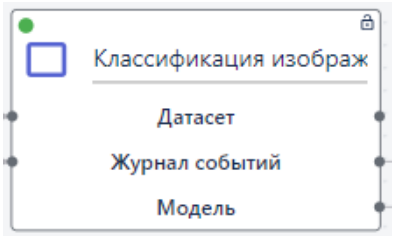


Рисунок 15.3.10 - Блок “Классификация изображений”

В поле “Тип функции” выберете “Классификация” - “Классификация изображений”.

В поле “Количество эпох” задайте 60.

В поле “Метрика для обучения” выберите “Accuracy”.

В поле “Алгоритм градиентного спуска” выберите “Adam”.

В поле “Шаг градиентного спуска” укажите null.

В поле “Функция потерь” выберите “binary_crossentropy”.

В поле “Порог классификации” оставьте значение по умолчанию (0,5).

Нажмите “Добавить слой” и заполните значения: Слой: 1.Dense; Число нейронов: 1; Функция активации: 3.sigmoid.

Нажмите “Добавить слой” еще раз и заполните значения: Слой: 2.Flatten.

Нажмите “Добавить слой” еще раз и заполните значения: Слой: 3.Conv2D; Количество фильтров: 4; Размер ядра свертки: 3,3; Размер шага свертки: 1; Толщина отбивки из нулей: Функция активации:

Сохраните настройки блока.

4. Настройки блока “Валидация”

Нажмите  , чтобы редактировать настройки блока.

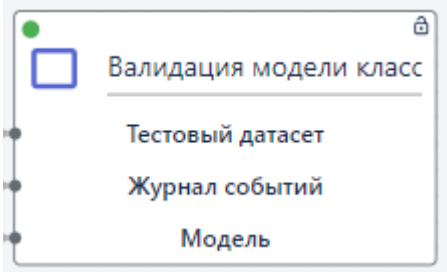


Рисунок 15.3.11 - Блок “Валидация модели классификации”

В поле “Тип функции” выберете “Глубокое обучение”. В поле “Список функций” выберете “Валидация модели классификации изображений”. В поле “Метрика” выберете “F1”.

5. Настройки блока “Сохранение”

Нажмите  , чтобы редактировать настройки блока “Сохранение”.

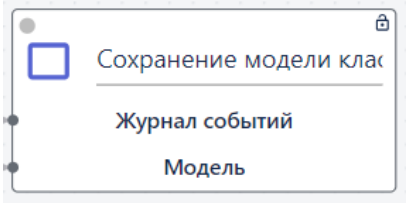


Рисунок 15.3.12 - Блок “Сохранение модели классификации”

В поле “Тип функции” выберите “Управление моделями”. В поле “Список функций” выберите “Сохранение модели классификации изображений”. В поле “Название модели” укажите название модели (например, “animals”).

6. Запуск модели классификации и визуализация результатов

После успешного запуска модели, нажмите , чтобы отобразить результаты выполнения для анализа.

6.1. Графики “История обучения”

Выберите пункт “История обучения”, чтобы оценить качество модели. Мы видим, что с каждым проходом (с каждой эпохой) точность увеличивается (см. рис. 14.3.12 Метрика Accuracy), а количество ошибок уменьшается (см. рис. 13.3.11 Функция потерь). Можно видеть, что поле “Количество эпох” равное 60 является оптимальным значением, в то время как 20 было бы недостаточно.

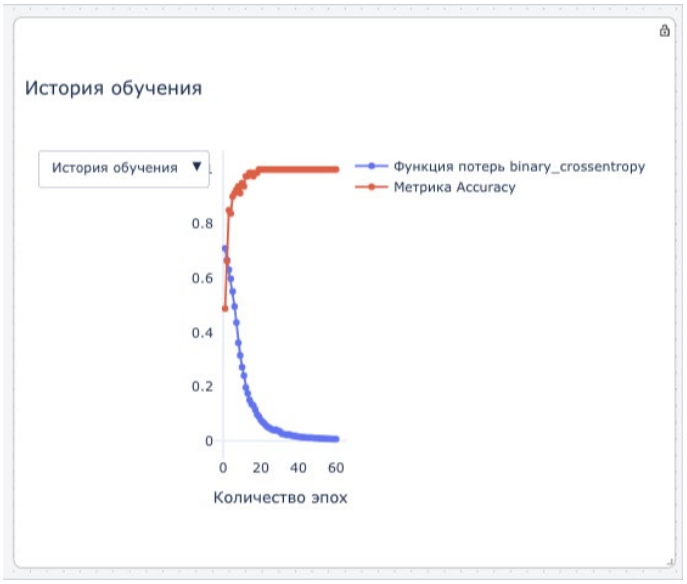


Рисунок 15.3.13 - Графики истории обучения

6.2. Матрица ошибок

Выберите пункт “Матрица ошибок”, чтобы увидеть числовые показатели. На рисунке 14.3.13 видно, что модель распознала 10 из 10 кошек, 7 из 10 собак. Для дальнейшего повышения показателей необходимо добавить новые изображения и перезапустить модель.

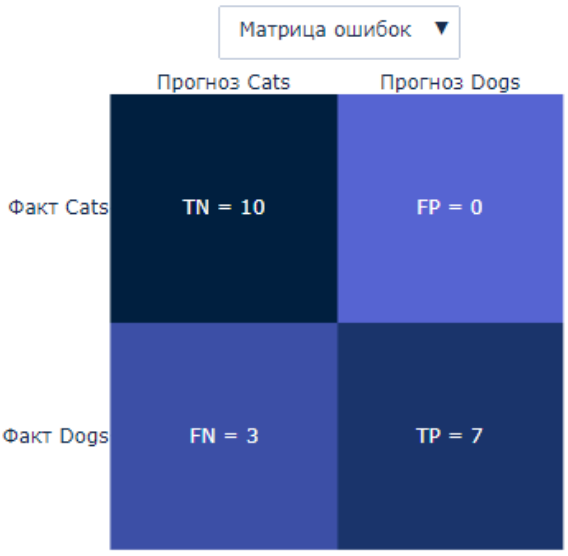


Рисунок 15.3.14 - “Тепловая карта” модели классификации

6.3. Табличные результаты валидации

Нажмите , чтобы ознакомиться табличными результатами валидации. Валидация на тестовой выборке (Метрика F1) и ошибки модели при прогнозировании классов приведены на рисунках ниже:

Валидация на тестовой выборке. Метрика F1:

	F1
1	0.889

Рисунок 15.3.15 - Таблица валидации

Ошибки модели при погнозировании классов

	Верно	Ошибка	Всего
Класс Cats	10	0	10
Класс Dogs	8	2	10

Рисунок 15.3.16 - Таблица при прогнозировании

6.4. Архитектура модели

Нажмите , чтобы посмотреть архитектуру модели:

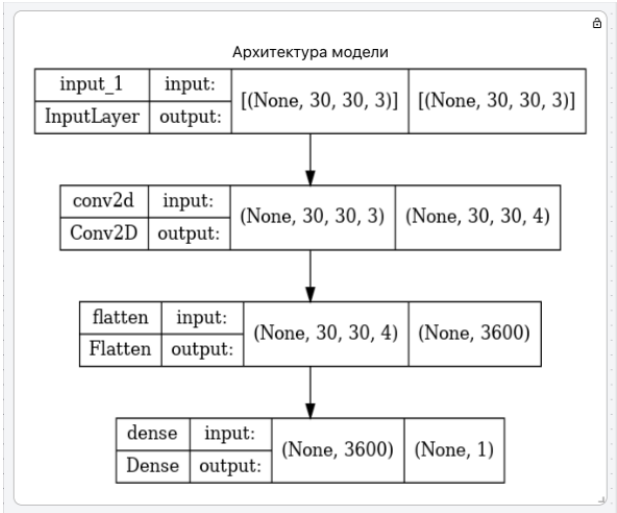


Рисунок 15.3.17 - Архитектура модели

15.3.3 Классификация родинок

1. Общая информация

В решении задачи классификации родинок требуется обнаружить на изображениях родинки и отнести их к тому или иному классу (доброкачественные или злокачественные). В настоящее время зачастую профильный специалист, классифицирующий родинки, проводит осмотр или ручную обработку изображений, вследствие чего рабочее время расходуется неэффективно и повышается риск пропуска злокачественной родинки, например медленно растущих меланом, которые могут быть ошибочно классифицированы им как доброкачественные.

Решение проблемы классификации родинок может способствовать раннему обнаружению меланомы и улучшению результатов лечения, что делает эту задачу важной и актуальной в медицинской практике.

2. Подход к решению задачи

Происходит распознавание родинок на статичных изображениях, а также определение их к тому или иному классу - злокачественные или доброкачественные. Пользователь загружает на Платформу изображения с объектом, который он хочет, чтобы был детектирован. Далее пользователь на этих изображениях выделяет объект, формируя группу разметки.

Обучение модели на стационарных изображениях

1. Загрузка данных (статичные изображения);
2. Разметка изображений;
3. Построение пайплайна;
4. Обучение модели классификации родинок;
5. Создание приложения.

Результат: Обученная на изображениях модель классификации родинок

3. Инструменты для решения задачи

Учитывая универсальность платформы BASIS AI, возможность упаковки модели в блок-схемы BPMN 2.0 и возможность использования пользователями динамических Dashboard, на платформе BASIS AI Platform необходимо:

1. Создать рабочую область “**Классификация родинок**” с созданным пайплайном для обучения модели распознавания и классификации родинок;
2. Обучить модель “**mole_classifier**” для последующего использования;
3. Подготовить дашборд “**Классификация_родинок_dashboard**”;
4. Собрать проект “**Классификация родинок**”;
5. Приложение для использования вне платформы на основе обученной модели.

4. Построение пайплайна модели бинарной классификации родинок

На рисунке ниже представлен пайплайн для создания и валидации модели.

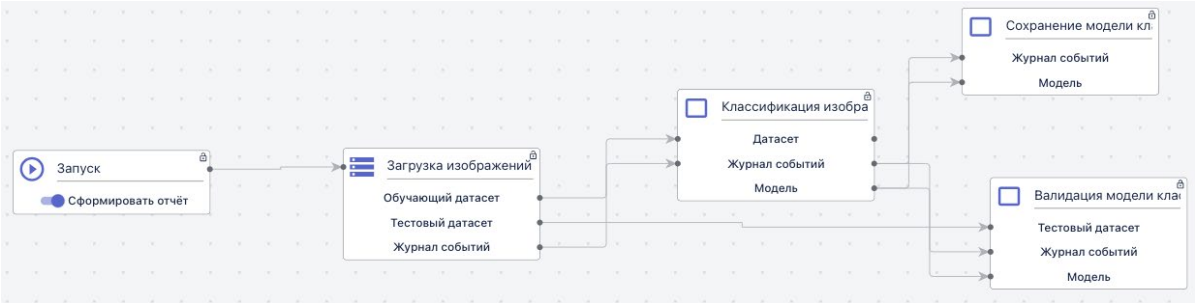


Рисунок 15.3.18 - Пайплайн для создания и валидации модели

- Блок “**Запуск**”. Блок является начальной точкой для пайплайна.

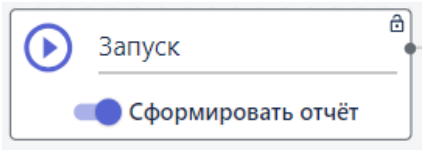


Рисунок 15.3.19 - Блок “Запуск”

Комментарий:

Если выбрана опция “Сформировать отчет” формируется отчет об обучении модели предиктивной аналитики. Краткое описание об обучении можно просмотреть нажав на в правом верхнем углу рабочей области. Оно включает в себя время обучения модели, лучшие метрики и список преобразований. Более подробный отчет, включающий в себя дополнительно визуализацию результатов обучения модели, информацию по датасету и валидацию на тестовой выборке можно посмотреть в разделе “**Отчеты**”.

- Блок “**Загрузка изображений для классификации**”. Блок реализует загрузку изображений для классификации из папок **moles_train** и **moles_test**.

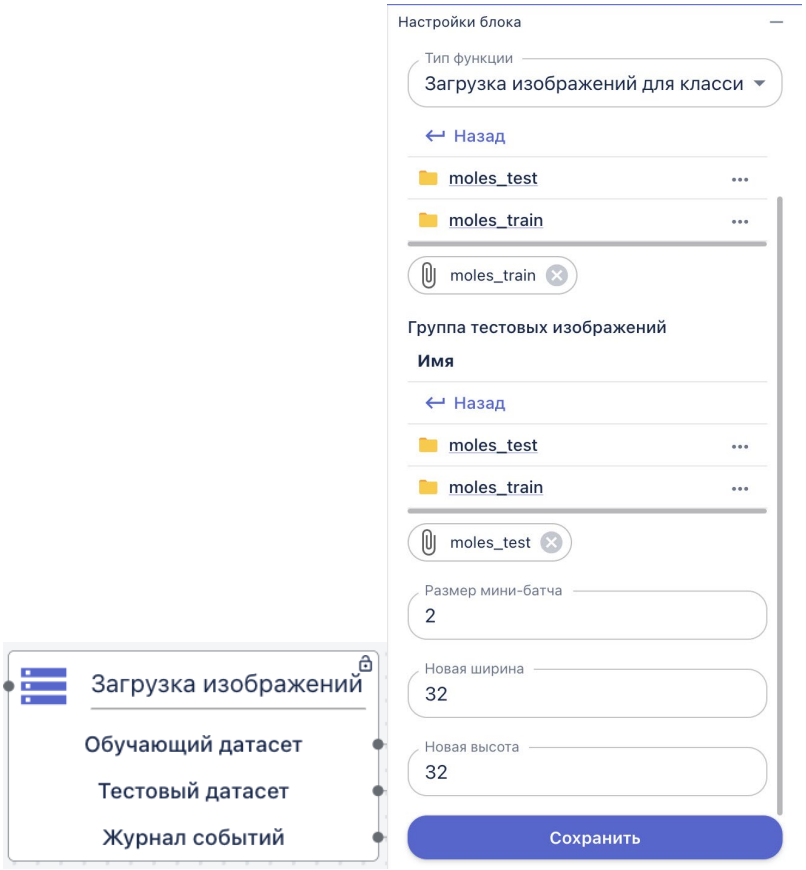


Рисунок 15.3.20 - Блок “Загрузка изображений для классификации”

Комментарий:

В папке **moles_train** размещаются изображения для обучения будущей модели машинного обучения. В папке **moles_test -** для валидации или проверки “качества” уже обученной модели. При этом для обучения модели необходимо использовать большее количество файлов, в нашем примере пропорция составляет 5 к 1.

- Блок “**Классификация изображений**”. Блок реализует восстановление зависимости между нецелевыми признаками и целевыми.

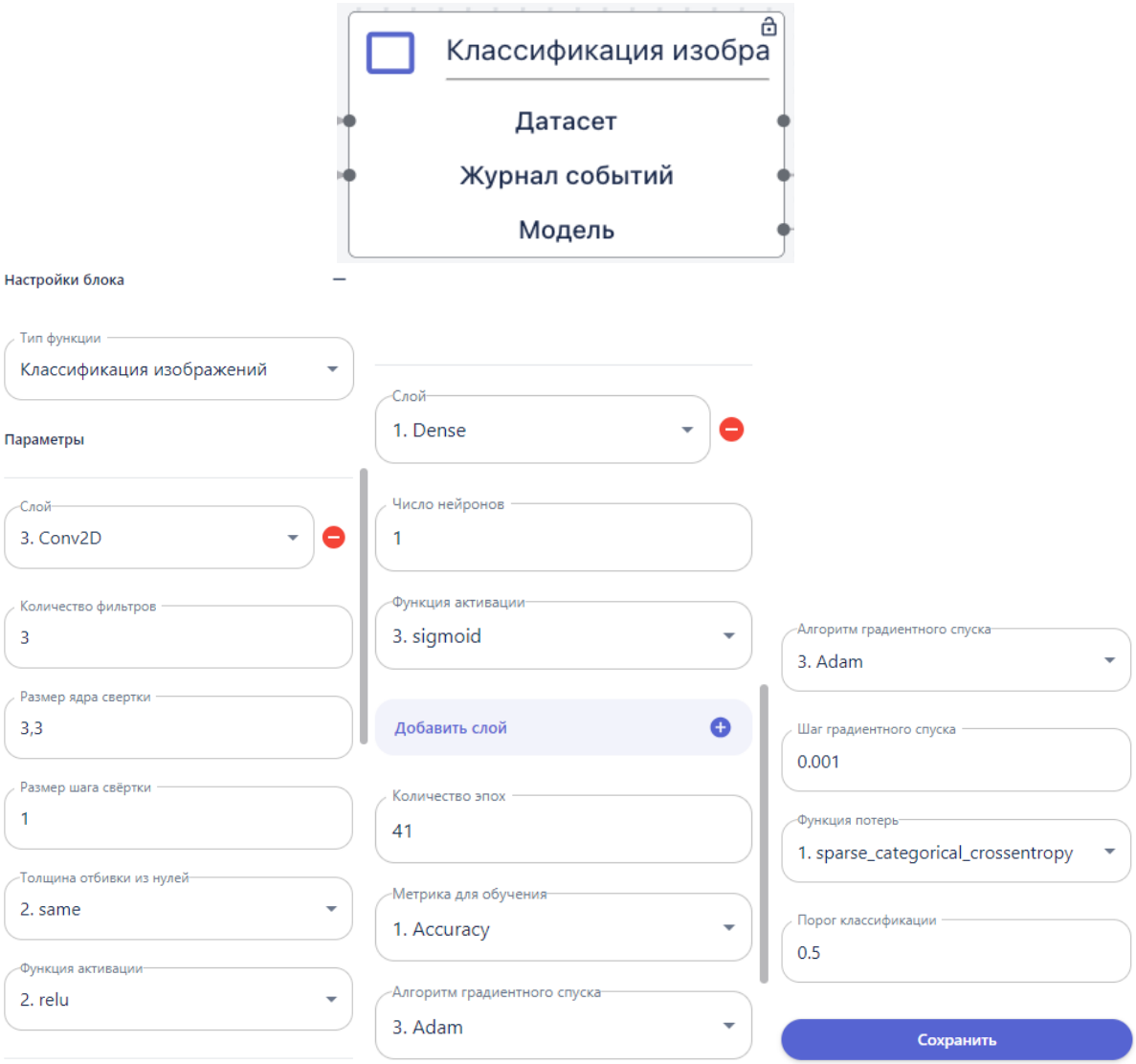


Рисунок 15.3.21 - Блок «Классификация изображений»

- Блок «**Сохранение модели классификации изображений**». В блоке происходит сохранение модели для дальнейшего использования.

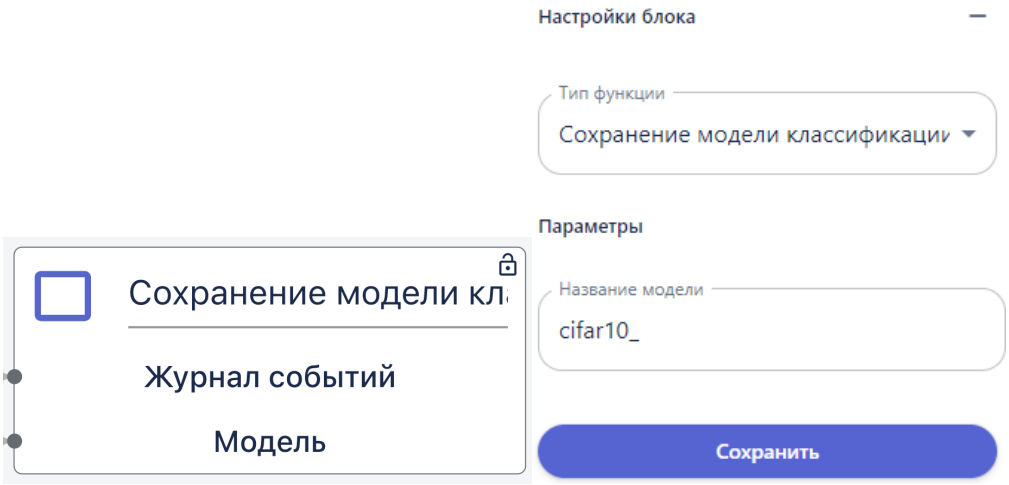


Рисунок 15.3.22 - Блок “Сохранение модели классификации изображений”

- Блок “**Валидация модели классификация**”. В блоке производится валидация обученной модели на тестовом датасете.

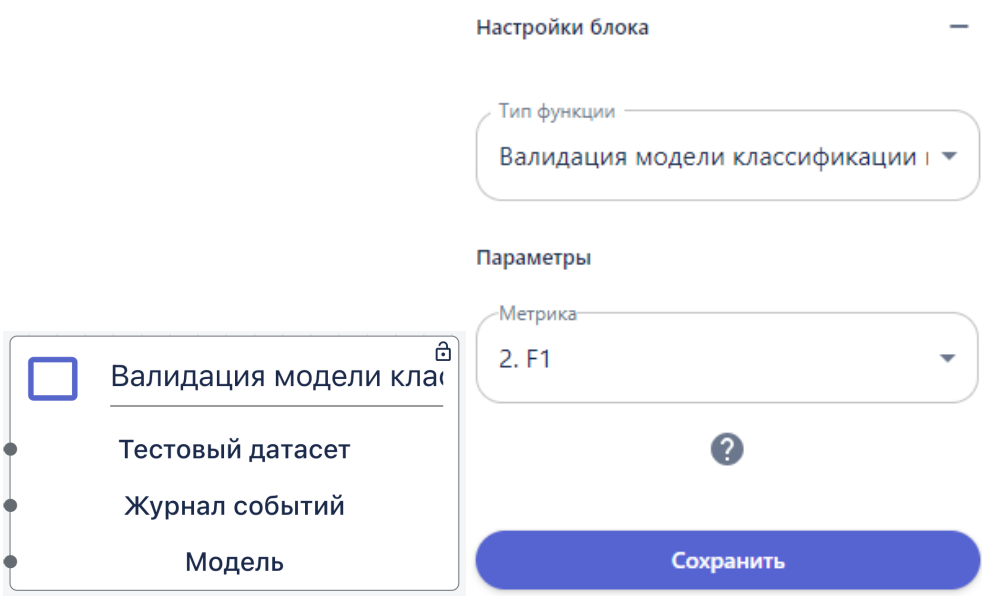


Рисунок 15.3.23- Блок “Валидация модели классификации”

2.3.4. Визуализация результатов выполнения пайплайна

Чтобы посмотреть используемые графики необходимо нажать кнопки «**Графики**», «**Таблицы**», «**Изображения**» , , и которые станут активными на панели инструментов после выполнения всех элементов пайплайна. Графики и таблицы генерируются автоматически после успешного выполнения пайплайна.

Дашборды

Для вывода инфографики на дашборд необходимо воспользоваться предварительно созданными коннекторами.

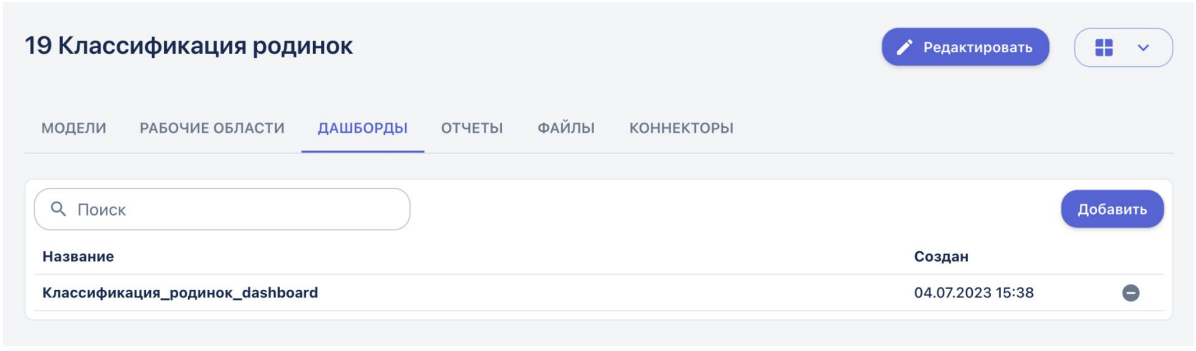


Рисунок 15.3.24 - Дашборды

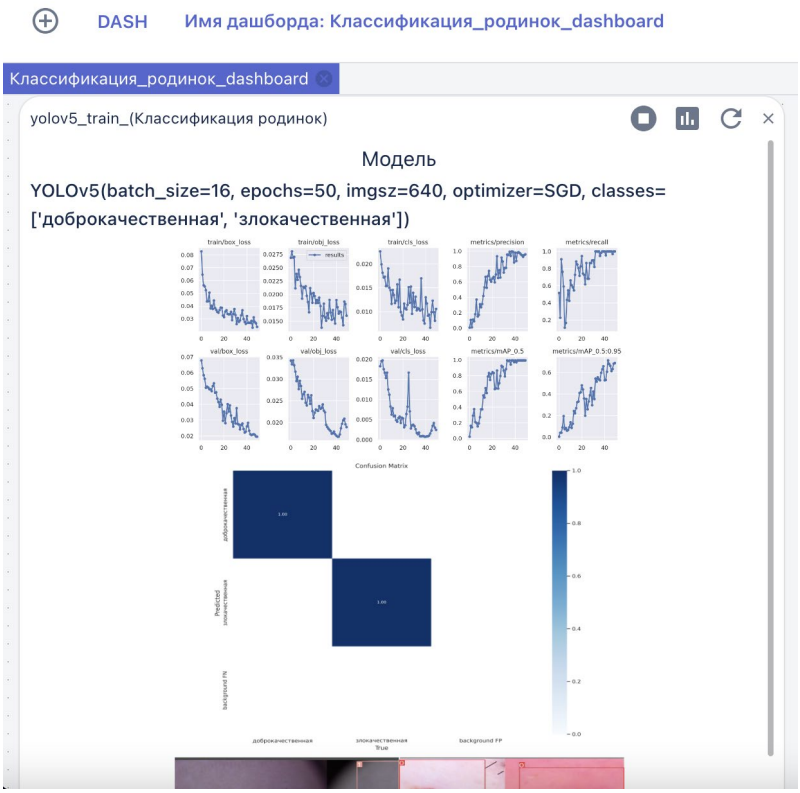


Рисунок 15.3.25 - Пример вывода данных на дашборд

Отчеты

После успешного выполнения пайплайна и нажатой на блоке “Запуск” галочке “Сформировать отчет” формируется отчет об обучении модели. Отчет сохраняется по названию рабочей области, по которой он сформирован с временной меткой создания отчета. Для просмотра отчета необходимо нажать на название и отчет откроется в отдельном окне.

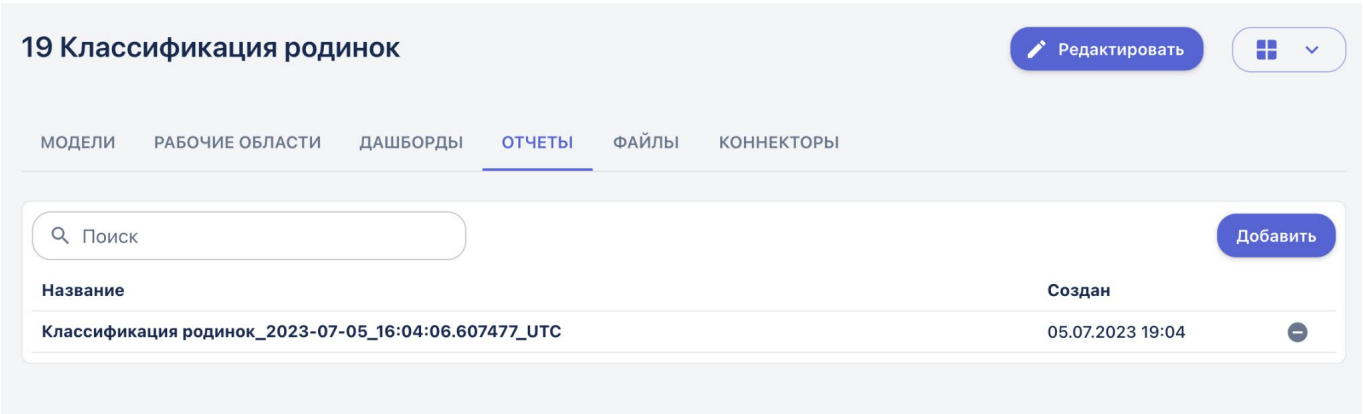


Рисунок 15.3.26 - Отчеты по проекту

15.4 Классификация текстов

Данный сценарий предполагает решение задачи бинарной классификации произведений по их авторов с использованием в качестве источника текстовых документов в формате .txt.

Для решения задачи выполните следующие действия:

1. **Загрузка данных на платформу.** Набор данных состоит из четырех книг. Половина книг написана Булгаковым, а остальная половина – Клиффордом. Эти книги представляют собой текстовые документы в формате .txt.
   1.1. Перейдите в раздел данные и нажмите кнопку «Создать»
   1.2. В открывшейся в поле «Тип» выберите значение «Категория», в поле «Название» введите название новой группы, например «Данные для классификации текстов», и нажмите кнопку «Создать».

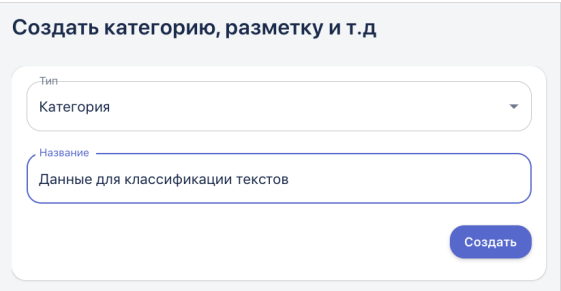


Рисунок 15.4.1 - Создание папки в разделе данные

   1.3. Перейдите в созданную папку «Данные для классификации текстов» и аналогичным образом создайте две новые папки внутри – «Text Train» и «Text Test». В группу «Text Train» будут складываться данные для обучения будущей модели машинного обучения, а в группу «Text Test» – данные для валидации или проверки ‘качества’ уже обученной модели. При этом для обучения модели необходимо использовать большее количество файлов, в нашем примере в обучающем текстовом файле содержится 5 книг, а в тестовом - 2.
   1.4. В папке «Text Train» создайте еще две папки, которые и будут определять классы, – «Clifford» и «Bulgakov». Количество классов равно двум, так как в данном сценарии решается задача бинарной классификации (для многоклассовой классификации создавалось бы больше двух классов). В класс «Clifford» загружается текстовый файл с произведениями Клиффорда, а в класс «Bulgakov» загружаются книги Булгакова.
   1.5. Для того чтобы загрузить файлы, перейдите в нужную папку и нажмите кнопку «Загрузить». Откроется окно загрузки файлов:

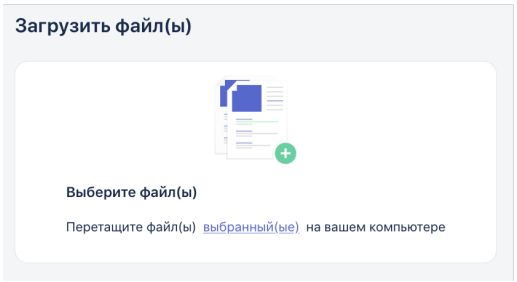


Рисунок 15.4.2 - Окно загрузки файлов

   1.6. Для выбора файлов кликните в область окна «Выберите файлы» или перетащите их по технологии drag and drop (из окна папки на вашем ПК в окно браузера). Обратите внимание: за раз можно добавить максимум 10 файлов. Соответственно, если нужно загрузить больше файлов, нужно повторить выбор несколько раз.
   1.7. После того, как все файлы выбраны, при необходимости вы можете удалить ненужные файлы, нажав на крестик в правой части строки с файлом, или нажать «Удалить все», если это требуется.

1.8. В нашем примере все произведения объединены в один текстовый файл. После его загрузки, нажмите кнопку «Загрузить»:

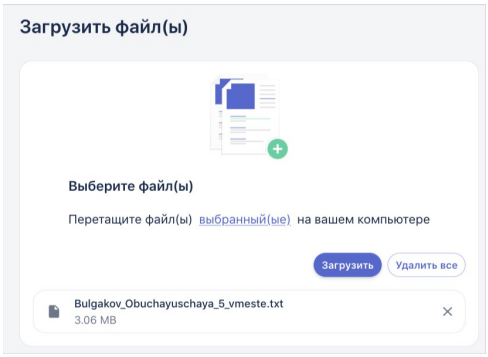


Рисунок 15.4.3 - Выбранные файлы в окне загрузки

1.9. В результате загруженные файлы отобразятся в папке:

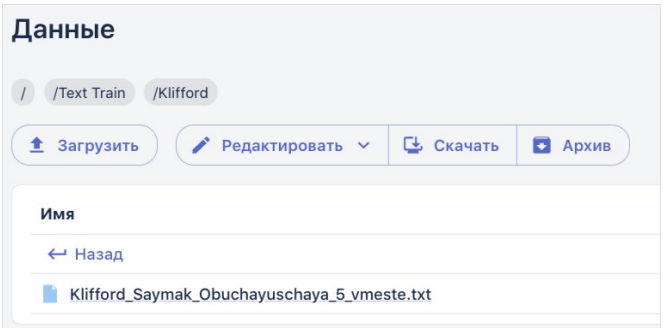


Рисунок 15.4.4 - Загруженные файлы в папке

1.10. Вышеописанные действия повторяются для папки «Text Train» -> «Bulgakov».

1.11. Далее по аналогии создаются и заполняются папки «Text Test» -> «Clifford» и «Bulgakov», туда загружаются файлы для валидации модели.

1.12. Чтобы удалить группу/класс достаточно удалить соответствующую папку в разделе Данные, нажав на три точки в строке с этой папкой.

1.13. После того, как обучающая и валидационная выборки собраны, для папок «Text Test» и «Text Train» добавляется параметр классификация. Для этого в строке с папкой нажмите на три точки и кликнуть на кнопку «Классификация», после этого содержимое папки будет готово для использования при построении модели:

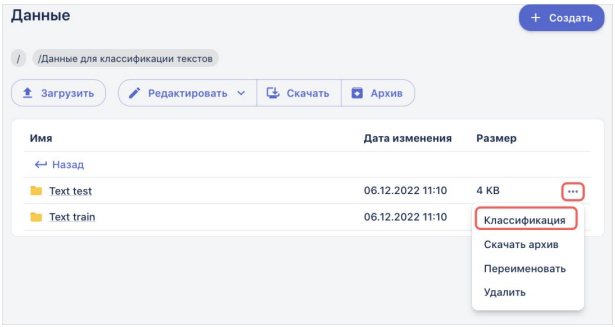


Рисунок 15.4.4 - Кнопка присвоения параметра Классификации папке

1.14. Обратите внимание: данное действие необходимо выполнить один раз. Даже если позже в папку будут добавлены новые файлы, они будут учтены при построении или запуске модели классификации.

2. **Построение блок-схемы** (блок-схему сценария см. в таблице 18.9 «Обучение модели классификации текстов»).

2.1. **Создание новой рабочей области.** Перейдите в раздел «Моделирование» -> Рабочая область. Нажав на ⊕ в верхней части экрана, создайте новую рабочую область с названием «Тексты».

2.2. **Блок запуск.** Добавьте на рабочую область элемент «Запуск»:

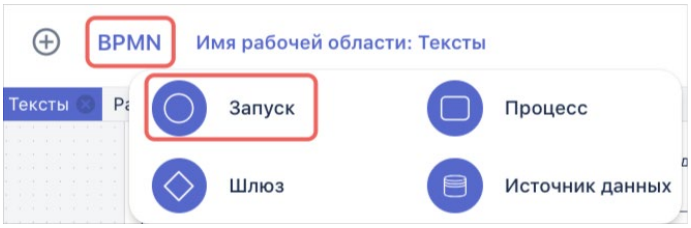
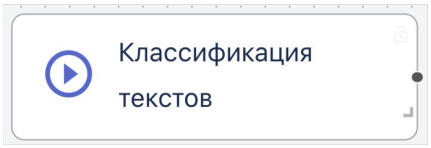


Рисунок 15.4.5 - Добавление элемента Запуск на рабочую область

Дважды кликните на название элемента, чтобы переименовать его в «Классификация текстов»:



2.3. **Блок Источник данных.** Добавьте на рабочую область элемент «Источник данных» и в качестве функции выберите «Загрузка текстовых файлов для классификации».

2.3.1. В разделе «Группа обучающих текстов» выберите папку «Text Train», нажав на три точки в строке с ее наименованием и кликнув «Выбрать», в результате папка отобразится в нижней части списка:

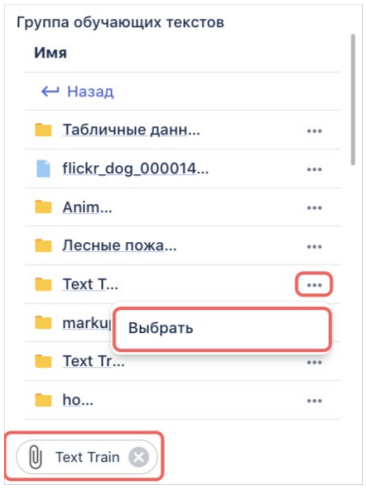
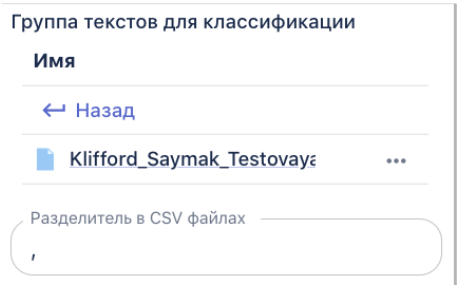


Рисунок 15.4.6 - Выбор папок для классификации

2.3.2. Аналогичным образом выберите папку «Text Test» в разделе ниже «Группа тестовых текстов»

2.3.3. В поле «Группа текстов для классификации» вы можете выбрать файл, который необходимо классифицировать с применением обученной модели.



2.3.4. Сохраните настройки блока

2.3.5. Переименуйте блок в «Загрузка текстов»

2.3.6. Соедините элементы блок схемы:

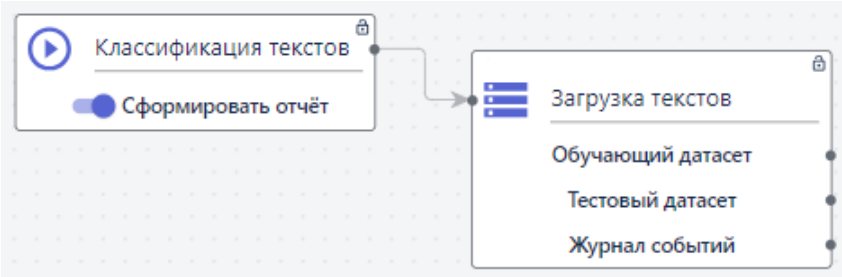


Рисунок 15.4.7 - Соединение элементов блок схемы

2.4. **Блок Процесс, Фильтрация шума.** Добавьте на рабочую область два элемента процесс. Для каждого выберите функцию «Предобработка данных» -> «Фильтрация текстового шума». Не забывайте нажимать «Сохранить» каждый раз после изменения параметров блока процесс.

2.4.1. Переименуйте один блок процесс в «Фильтрация шума train», второй в - «Фильтрация шума test».

2.4.2. Соедините элементы:

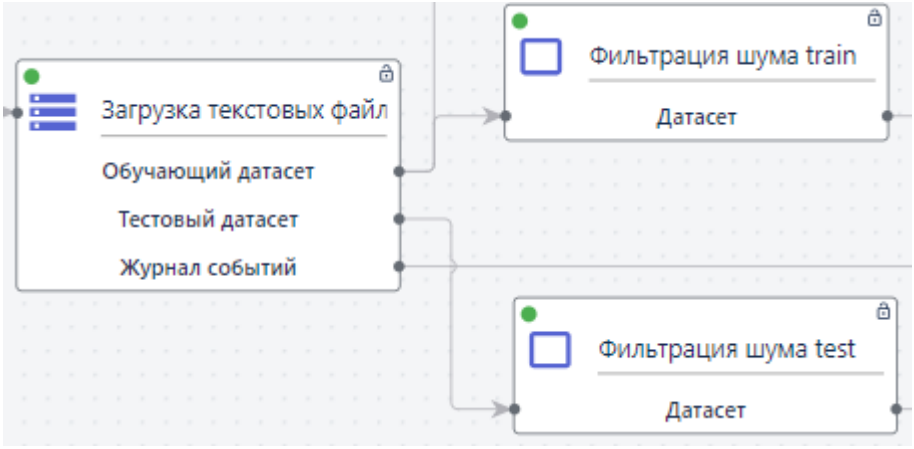


Рисунок 15.4.8 - Соединение элементов блок схемы

2.5. **Блок Процесс, Лемматизация.** Добавьте на рабочую область три элемента процесс. Для каждого выберите функцию «Предобработка данных» -> «Лемматизация текста». Не забывайте нажимать «Сохранить» каждый раз после изменения параметров блока процесс.

2.5.1. Переименуйте один блок процесс в «Лемматизация train», второй в - «Лемматизация test»:

2.5.2. Соедините элементы:

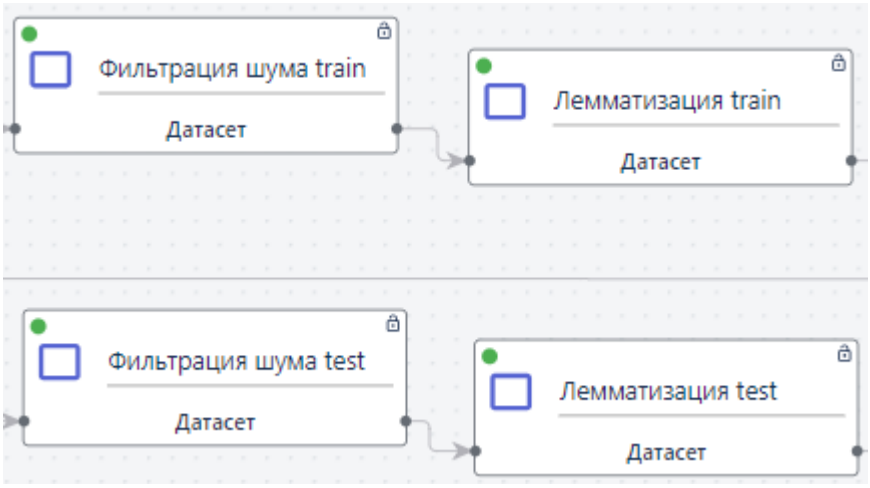


Рисунок 15.4.9 - Соединение элементов блок схемы

2.6. **Блок Процесс, Автореферирование.** Добавьте на рабочую область элемент процесс. Выберите функцию «Работа с текстом» -> «Автореферирования текста».

2.6.1. В параметрах блока в поле «Объем автореферата» укажите 200 (это максимальное количество символов, которое отобразится в качестве краткого содержания после применения функции).

2.6.2. Сохраните параметры блока.

2.6.3. Соедините элементы:

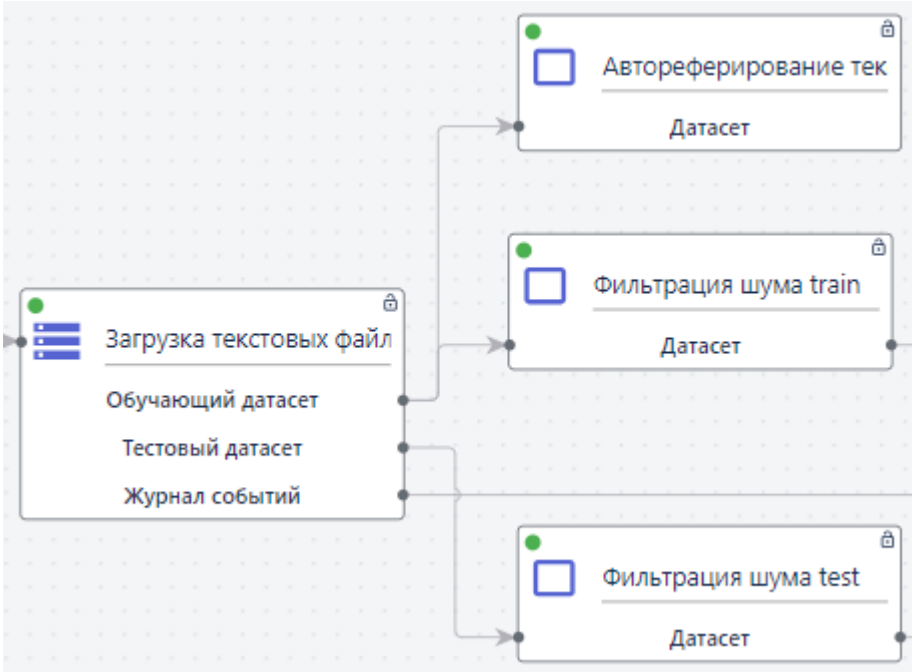


Рисунок 15.4.10 - Соединение элементов блок схемы

2.7. **Блок Процесс, Векторизация.** Добавьте на рабочую область элемент процесс. Выберите функцию «Предобработка данных» -> «Векторизация текста».

2.7.1. В параметрах блока выберите «Метод векторизации»: Word to Vec; «Максимальная размерность текста» - 25000; «Количество признаков» - 25:

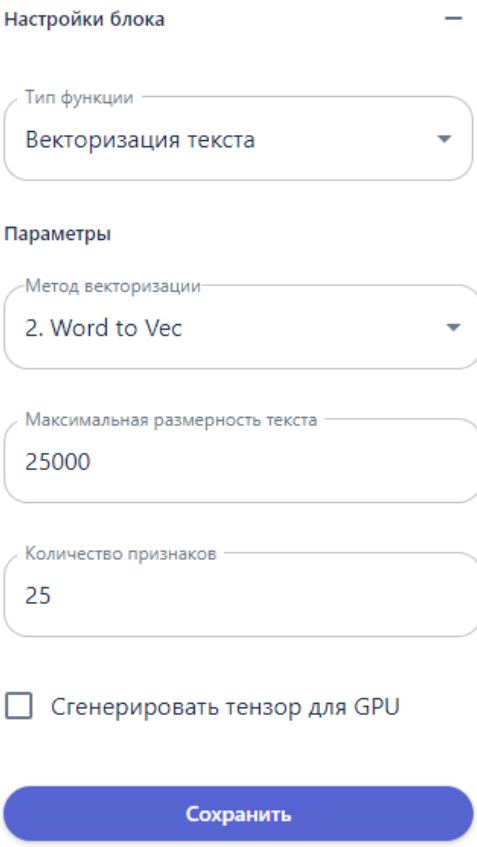


Рисунок 15.4.11 - Параметры блока «Векторизация текста»

2.7.2. Сохраните параметры блока

2.7.3. Переименуйте блок в «Векторизация»

2.7.4. Соедините элементы:

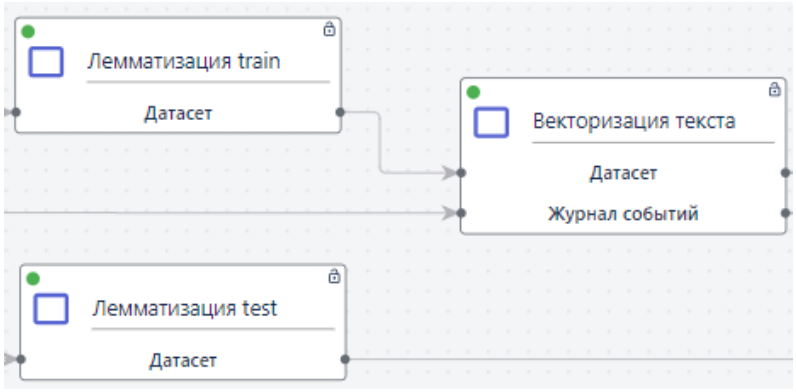


Рисунок 15.4.12 - Соединение элементов блок схемы

2.8. **Блок Процесс, Классификация.** Добавьте на рабочую область элемент процесс. Выберите функцию «Классификация» -> «Логистическая регрессия».

2.8.1. В параметрах блока выберите «Коэффициент регуляции»: 1; «Порог классификации» - 0,5; не нужно устанавливать галочки в полях «Флаг возврата вероятности при прогнозе» и «Оптимизация гиперпараметров»:

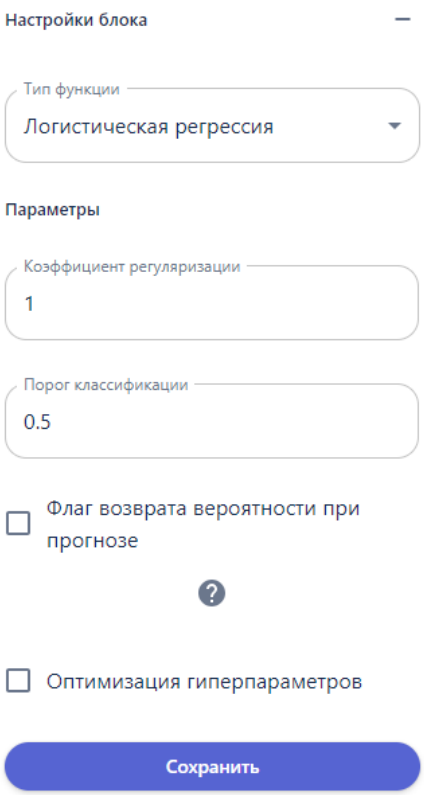


Рисунок 15.4.13 - Параметры блока «Логистическая регрессия»

2.8.2. Сохраните параметры блока

2.8.3. Соедините элементы:

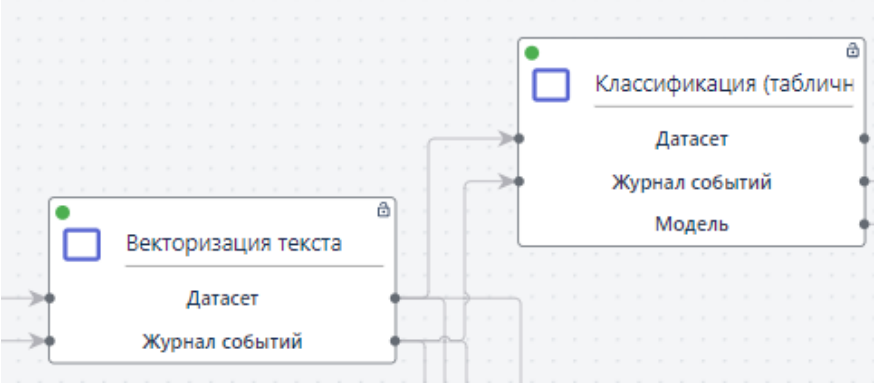
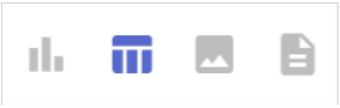


Рисунок 15.4.14 - Соединение элементов блок схемы

3. **Запуск блок-схемы**. Чтобы запустить *собранную блок-схему* нажмите кнопку «▶» на первом элементе блок-схемы – «Запуск». После этого внешний вид блока изменится и появится возможность создания отчета, активируйте настройку при необходимости. Повторно нажмите кнопку «Запуск», блок схема начнет обрабатывать блоки. После того, как все блоки отработают с зелеными индикаторами, процесс запуска считается успешно пройденным.
4. **Визуализация результатов.** После успешной отработки блок схемы на верхней панели активируются кнопки визуализации:



Нажав на иконку с таблицей, вы сможете выбрать из следующих доступных визуализаций:

- Отфильтрованные тестовые данные для каждого датасета:

Отфильтрованые текстовые данные

text	label
всё живое выехал нашего городишка повернул шоссе позади оказался грузовик	0
белая гвардия посвящается любови евгеньевне белозерской пошел мелкий снег повалил	1

- Лемматризированные текстовые данные для каждого датасета

Лемматизированные данные

text	label
все живой выезжать наш городишко повертывать шоссе позади оказываться грузовик	0
белый гвардия посвящаться любовь евгеньевна белозерский пойти мелкий снег повалить	1

- Краткое содержание после применения блока «Автореферирование»:

Краткое содержание

text	label
В Лос-Анджелесе я протаранил грузовик, за рулем которого был Джералд За рулем был Шкалик Грант, который меня заклинал не допустить, чтобы	0
На русском языке вышла книга «Капитанская дочка» Александра Турби мертвые», написанная в год Рождества Христова 1918 года.	1

15.5 Кластеризация Spark

В данном примере рассматривается пример работы с платформой с применением функций Spark. Основная цель - провести кластеризацию (обучение без учителя), а также сформировать разметку, то есть разбить объекты на 2 класса, "0" и "1". В данном примере обрабатываются данные о сетевом трафике, объектами являются сессии. Глобальная цель - обнаружить аномальные сессии, то есть решить задачу бинарной классификации. Цель пайплайна - разделить сессии на кластеры, а затем решить, какие кластеры являются аномальными. Сессии, попавшие в аномальный кластер, получат метку "1", остальные - "0". На выходе имеем размеченные данные, которые далее могут быть использованы для обучения.

1. Загрузка входных данных:
   1.1. В левой части главного окна на панели вкладок Системы откройте вкладку «Данные».
   1.2. Нажмите на кнопку «Загрузить» на верхней панели.
   1.3. В открывшемся окне нажмите на кнопку «Выбрать файлы» и укажите путь к заранее подготовленному файлу **1000_first_sessions.csv**, в котором содержатся данные о сетевом трафике. Второй вариант – перенести файлы в этот раздел по технологии «drag n drop».

   Выбранные файлы отобразятся в нижней части окна загрузки:

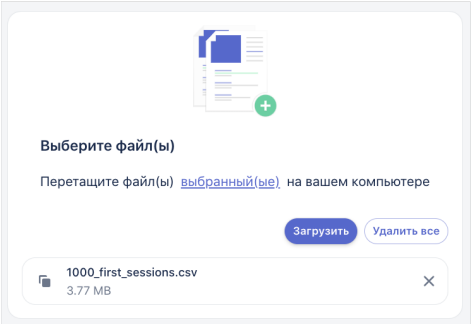


Рисунок 15.5.1 – Отображение выбранного файла

1.4. Нажмите на кнопку «Загрузить». Файл с входными данными отобразится в папке.

2. Создание новой рабочей области

Полностью блок схема представлена в Таблице.

2.1. Перейдите в пункт меню системы **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка ⊕):

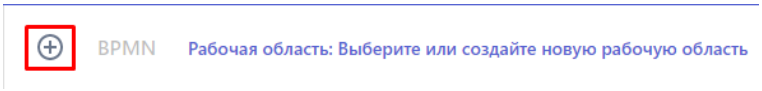


Рисунок 15.5.2 - Создание новой рабочей области

2.2. В открывшейся форме введите название новой рабочей области «Spark_Traffic» и нажмите кнопку «Создать»:

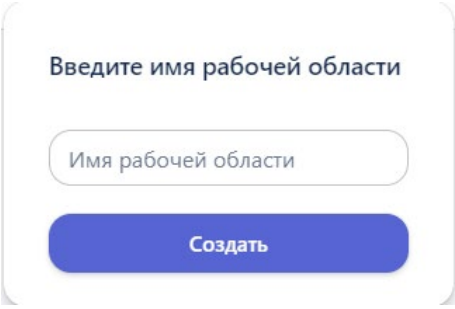


Рисунок 15.5.3 - Ввод имени рабочей области

2.3. На панели инструментов отобразится название созданной рабочей области.

3. Добавление элемента «Запуск»:

3.1. На панели инструментов блок-схемы нажмите кнопку «Добавить элемент» (кнопка BPMN)

3.2. В открывшейся библиотеке графических элементов выберите элемент «Запуск» :

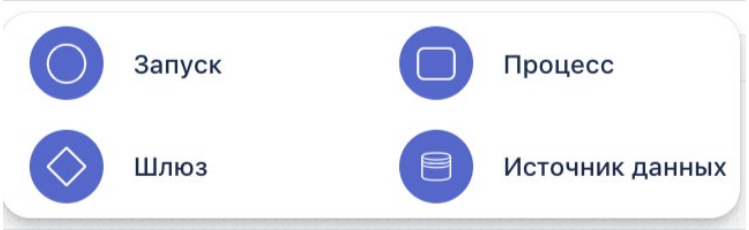


Рисунок 15.5.4 - Возможные элементы блок схемы

3.3. На рабочую область добавится элемент «Запуск»:

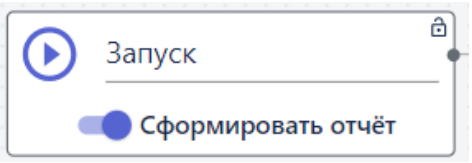


Рисунок 15.5.5 - Блок Запуск

3.4. Переименуйте элемент дважды кликнув на слово «Запуск», задайте новое название - «Кластеризация» и кликните в пустое место на рабочей области для сохранения.

4. Добавление и настройка элемента «Источник данных».

4.1. Добавьте на рабочую область элемент «Источник данных»:

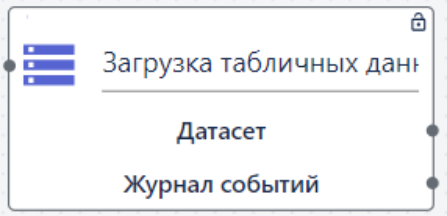


Рисунок 15.5.6 - Блок Источник данных

4.2. **Открытие настроек элемента**. На элементе «Источник данных» нажмите на кнопку. Справа откроется панель настроек элемента, где будут отображаться созданные в разделе папки и файлы с табличными данными.

4.3. **Выбор типа загрузки.** Из списка выпадающих функций выберите «Загрузка табличных данных из файла csv Spark»

4.4. **Выбор данных для загрузки в блок-схему**. Для того чтобы найти нужный файл, кликните на папку и перейдите в нее, выберите из списка файл, загруженный в Систему в шаге 1 «1000_first_sessopns.csv», нажмите на три точки в строке с ним и кликните «Выбрать». Внизу отобразится название выбранного файла:

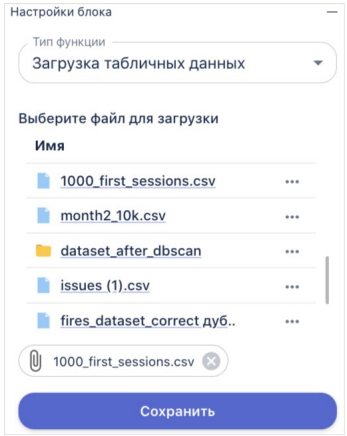


Рисунок 15.5.7 - Выбор файла

4.5. **Сохранение настроек элемента**. На панели настроек элемента нажмите на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

4.6. **Ввод названия элемента**. Чтобы задать название элемента нужно дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области, и ввести нужное название в поле с названием, доступным для редактирования:

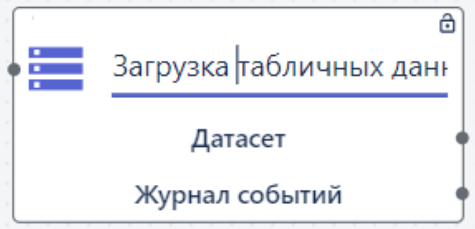


Рисунок 15.5.8 - Блок источник данных

4.7. Введите название «Загрузка Spark CSV (file)» и кликните на пустое место на рабочей области для сохранения.

4.8. **Установка соединений**. Соедините выходную точку элемента «Запуск» с входной точкой элемента «Источник данных» с помощью левой кнопка мыши:

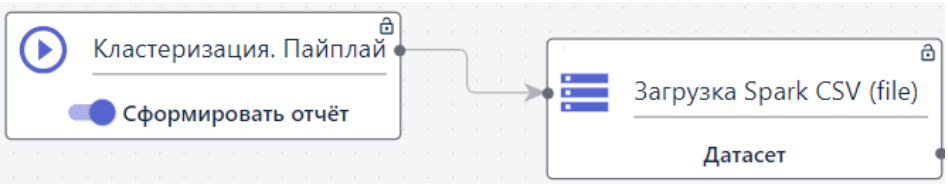


Рисунок 15.5.9 - Соединение элементов Запуск и Источник данных

5. Выбор признаков и целевых признаков. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

5.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> функция «Выбор признаков и целевых признаков».

5.2. В поле «Признаки» укажите: поочередно следующие признаки, нажимая Enter после ввода каждого: source_ip, destination_ip, source_port, destination_port, bytes, packages_count. Или вы можете найти в списке исходный файл 1000_first_sessions.csv, нажать на три точки в строке с его названием и кликнуть «Выгрузить признаки», тогда система автоматически заполнит поле «Признаки» всем вариантами из датасета и вам останется только убрать лишние.

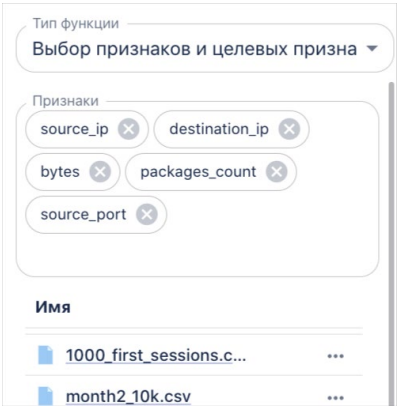


Рисунок 15.5.10 - Параметры блока «Выбор признаков и целевых признаков»

5.3. На панели настроек элемента нажмите на кнопку «Сохранить».

5.4. Соедините с элементы:

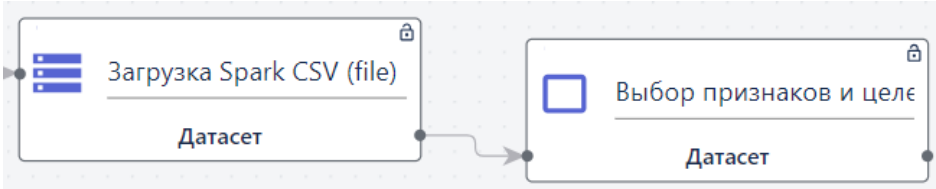


Рисунок 15.5.11 - Соединение элементов «Источник данных» и «Выбор признаков»

6. Порядковое кодирование признаков. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

6.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> группа «Препроцессинг» -> функция «Порядковое кодирование признаков».

6.2. В поле «Выбранные признаки» вместе с квадратными скобками введите следующие признаки, нажимая Enter после ввода каждого: source_ip, destination_ip, source_port, destination_port. Или вы можете найти в списке исходный файл 1000_first_sessions.csv, нажать на три точки в строке с его названием и кликнуть «Выгрузить признаки», тогда система автоматически заполнит поле «Признаки» всем вариантами из датасета и вам останется только убрать лишние:

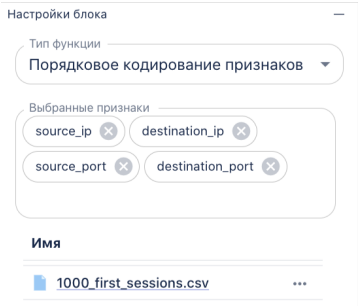


Рисунок 15.5.12 - Параметры блока «Порядковое кодирование»

6.3. На панели настроек элемента нажмите на кнопку «Сохранить».

6.4. Соедините с элементы:

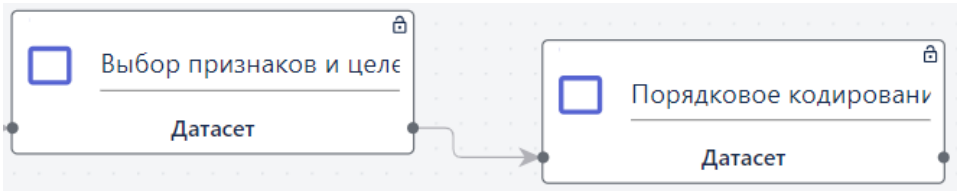


Рисунок 15.5.13 - Соединение элементов «Выбор признаков» и «Порядковое кодирование»

7. Нормализация признаков. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

7.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> группа «Препроцессинг» -> функция «Нормализация признаков».

7.2. На панели настроек элемента нажмите на кнопку «Сохранить».

7.3. Измените название элемента на «Нормализация».

7.4. Соедините с элементы:

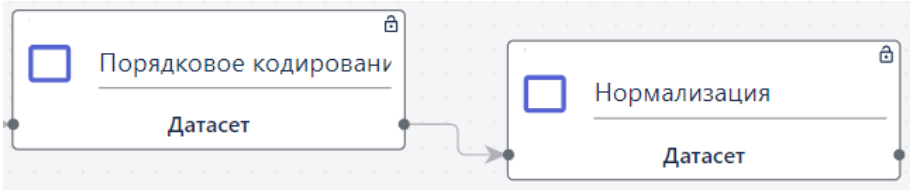


Рисунок 15.5.14 - Соединение элементов «Порядковое кодирование» и «Нормализация»

8. Кластеризация Spark DBSCAN. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

8.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> группа «Кластеризация» -> функция «Кластеризация Spark DBSCAN».

8.2. В поле «Порог для отнесения кластера к аномалиями» укажите: 100.

8.3. В поле «Радиус» укажите: 0,1.

8.4. В поле «Число соседей» укажите: 4.

8.5. В поле «Метрика расстояния» из выпадающего списка выберите: «Евклидово»

8.6. Установите галочку для параметра «Флаг векторизации признаков».

8.7. В поле столбец для группировки перед векторами впишите: session_id

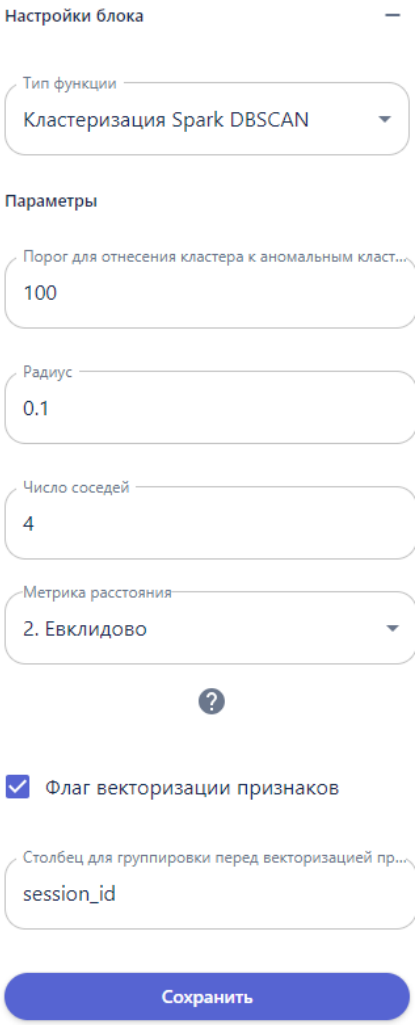


Рисунок 15.5.15 - Параметры блока «Кластеризация Spark DBSCAN»

8.8. На панели настроек элемента нажмите на кнопку «Сохранить».

8.9. Соедините с элементы:

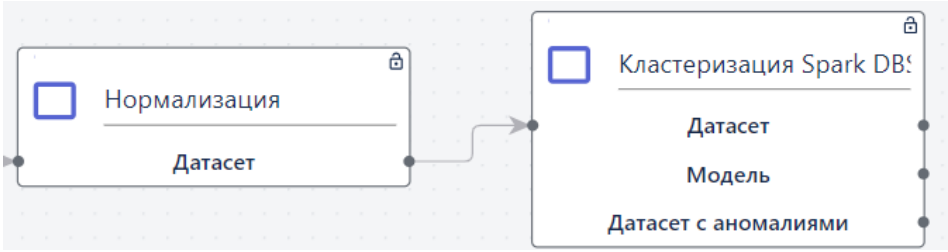


Рисунок 15.5.16 - Соединение элементов Нормализация и Кластеризация Spark DBSCAN

9. Сохранение датасета Spark. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

9.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> функция «Сохранение датасета Spark в CSV»

9.2. В поле «Путь до директории для датасета» укажите название папки в разделе «Данные», куда будет сохраняться датасет

9.3. В поле «Название датасета» пропишите вручную необходимое наименование, например, dataset_after_dbscan

9.4. Не ставьте галочку для параметра «Добавить данные к датасету»

9.5. Сохраните настройки

9.6. Измените название блока на «Датасет в CSV»

10. Сохранение модели Spark. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

10.1. На панели свойств элемента выбрать из списка функцию: тип функции «Управление моделями» -> функция «Сохранение модели Spark»

10.2. В поле «Название» пропишите вручную необходимое наименование, например, DBSCAN_spark_model:

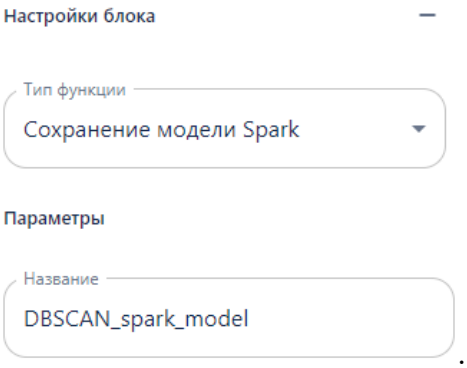


Рисунок 15.5.17 - Параметры блока «Сохранение модели Spark»

10.3. Сохраните настройки

10.4. Измените название блока на «Сохранение модели DBSCAN»

11. Сохранение датасета Spark. Чтобы в загруженном датасете выделить признаки и целевые признаки нужно добавить на рабочую область элемент «Процесс» и настроить его:

11.1. На панели свойств элемента выбрать из списка функцию: тип функции «Spark» -> функция «Сохранение датасета Spark в CSV»

11.2. В поле «Путь до директории для датасета» укажите название папки в разделе «Данные», куда будет сохраняться датасет

11.3. В поле «Название датасета» пропишите вручную необходимое наименование, например, dataset_with_anomalies_after_dbscan

11.4. Не ставьте галочку для параметра «Добавить данные к датасету»

11.5. Сохраните настройки

11.6. Измените название блока на «Датасет в CSV (аномалии)»

12. Соединение элементов. Соедините элементы, созданные в пункте 9, 10 и 11 следующим образом:

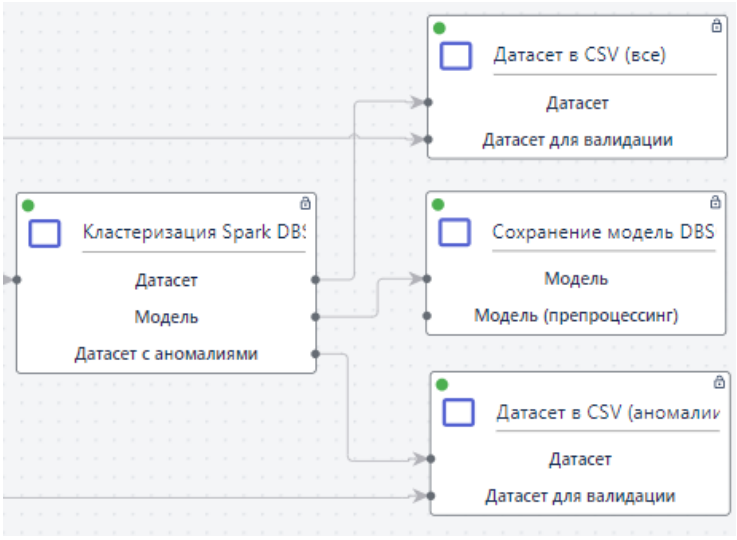


Рисунок 15.5.18 - Соединение элементов «Кластеризация Spark DBSCAN» и блоков сохранения

13. Запуск пайплайна. Чтобы запустить блок схему нажмите на кнопку на первом элементе «Запуск» собранной блок-схемы. При этом отображение элемента «Запуск» изменится и появится опция **Сформировать отчет**:

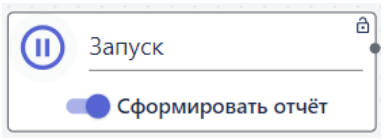


Рисунок 15.5.19 - Блок Запуск

Если активировать параметр «Сформировать отчет», в результате запуска пайплайна будет создан отчет.

14. Визуализация результатов. После того как все элементы схемы будут успешно обработаны, на панели инструментов появляются кнопки:

Вы должны увидеть следующие визуализации:

1) График «Spark DBSCAN». Этот график позволяет построить диаграмму рассеяния для трех и более пар признаков. Чем больше размер точек или пузырей на диаграмме

- тем больше взаимосвязь между признаками. График можно удалять, приближать и т д:

Рисунок 15.5.20 - График Spark DBSCAN

2) График «Объем кластеров». Круговая диаграмма показывает объем кластеров - т.е. сколько % объектов входит в каждый отдельный кластер. Имена кластеров сортируются в зависимости от их веса в общем проценте. Так в нашем примере большинство объектов попали в кластер -1 и составили 96,1%:

Рисунок 15.5.21 - График Объем кластеров

3) Таблица «Датасет 1000_first_sessions.csv». Это исходный датасет, загруженный в качестве источника данных:

Датасет 1000_first_sessions.csv

session_id	date_time	source_ip	source_port	destination_ip	destination_port	bytes	packages_count
64210550153181.0	2022-04-20 15:03:27.404	10.174.174.236	52263	10.253.252.11	9400	523	5
64210550153181.0	2022-04-20 15:03:27.405	10.253.252.11	9400	10.174.174.236	52263	1523	3
64210550153181.0	2022-04-20 15:05:07.482	10.174.174.236	52263	10.253.252.11	9400	40	2
64210550153181.0	2022-04-20 15:05:07.482	10.253.252.11	9400	10.174.174.236	52263	20	1
64210550153181.0	2022-04-20 15:05:27.734	10.174.174.236	52302	10.253.252.11	9400	503	4
64210550153181.0	2022-04-20 15:05:27.734	10.253.252.11	9400	10.174.174.236	52302	1523	3
64210550153181.0	2022-04-20 15:07:07.810	10.174.174.236	52302	10.253.252.11	9400	40	2
64210550153181.0	2022-04-20 15:07:07.810	10.253.252.11	9400	10.174.174.236	52302	20	1
64210550153181.0	2022-04-20 15:07:28.110	10.174.174.236	52347	10.253.252.11	9400	503	4
64210550153181.0	2022-04-20 15:07:28.110	10.253.252.11	9400	10.174.174.236	52347	1523	3

Рисунок 15.5.22 - Таблица Датасет 1000_first_sessions.csv

4) Таблица «Количество объектов в каждом кластере». Данная таблица показывает, сколько объектов попало в разные кластеры:

Количество объектов в каждом кластере

label	volume
4	3
3	4
2	6
1	8
0	18
-1	961

Рисунок 15.5.23 - Таблица «Количество объектов в каждом кластере»

5) Таблица «Датасет dataset_after_dbscan» отображает преобразованный датасет, где содержатся данные по кластеризации:

Датасет dataset_after_dbscan

label	session_id	concat[0]	concat[1]	concat[2]	concat[2822]	concat[2823]	concat[2824]	concat[2825]	is_anomaly
-1	64210550153181	0.21713441610336304	0.001485884073190391	0.15901444852352142	0	0	0	0	0
-1	76771566702416	0.875923216342926	0.001485884073190391	0.783297061920166	0	0	0	0	0
-1	129323973953981	0.10930576175451279	0.01634472423017693	0.8732436299324036	0	0	0	0	0
-1	136859227321959	0.6174298524856567	0.004457652103155851	0.3076390326023102	0	0	0	0	0
-1	148877881840344	0.0014771048445254564	0.22288261353969574	0.0000989511172519996	0	0	0	0	0
-1	210567688532991	0.005908419378101826	0.005943536292761564	0.0006926578535315913	0.9161884188652039	0.0006721075624227524	0.0001346979024674510	0.0002227171498816460	0
-1	217590991923358	0	0.01931649260222912	0	0	0	0	0	0
-1	218100980642663	0.007385524455457926	0.667161941528320	0.0001979022345039993	0	0	0	0	0
-1	255330641158839	0.005908419378101826	0.3521545231342315	0.0006926578535315913	0	0	0	0	0
-1	260730379318740	0.3308714926242828	0.9034175276756287	0.0003958044690079987	0	0	0	0	0
-1	261652820257666	0.31905466318130493	0.2823179662227630	0.017910152673721313	0	0	0	0	0
-1	274134409307370	0	0.037147101014852524	0.0002968533663079142	0	0	0	0	0
-1	347983599388379	0.32348597049713135	0.338781565427780	0.225509598851203	0	0	0	0	0
-1	385729102920192	0.1831610053776947	0.750371456146240	0.0001979022345039993	0	0	0	0	0
-1	432514239859384	0.478581964969635	0.213967308402061	0.087373837828636	0	0	0	0	0
-1	461858665095623	0.7223042845726013	0.294205039739608	0.449435979127883	0	0	0	0	0
-1	464436599479288	0.2466765195313203	0.002971768146380782	0.168117955327034	0	0	0	0	0

Рисунок 15.5.24 - Таблица «Датасет dataset_after_dbscan»

6) График «Сформированные кластеры» показывает результат разбора текстовых данных на некоторое количество групп (кластеров), связанных между собой наборов ключевых слов. Строки, в которых встречаются похожие по смыслу слова, объединяются в один кластер. На графике отображаются сформированные кластеры.

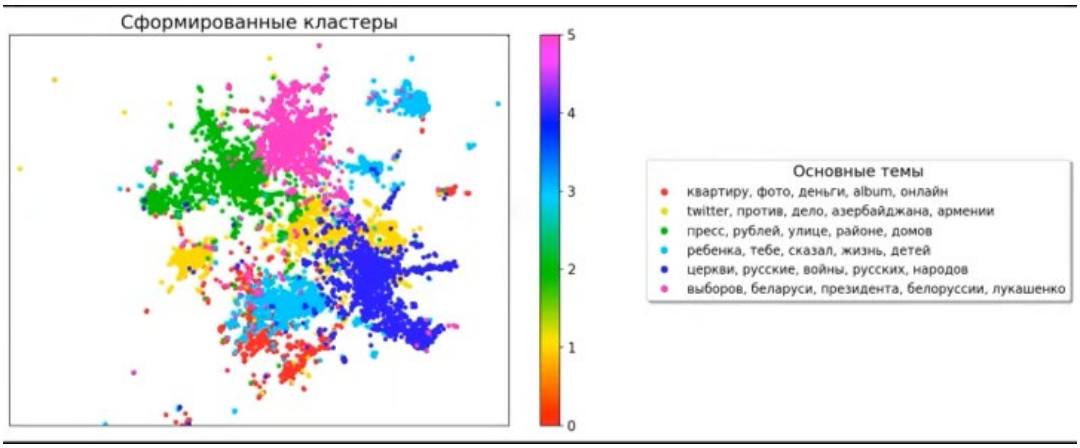


Рисунок 15.5.25 - График «Сформированные кластеры»

15. Сохранение файлов в раздел данные. В результате отработки блок схемы в разделе «Данные» должны появится следующие папки:

1) dataset_after_dbscan, в которой отображается сохраненный датасет в формате .csv. Остальные файлы являются системными

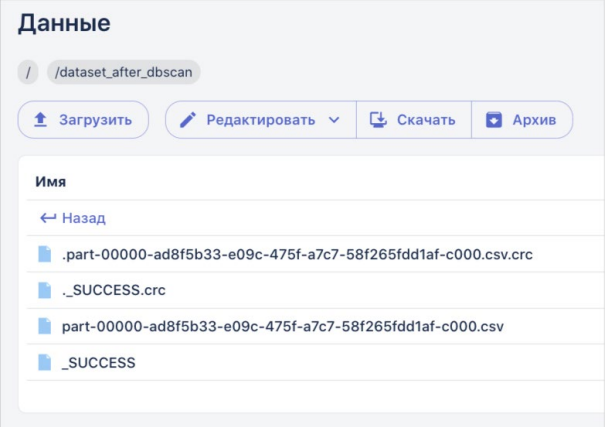


Рисунок 15.5.25 - Папка с сохраненным датасетом в разделе Данные

2) dataset_with_anomalies_after_dbscan, в которой отобразится сохраненный датасет в формате .csv. Остальные файлы являются системными:

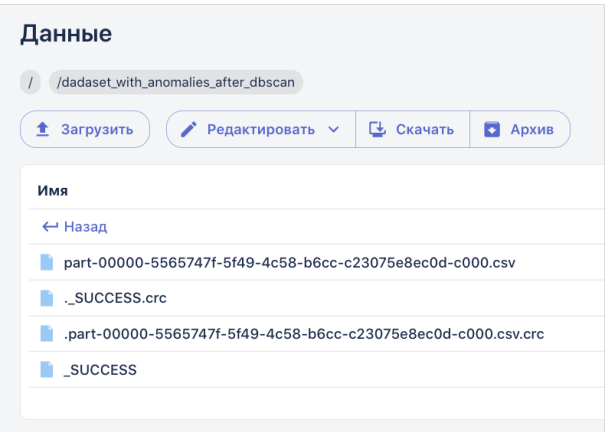


Рисунок 15.5.26 - Папка с сохраненным датасетом в разделе Данные

15.6 Классификация текстовых данных с использованием слоя нейронной сети LSTM

1. **Загрузка входных данных:**
   - 1.1. В боковом меню главного окна системы выберите пункт «Данные».
   - 1.2. Перейдите в пустую папку и затем нажмите кнопку «Загрузить» на верхней панели.
   - 1.3. В открывшемся окне нажмите кнопку «Выбрать файлы» и укажите путь к родительской папке с дочерними папками, содержащими 2 заранее подготовленных PDF-файла литературных произведений разных авторов.
   - 1.4. Нажмите на кнопку «Загрузить». Выбранная папка будет загружена.
   - 1.5. Перейдите в другую папку и затем нажмите кнопку «Загрузить» на верхней панели.
   - 1.6. В открывшемся окне нажмите кнопку «Выбрать файлы» и укажите путь к другой родительской папке с дочерними папками, содержащими 2 заранее подготовленных PDF-файла других литературных произведений этих же авторов.

1.7. Нажмите на кнопку «Загрузить». Выбранная папка будет загружена.

2. Создание новой рабочей области

2.1. В боковом меню главного окна системы выберите **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка ⊕):

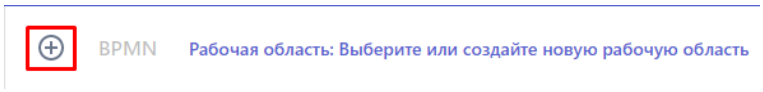


Рисунок 15.6.1 - Создание рабочей области

2.2. В открывшейся форме введите название новой рабочей области «LSTM» и нажмите кнопку «Создать»:

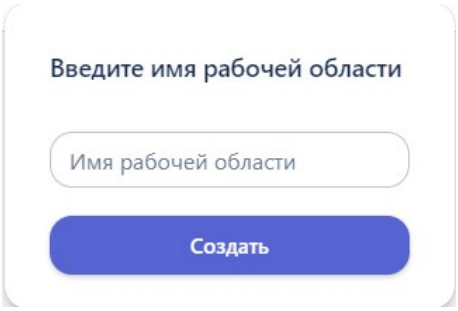


Рисунок 15.6.2 - Ввод имени рабочей области

2.3. На панели инструментов отобразится название созданной рабочей области.

3. Добавление элемента «Запуск»:

3.1. На панели инструментов блок-схемы нажмите кнопку «Добавить элемент» (кнопка BPMN).

3.2. В открывшейся библиотеке графических элементов выберите элемент «Запуск»:

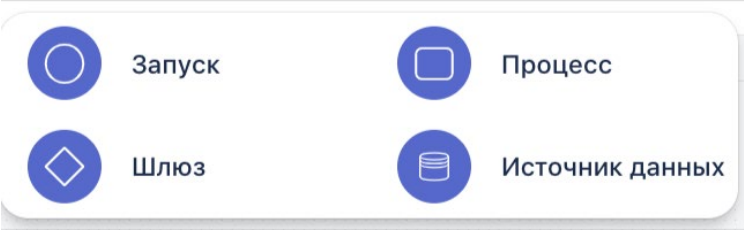


Рисунок 15.6.3 - Возможные элементы блок схемы

3.3. На рабочую область будет добавлен элемент «Запуск»:

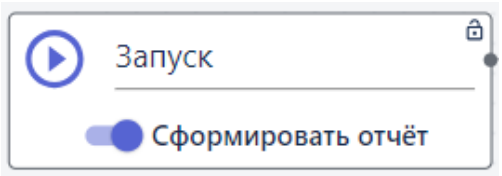


Рисунок 15.6.4 - Блок Запуск

4. Добавление и настройка элемента «Источник данных».

4.1. Добавьте на рабочую область элемент «Источник данных»:

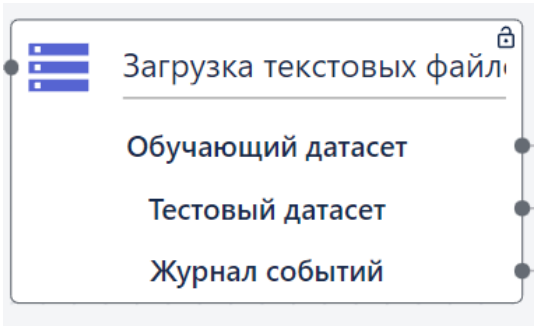


Рисунок 15.6.5 - Блок Источник данных

4.2. **Открытие настроек элемента**. На элементе «Источник данных» нажмите на кнопку . Справа откроется панель настроек элемента, где будут отображаться созданные папки и файлы с табличными данными.

4.3. **Выбор типа загрузки.** Из списка выпадающих функций выберите «Загрузка текстовых файлов для классификации.

4.4. **Выбор данных для загрузки в блок-схему**. В списках «Группа обучающих текстов» и «Группа тестовых текстов» выберите папки с файлами, загруженные в Систему в шаге 1, нажмите на три точки и кликните «Выбрать».

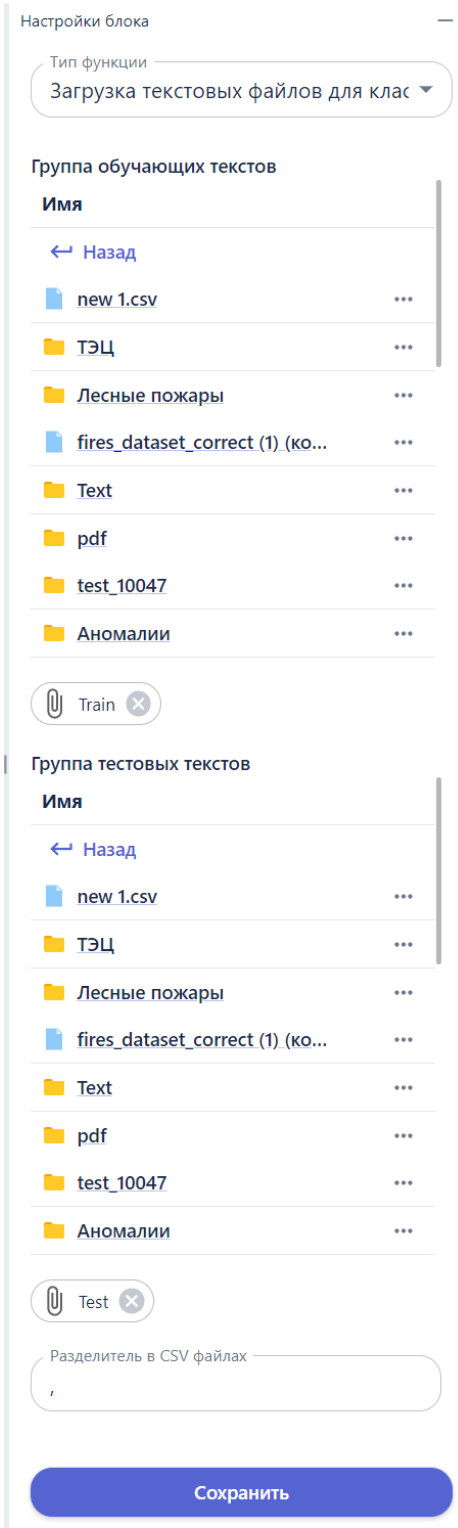


Рисунок 15.6.6 - Выбор файла

4.5. **Сохранение настроек элемента**. На панели настроек элемента нажмите на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

4.6. **Ввод названия элемента**. Чтобы задать название элемента нужно дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области, и ввести нужное название в поле с названием, доступным для редактирования:

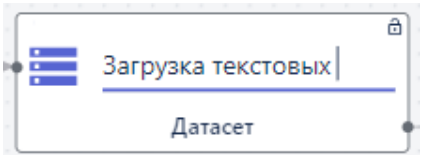


Рисунок 15.6.7 - Блок источник данных

4.7. Введите название «Загрузка текстовых данных» и кликните на пустое место на рабочей области для сохранения.

5. Добавление и настройка элемента «Процесс».

5.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настройте элемент:

5.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Фильтрация текстового шума».

5.1.2. Нажать кнопку «Сохранить».

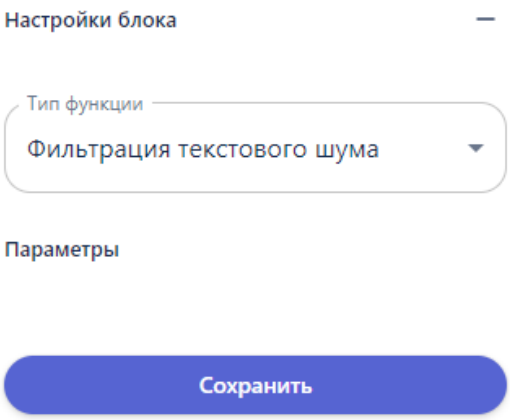


Рисунок 15.6.8 - Настройка блока «Процесс»

6. Добавление и настройка элемента «Процесс».

6.1. Повторите действия, описанные на шаге выше, чтобы продублировать созданный элемент «Процесс».

6.2. Настройте элемент, как указано на шаге выше.

7. Добавление и настройка элемента «Процесс».

7.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настройте элемент:

7.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Лемматизация текста».

7.1.2. Нажать кнопку «Сохранить».

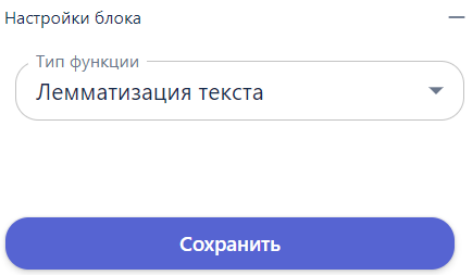


Рисунок 15.6.9 - Блок процесс

8. Добавление и настройка элемента «Процесс».

8.1. Повторите действия, описанные на шаге выше, чтобы продублировать созданный элемент «Процесс».

8.2. Настройте элемент, как указано на шаге выше.

9. Добавление и настройка элемента «Процесс».

9.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настроить элемент:

9.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Векторизация текста».

9.1.2. В поле «Максимальная размерность текста» выберите 10000.

9.1.3. В поле «Количество признаков» выберите 10.

9.1.4. Нажать кнопку «Сохранить».

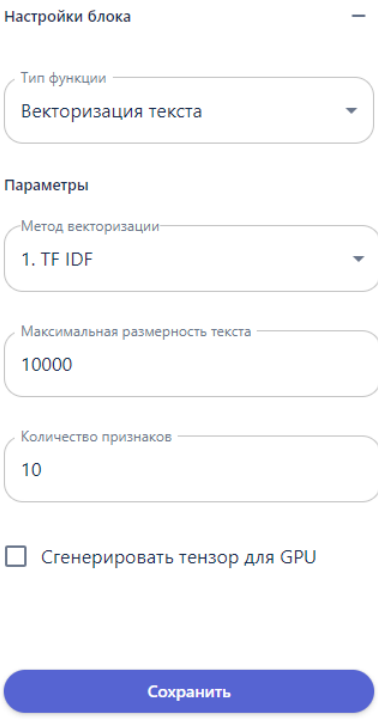


Рисунок 15.6.10 - Блок процесс

10. Выбор функции и настройка параметров:

10.1. Добавить элемент «Процесс» (кнопка ).

10.2. В карточке элемента выбрать из списка функцию: раздел «Глубокое обучение» -> «Классификация» -> функция «Классификация (табличные данные)».

10.3. В секции “Добавить слой” нажать кнопку + и затем в поле “Слой” выбрать значение “4. LSTM”

10.4. В поле “Число нейронов” ввести количество нейронов, соответствующее размерности текста.

10.5. В поле “Функция активации” выбрать значение “Sigmoid” (требуемое значение рекомендуется выбирать практическим путем).

10.6. В поле “Функция активации (рекурсия)” выбрать значение “Sigmoid” (требуемое значение рекомендуется выбирать практическим путем).

10.7. В секции “Добавить слой” нажать кнопку + и затем в поле “Слой” выбрать значение “1. Dense”.

10.8. В поле “Число нейронов” ввести 1.

10.9. В поле “Функция активации” выбрать значение “Sigmoid” (требуемое значение рекомендуется выбирать практическим путем).

10.10. В поле “Количество эпох” установить значение 5.

10.11. В поле “Размер мини-батча” установить значение 16.

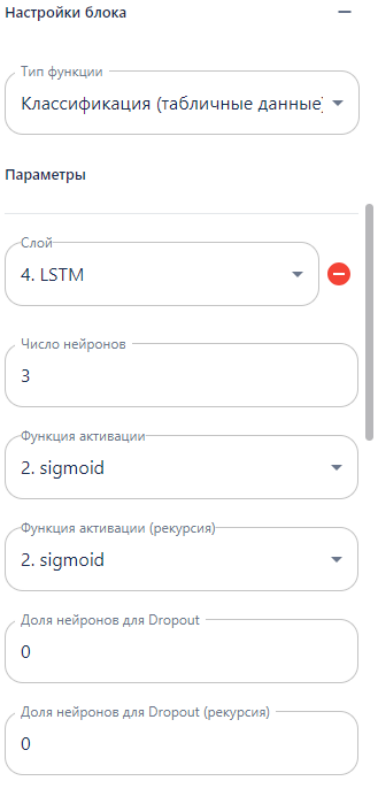
10.12. В поле “Алгоритм градиентного спуска” выбрать значение “3. Adam”.

10.13. В поле “Шаг градиентного спуска” установить значение 0.001.

10.14. В поле “Порог классификации” задать значение 0.5.

10.15. Значения остальных параметров оставить без изменения.

10.16. Нажать кнопку “Сохранить”.



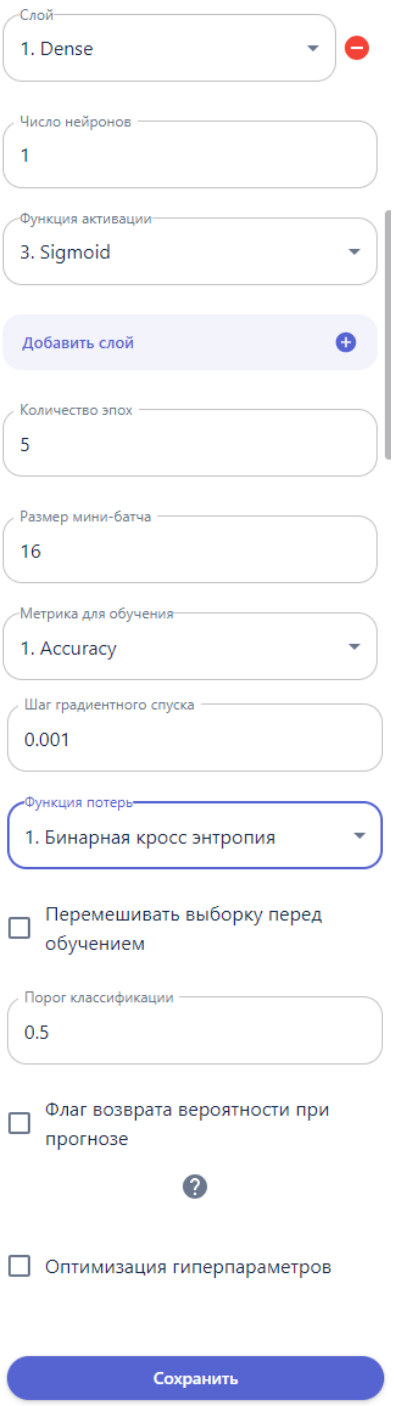


Рисунок 15.6.11 - Блок процесс

11. Добавление и настройка элемента «Процесс».

11.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настроить элемент:

11.1.1. В карточке элемента в списке «Тип функции» в разделе «Машинное обучение» выбрать «Валидация модели».

11.1.2. В поле «Метрика» выбрать «5. Accuracy».

11.1.3. Нажать кнопку «Сохранить».

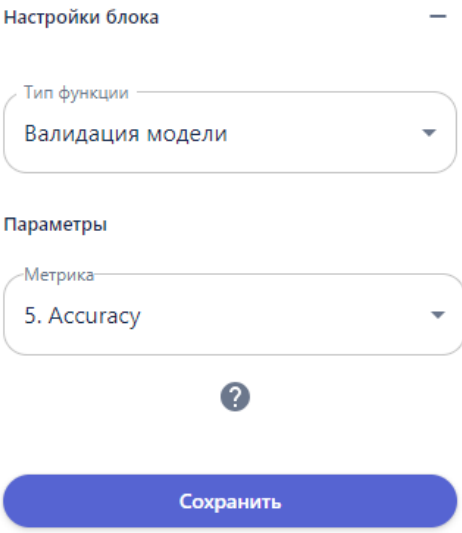


Рисунок 15.6.12 - Блок процесс

12. **Установка соединений.** Соедините выходные и входные точки элементов, как показано на рисунке ниже.

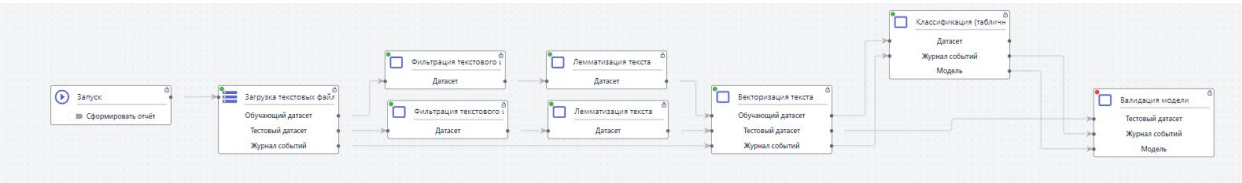


Рисунок 15.6.13 - Установка соединений между блоками

13. **Запуск пайплайна**. Чтобы запустить блок схему нажмите на кнопку на первом элементе «Запуск» собранной блок-схемы. При этом отображение элемента «Запуск» изменится и появится опция «Сформировать отчет»:

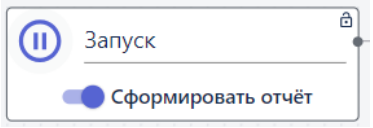


Рисунок 15.6.14 - Блок Запуск

Если активировать параметр «Сформировать отчет», в результате запуска пайплайна будет создан отчет.

14. **Визуализация результатов**. После того как все элементы схемы будут успешно обработаны, на панели инструментов появляются кнопки:

Нажмите на вторую слева кнопку и выберите команду «Отфильтрованные текстовые данные». Отобразится таблица с результатами обработки.

15.7 Извлечение текстового слоя из текстовых данных

Данный сценарий предполагает извлечение текстового слоя из текстовых данных в формате PDF.

Для решения задачи выполните следующие действия:

15. **Загрузка входных данных:**

15.1. В боковом меню главного окна системы выберите пункт «Данные».

15.2. Перейдите в пустую папку и затем нажмите кнопку «Загрузить» на верхней панели.

15.3. В открывшемся окне нажмите кнопку «Выбрать файлы» и укажите путь к родительской папке с дочерними папками, содержащими 2 заранее подготовленных PDF-файла литературных произведений разных авторов.

15.4. Нажмите на кнопку «Загрузить». Выбранная папка будет загружена.

15.5. Перейдите в другую папку и затем нажмите кнопку «Загрузить» на верхней панели.

15.6. В открывшемся окне нажмите кнопку «Выбрать файлы» и укажите путь к другой родительской папке с дочерними папками, содержащими 2 заранее подготовленных PDF-файла других литературных произведений этих же авторов.

15.7. Нажмите на кнопку «Загрузить». Выбранная папка будет загружена.

16. Создание новой рабочей области

16.1. В боковом меню главного окна системы выберите **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка ⊕):

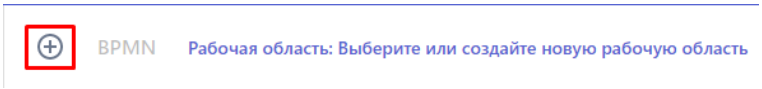


Рисунок 15.7.1 - Создание рабочей области

16.2. В открывшейся форме введите название новой рабочей области «Классификация текстов (new lib)» и нажмите кнопку «Создать»:

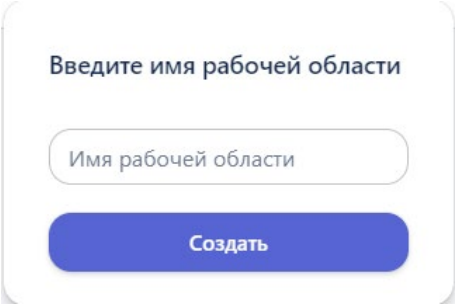


Рисунок 15.7.2 - Ввод имени рабочей области

16.3. На панели инструментов отобразится название созданной рабочей области.

17. Добавление элемента «Запуск»:

17.1. На панели инструментов блок-схемы нажмите кнопку «Добавить элемент» (кнопка BPMN).

17.2. В открывшейся библиотеке графических элементов выберите элемент «Запуск»:

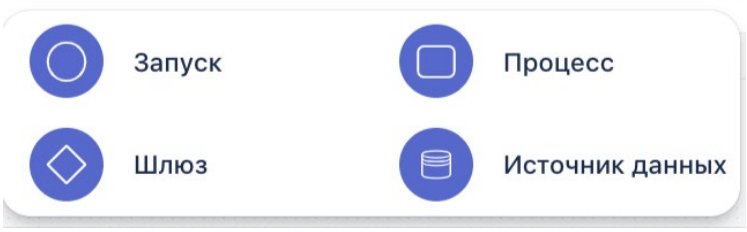


Рисунок 15.7.3 - Возможные элементы блок схемы

17.3. На рабочую область будет добавлен элемент «Запуск»:

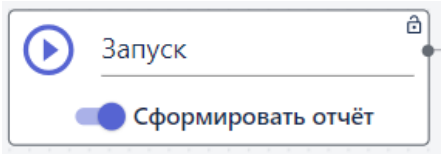


Рисунок 15.7.4 - Блок Запуск

18. Добавление и настройка элемента «Источник данных».

18.1. Добавьте на рабочую область элемент «Источник данных»:

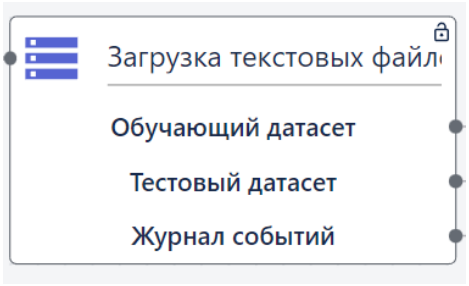


Рисунок 15.7.5 - Блок Источник данных

18.2. **Открытие настроек элемента**. На элементе «Источник данных» нажмите на кнопку  . Справа откроется панель настроек элемента, где будут отображаться созданные папки и файлы с табличными данными.

18.3. **Выбор типа загрузки.** Из списка выпадающих функций выберите «Загрузка текстовых файлов для классификации.

18.4. **Выбор данных для загрузки в блок-схему**. В списках «Группа обучающих текстов» и «Группа тестовых текстов» выберите папки с файлами, загруженные в Систему в шаге 1, нажмите на три точки и кликните «Выбрать».

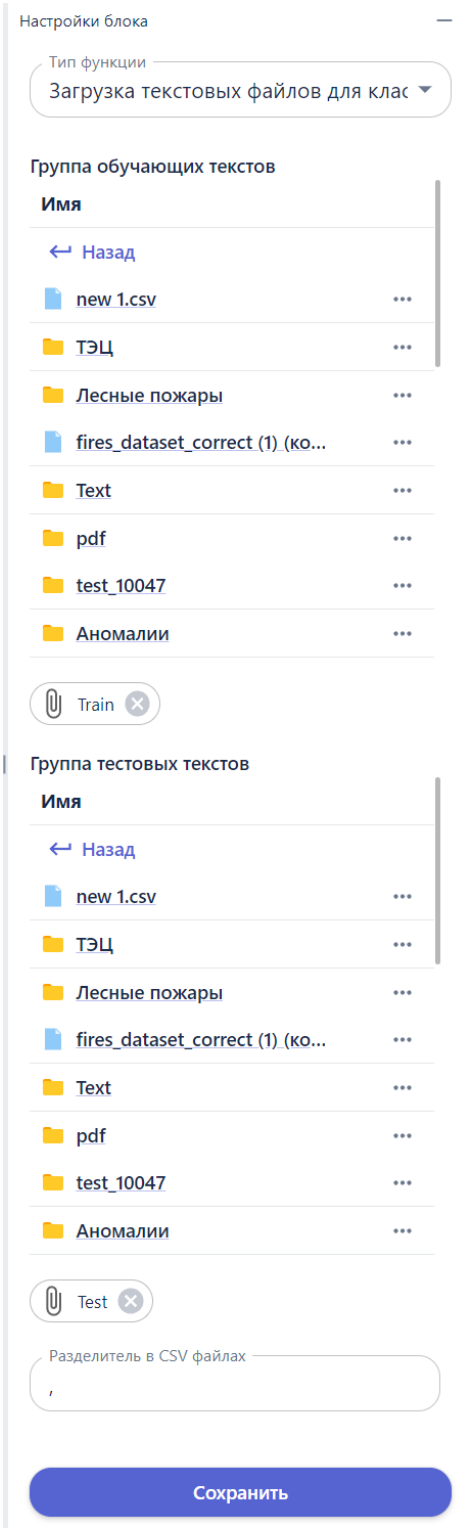


Рисунок 15.7.6 - Выбор файла

18.5. **Сохранение настроек элемента**. На панели настроек элемента нажмите на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

18.6. **Ввод названия элемента**. Чтобы задать название элемента нужно дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области, и ввести нужное название в поле с названием, доступным для редактирования:

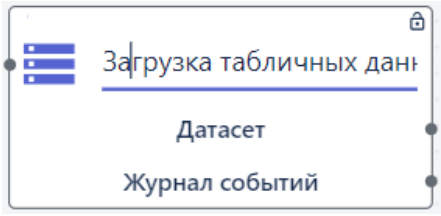


Рисунок 15.8.7 - Блок источник данных

18.7. Введите название «Загрузка текстовых данных» и кликните на пустое место на рабочей области для сохранения.

19. Добавление и настройка элемента «Процесс».

19.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настройте элемент:

19.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Фильтрация текстового шума».

19.1.2. Нажать кнопку «Сохранить».

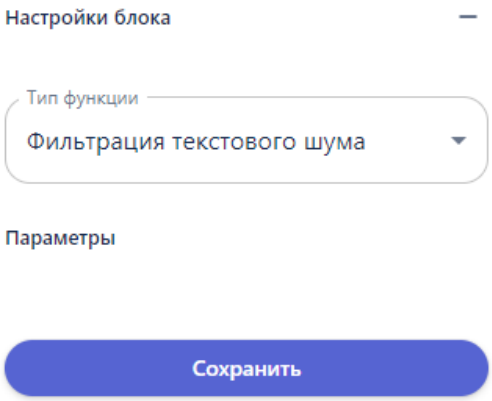


Рисунок 15.8.8 - Настройка блока «Процесс»

20. Добавление и настройка элемента «Процесс».

20.1. Повторите действия, описанные на шаге выше, чтобы продублировать созданный элемент «Процесс».

20.2. Настройте элемент, как указано на шаге выше.

21. Добавление и настройка элемента «Процесс».

21.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настройте элемент:

21.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Лемматизация текста».

21.1.2. Нажать кнопку «Сохранить».

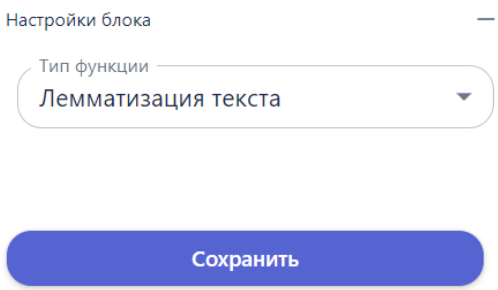


Рисунок 15.8.9 - Блок процесс

22. Добавление и настройка элемента «Процесс».

22.1. Повторите действия, описанные на шаге выше, чтобы продублировать созданный элемент «Процесс».

22.2. Настройте элемент, как указано на шаге выше.

23. Добавление и настройка элемента «Процесс».

23.1. Добавьте на рабочую область элемент «Процесс» (кнопка ). Настроить элемент:

23.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Векторизация текста».

23.1.2. В поле «Максимальная размерность текста» выберите 10000.

23.1.3. В поле «Количество признаков» выберите 10.

23.1.4. Нажать кнопку «Сохранить».

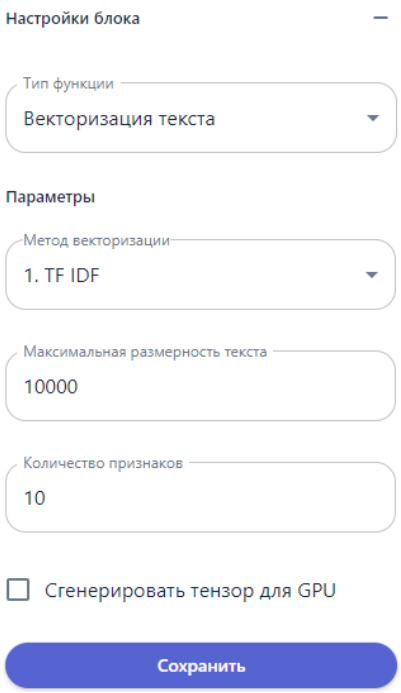


Рисунок 15.8.10 - Блок процесс

24. Добавление и настройка элемента «Процесс».

24.1. Добавьте на рабочую область элемент «Процесс» (кнопка ). Настроить элемент:

24.1.1. В карточке элемента в списке «Тип функции» в разделе «Классификация» выбрать «Логистическая регрессия».

24.1.2. В поле «Коэффициент регуляризации» выберите 1.

24.1.3. В поле «Порог классификации» выберите 0,5.

24.1.4. Нажать кнопку «Сохранить».

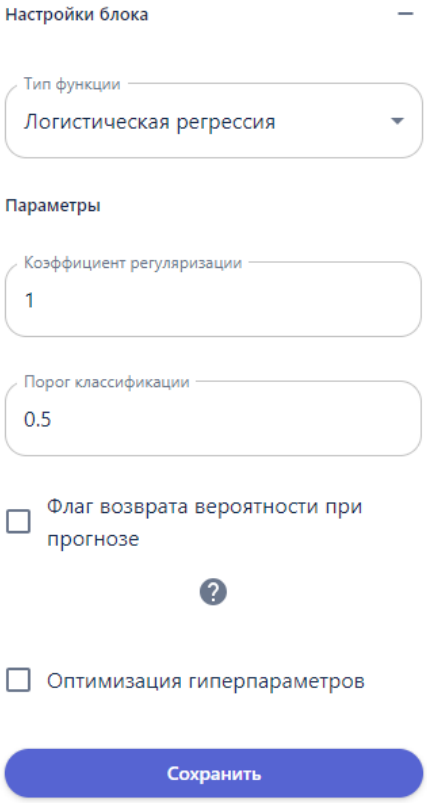


Рисунок 15.8.11 - Блок процесс

25. Добавление и настройка элемента «Процесс».

25.1. Добавьте на рабочую область элемент «Процесс» (кнопка ). Настроить элемент:

25.1.1. В карточке элемента в списке «Тип функции» в разделе «Классификация» выбрать «Модель XGBClassifier».

25.1.2. В поле «Глубина дерева» выберите 2.

25.1.3. В поле «Количество базовых моделей» выберите 100.

25.1.4. В поле «Порог классификации» выберите 0,5.

25.1.5. Нажать кнопку «Сохранить».

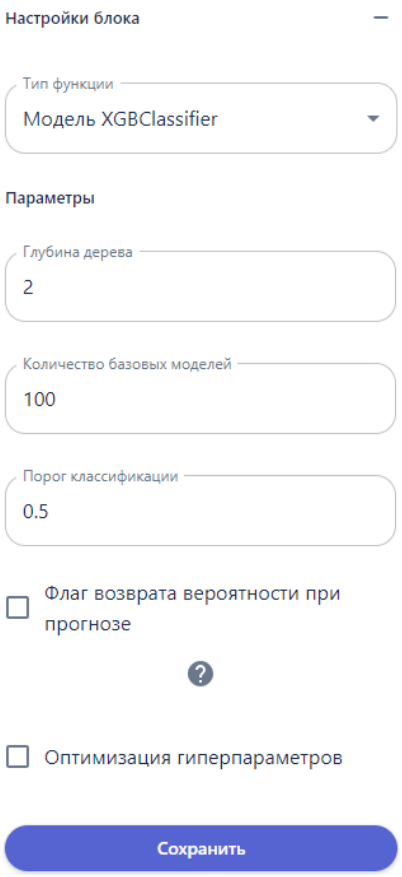


Рисунок 15.8.12 - Блок процесс

26. Добавление и настройка элемента «Процесс».

26.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настроить элемент:

26.1.1. В карточке элемента в списке «Тип функции» в разделе «Машинное обучение» выбрать «Валидация модели».

26.1.2. В поле «Метрика» выбрать «5. Accuracy».

26.1.3. Нажать кнопку «Сохранить».

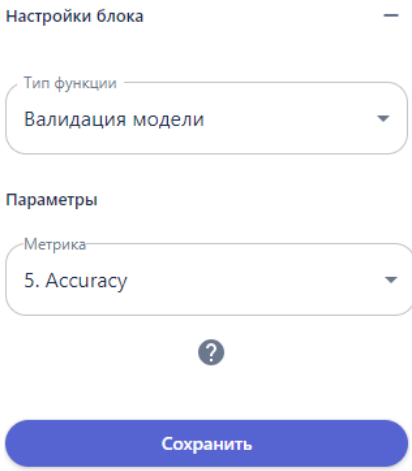


Рисунок 15.8.12 - Блок процесс

27. **Установка соединений.** Соедините выходные и входные точки элементов, как показано на рисунке ниже.

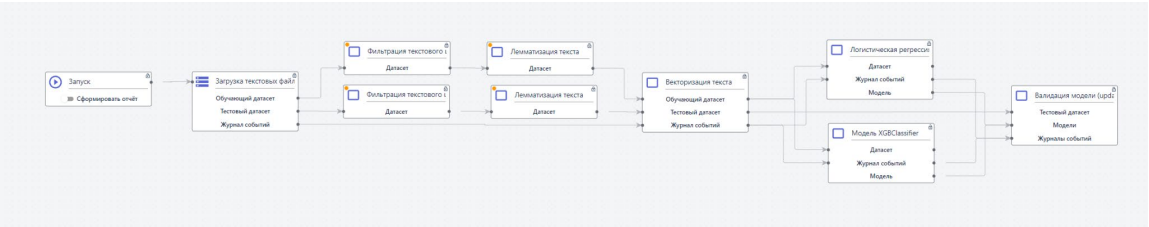


Рисунок 15.8.13 - Установка соединений между блоками

28. **Запуск пайплайна**. Чтобы запустить блок схему нажмите на кнопку на первом элементе «Запуск» собранной блок-схемы. При этом отображение элемента «Запуск» изменится и появится опция «Сформировать отчет»:

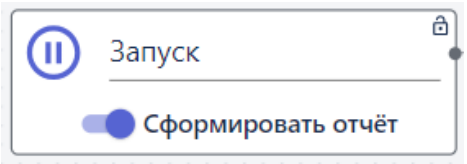


Рисунок 15.8.14 - Блок Запуск

Если активировать параметр «Сформировать отчет», в результате запуска пайплайна будет создан отчет.

29. **Визуализация результатов**. После того как все элементы схемы будут успешно обработаны, на панели инструментов появляются кнопки:

Нажмите на вторую слева кнопку и выберите команду «Отфильтрованные текстовые данные». Отобразится таблица с результатами обработки.

15.8 Заполнение и работа с пропусками в табличных данных

Данный сценарий предполагает работу с пропущенными значениями или пропусками в табличных данных в формате CSV. Система находит пропуски в указанных столбцах и удаляет строки, в которых содержатся пропуски.

Для решения задачи выполните следующие действия:

1. **Загрузка входных данных:**
   - 1.1. В боковом меню главного окна системы выберите пункт «Данные».
   - 1.2. Нажмите кнопку «Загрузить» на верхней панели.
   - 1.3. В открывшемся окне нажмите кнопку «Выбрать файлы» и укажите путь к заранее подготовленному файлу **train.csv**, в котором содержатся данные с пропусками.
   - 1.4. Нажмите на кнопку «Загрузить». Файл с входными данными отобразится в папке.
2. **Создание новой рабочей области**
   - 2.1. В боковом меню главного окна системы выберите **Моделирование** –> **Рабочая область**. На панели инструментов блок-схемы нажмите кнопку «Создание рабочей области» (кнопка):

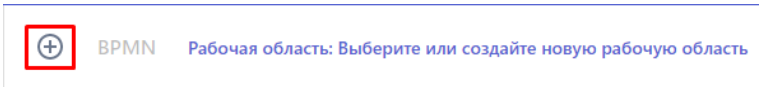


Рисунок 15.8.1 - Создание рабочей области

2.2. В открывшейся форме введите название новой рабочей области «Заполнение пропусков (new lib)» и нажмите кнопку «Создать»:

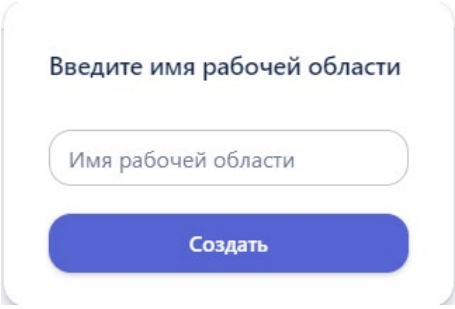


Рисунок 15.8.2 - Ввод имени рабочей области

2.3. На панели инструментов отобразится название созданной рабочей области.

3. Добавление элемента «Запуск»:

3.1. На панели инструментов блок-схемы нажмите кнопку «Добавить элемент» (кнопка BPMN).

3.2. В открывшейся библиотеке графических элементов выберите элемент «Запуск»:

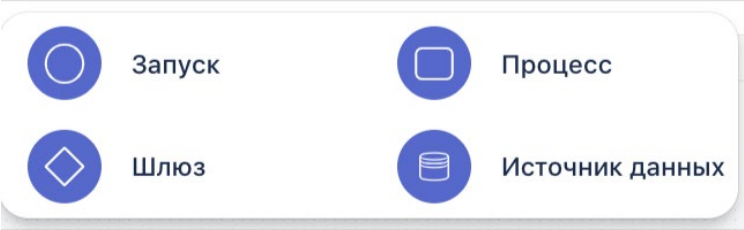


Рисунок 15.8.3 - Возможные элементы блок схемы

3.3. На рабочую область будет добавлен элемент «Запуск»:

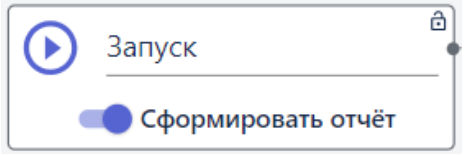


Рисунок 15.8.4 - Блок Запуск

4. Добавление и настройка элемента «Источник данных».

4.1. Добавьте на рабочую область элемент «Источник данных»:

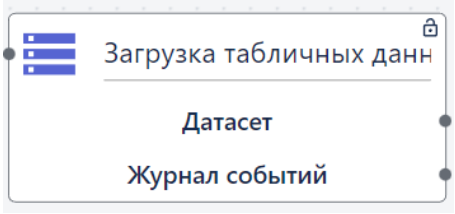


Рисунок 15.8.5 - Блок Источник данных

4.2. **Открытие настроек элемента**. На элементе «Источник данных» нажмите на кнопку  . Справа откроется панель настроек элемента, где будут отображаться созданные папки и файлы с табличными данными.

4.3. **Выбор типа загрузки.** Из списка выпадающих функций выберите «Загрузка табличных данных».

4.4. **Выбор данных для загрузки в блок-схему**. Чтобы найти нужный файл, кликните на папку и перейдите в нее, выберите из списка файл, загруженный в Систему в шаге 1 «train.csv», нажмите на три точки в строке с ним и кликните «Выбрать». Внизу отобразится название выбранного файла:

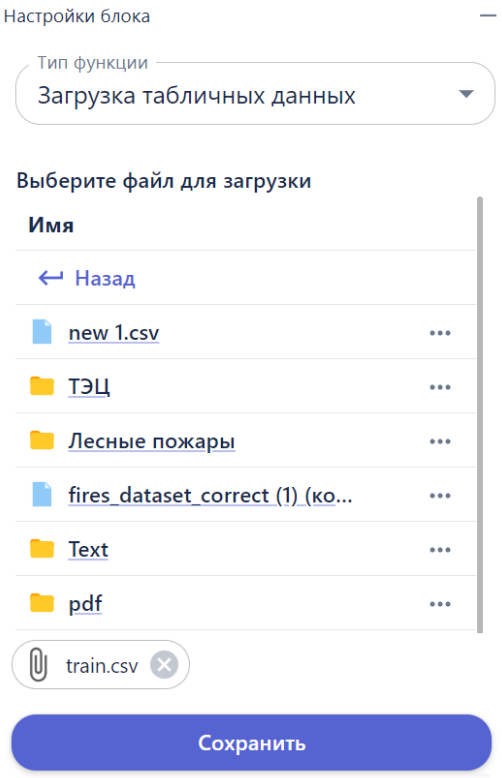


Рисунок 15.8.6 - Выбор файла

4.5. **Сохранение настроек элемента**. На панели настроек элемента нажмите на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

4.6. **Ввод названия элемента**. Чтобы задать название элемента нужно дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области, и ввести нужное название в поле с названием, доступным для редактирования:

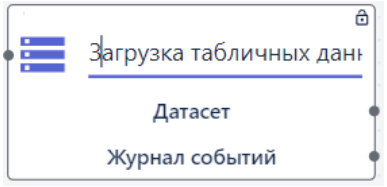


Рисунок 15.8.7 - Блок источник данных

4.7. Введите название «Загрузка табличных данных» и кликните на пустое место на рабочей области для сохранения.

5. Добавление и настройка элемента «Процесс».

5.1. Добавьте на рабочую область элемент «Процесс» (кнопка ). Настройте элемент:

5.1.1. В карточке элемента в списке «Тип функции» в разделе «Анализ данных» выбрать «Поиск пропущенных значений».

5.1.2. В поле «Признаки» оставить нужные признаки (поля таблицы, в которых будет выполнен поиск пропущенных значений).

5.1.3. Нажать кнопку «Сохранить».

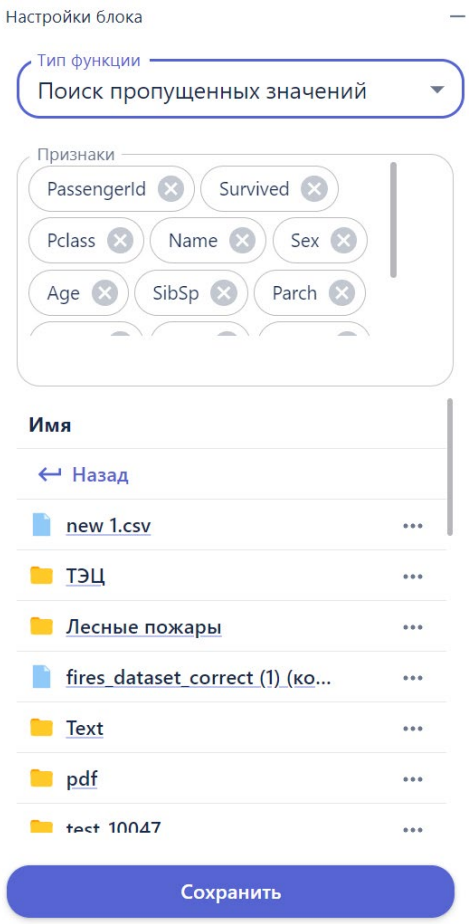


Рисунок 15.8.8 - Блок процесс

6. Добавление и настройка элемента «Процесс».

6.1. Добавьте на рабочую область элемент «Процесс» (кнопка ). Настроить элемент:

15.1.1. В карточке элемента в списке «Тип функции» в разделе «Анализ данных» выбрать «Выбор признаков и целевых признаков».

15.1.2. В поле «Признаки» оставить нужные общие признаки.

15.1.3. В поле «Целевые признаки» оставить нужные целевые признаки.

15.1.4. Нажать кнопку «Сохранить».

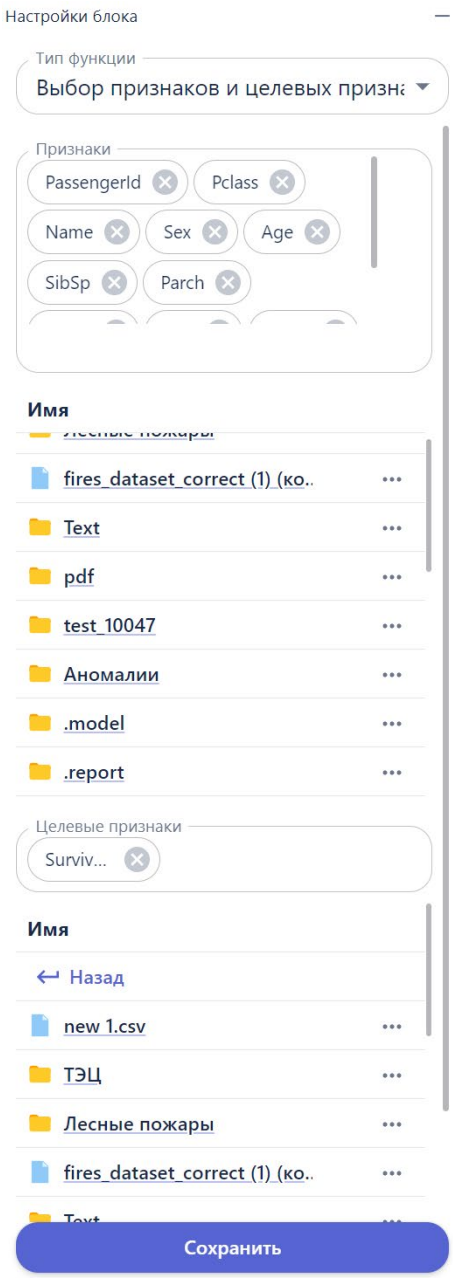


Рисунок 15.8.9 - Блок источник данных

7. Добавление и настройка элемента «Процесс».

7.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настроить элемент:

7.1.1. В карточке элемента в списке «Тип функции» в разделе «Предобработка данных» выбрать «Заполнение пропусков».

7.1.2. В разделе «Имя» выбрать файл в формате CSV и выполнить команду «Выгрузить признаки».

7.1.3. В поле «Признак» оставить нужные признаки.

7.1.4. В списке «Метод заполнения пропусков» выбрать следующий методов частичного заполнения пропусков:

- «Удалить строки» - удаление всех строк с пропусками.

7.1.5. Нажать кнопку «Сохранить».

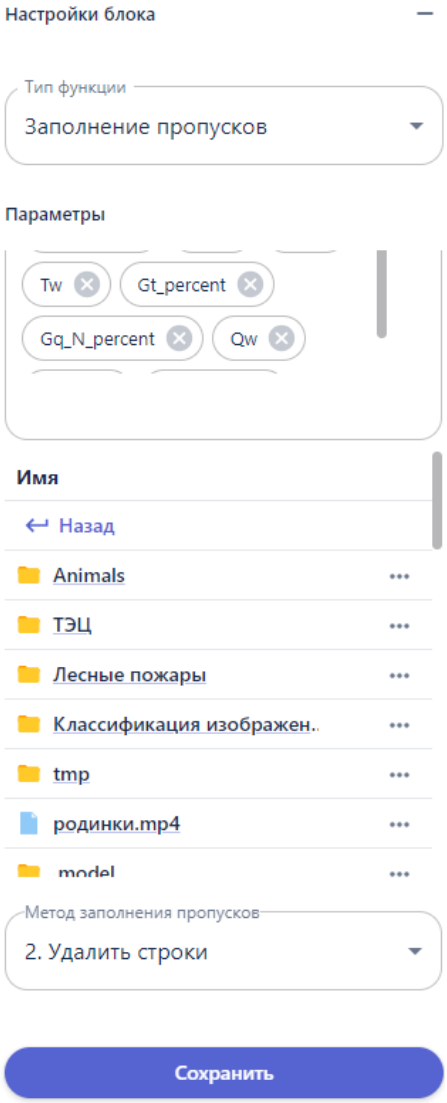


Рисунок 15.8.10 - Блок Процесс

8. Добавление и настройка элемента «Процесс».

8.1. Добавьте на рабочую область элемент «Процесс» (кнопка). Настроить элемент:

8.2. В карточке элемента в списке «Тип функции» в разделе «Анализ данных» выбрать «Поиск пропущенных значений».

15.1.5. В разделе «Имя» выбрать файл в формате CSV и выполнить команду «Выгрузить признаки».

15.1.6. В поле «Признаки» оставить нужные признаки.

15.1.7. Нажать кнопку «Сохранить».

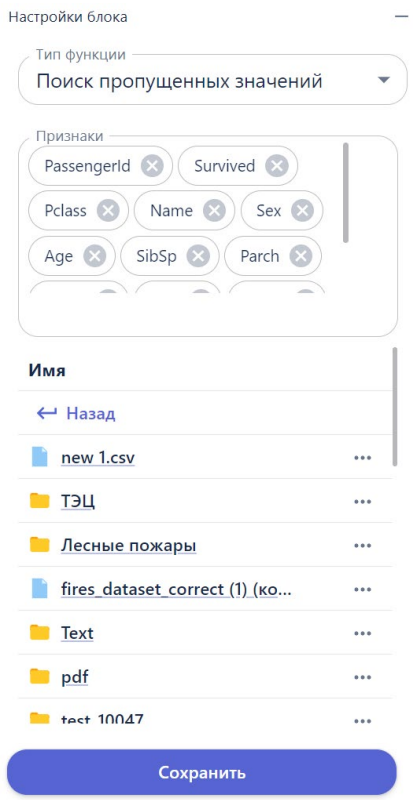


Рисунок 15.8.11 - Блок Процесс

9. **Установка соединений.** Соедините выходные и входные точки элементов, как показано на рисунке ниже.

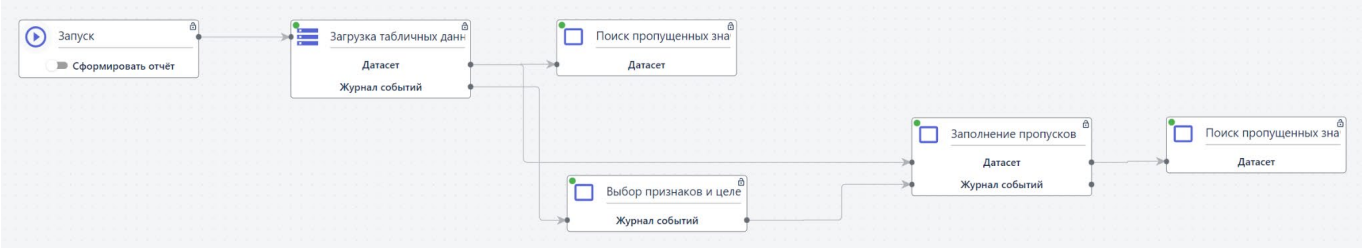


Рисунок 15.8.12 - Установка соединений

10. **Запуск пайплайна**. Чтобы запустить блок схему нажмите на кнопку на первом элементе «Запуск» собранной блок-схемы. При этом отображение элемента «Запуск» изменится и появится опция «Сформировать отчет»:

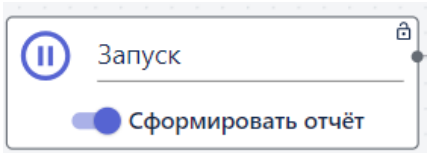


Рисунок 15.8.13 - Блок Запуск

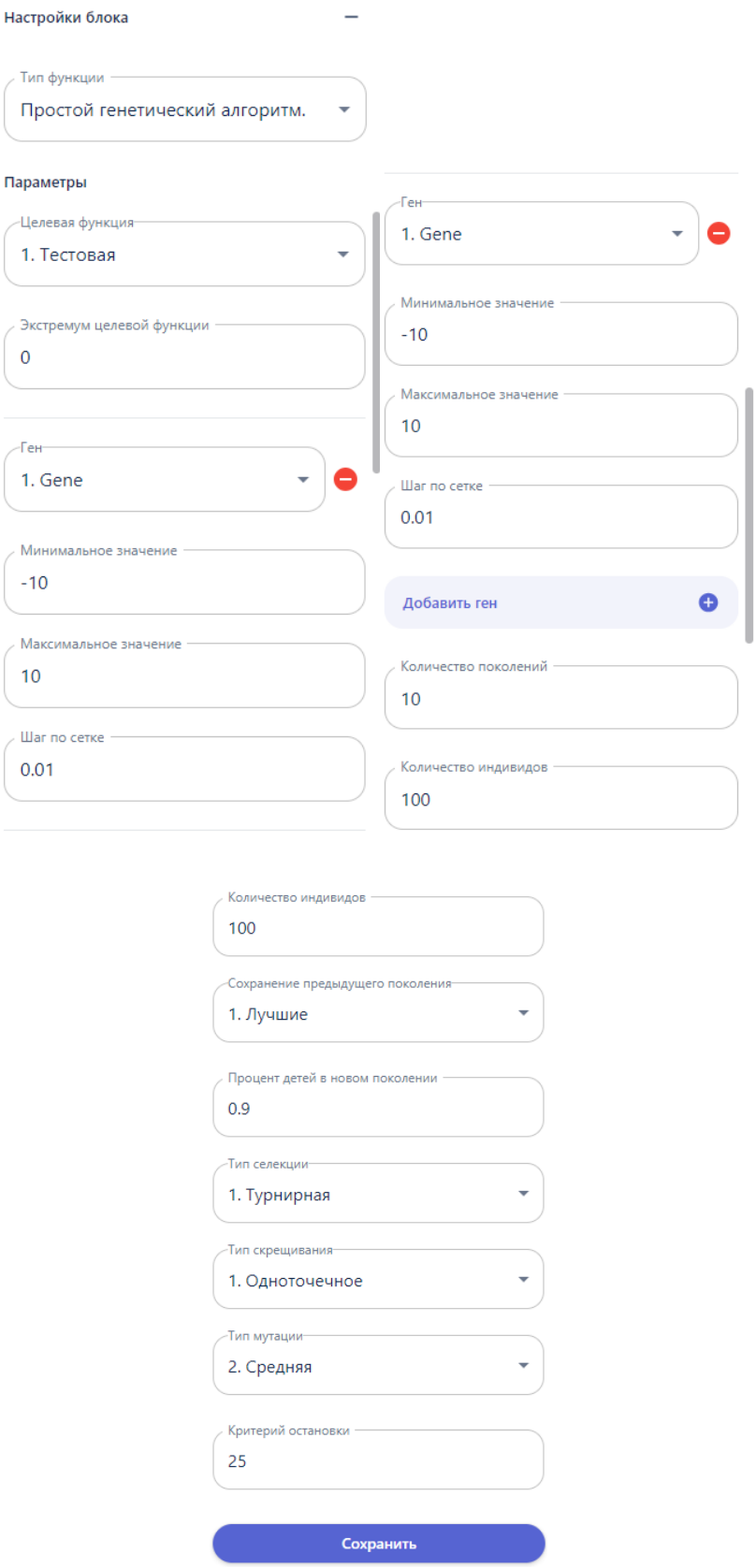
Если активировать параметр «Сформировать отчет», в результате запуска пайплайна будет создан отчет.

11. **Визуализация результатов**. После того как все элементы схемы будут успешно обработаны, на панели инструментов появляются кнопки:

Нажмите на вторую слева кнопку и выберите команду «Количество пропусков». Отобразится таблица с результатами обработки.

15.9 Использование генетического алгоритма

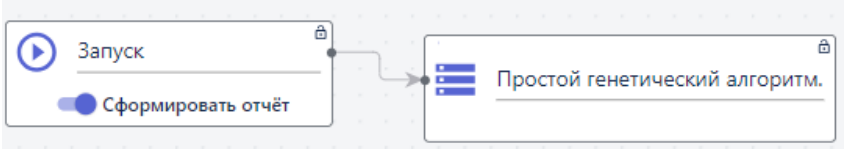
1. **Создание блок-схемы**. Перейти в пункт меню **Моделирование** -> **Рабочая область**. На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ). В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать». На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элемента «Запуск»**. На панели инструментов блок-схемы нажать кнопку «BPMN» и в открывшемся окне выбрать элемент «Запуск» (кнопка ). Выбранный элемент появится на рабочей области (далее это действие предполагается по умолчанию). Кнопка «Запуск» обозначает начало блок-схемы.
3. **Добавление и настройка элемента «Генетический алгоритм»**.
   - На рабочую область нужно добавить элемент «Генетический алгоритм» (кнопка ).
   - На элементе нажать на кнопку  (далее открытие настроек элемента предполагается по умолчанию). Откроется панель настроек элемента.
   - В поле **Тип функции** выбрать **Оптимизация -> Простой генетический алгоритм**.
   - В поле **Целевая функция** выбрать **1. Тестовая**.
   - В поле **Добавить ген** выбрать **1. Gene**.
   - В поле **Минимальное значение** выбрать **-** 10.
   - В поле **Максимальное значение** выбрать 10.
   - В поле **Шаг по сетке** выбрать 0,01.
   - В поле **Ген** выбрать **1. Gene**.
   - В поле **Минимальное значение** выбрать **-** 10.
   - В поле **Максимальное значение** выбрать 10.
   - В поле **Шаг по сетке** выбрать 0,01.
   - В поле **Количество поколений** выбрать 10.
   - В поле **Количество индивидов** выбрать 100.
   - В поле **Сохранение предыдущего поколения** выбрать 1. Лучшие.
   - В поле **Процент детей в новом поколении** выбрать 0.9.
   - В поле **Тип селекции** выбрать **1. Турнирная**.
   - В поле **Тип скрещивания** выбрать **1. Одноточечное**.
   - В поле **Тип мутации** выбрать **2. Средняя**.
   - В поле **Критерий остановки** указать 25.
   - Задать настройки, как показано на рисунках ниже.



- Нажать на кнопку «Сохранить» (далее сохранение настроек элемента предполагается по умолчанию).

4. **Задать название элемента.** Для этого дважды щелкнуть левой кнопкой мыши на название элемента в рабочей области. Ввести нужное название в поле с названием, доступным для редактирования. Чтобы новое название сохранилось достаточно щелкнуть мышью в любом месте на рабочей области.

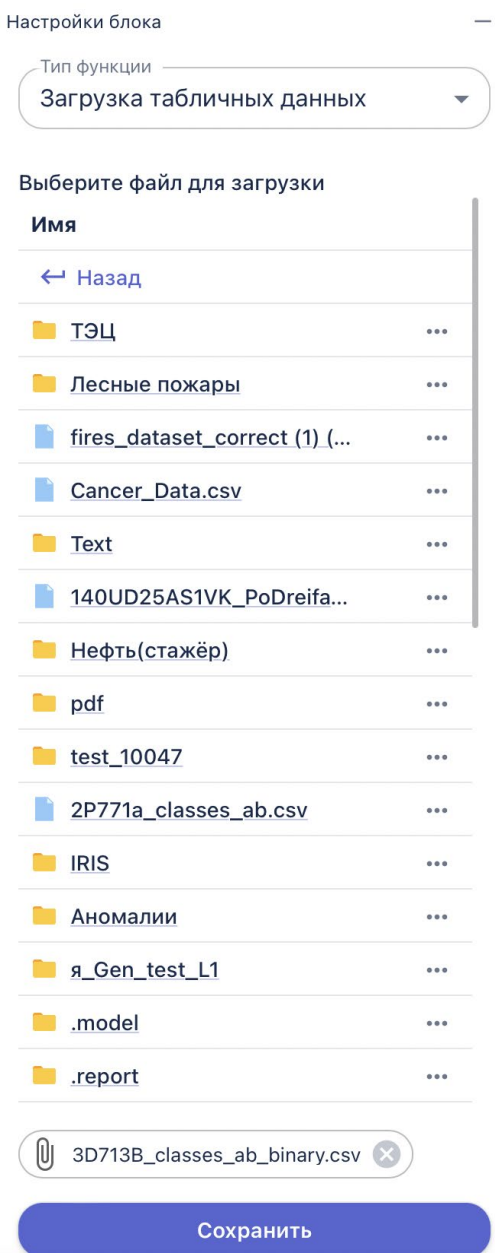
5. **Соединить выходную точку элемента «Запуск» с входной точкой элемента «Простой генетический алгоритм»**:



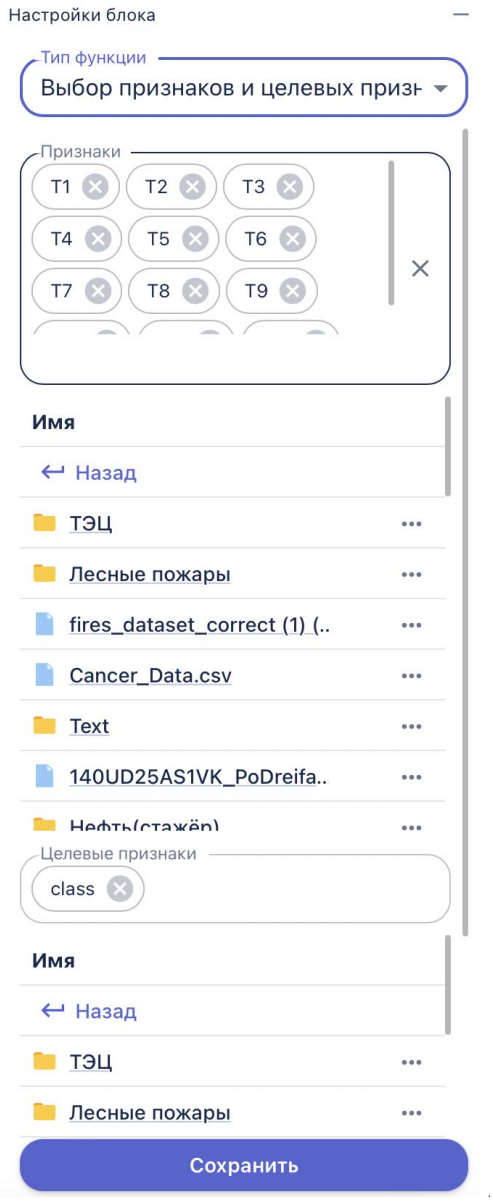
- Для соединения точек элементов используется левая кнопка мыши.

15.10 Выполнение логического анализа данных

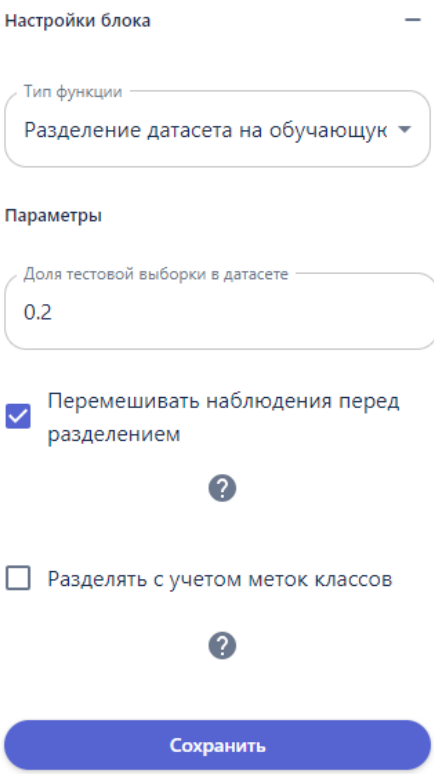
1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка табличных данных**.
     – Выбрать файл для загрузки в формате CSV.
     – Сохранить изменения.



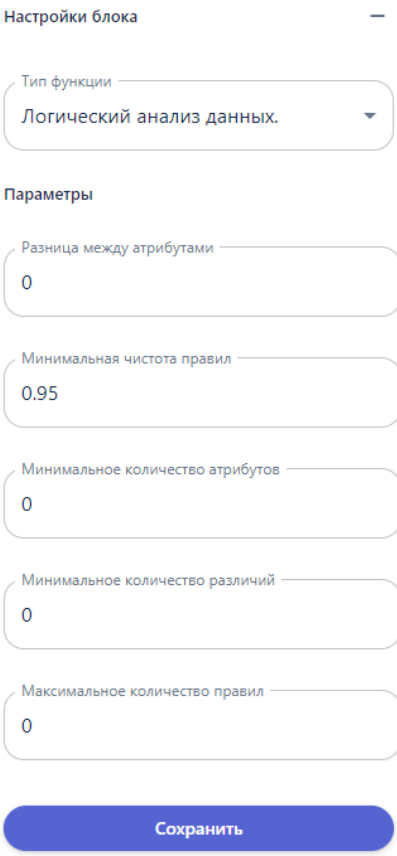
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
  – В поле **Признаки** выбрать нужные признаки для обучения.
  – В поле **Целевые признаки** выбрать нужные целевые признаки.
  – Сохранить изменения.



- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Разделение датасета на обучающую и тестовую выборки**.
  – В поле **Доля тестовой выборки в датасете** ввести нужное значение.
  – При необходимости установить флажок для параметра «**Перемешивать наблюдения перед разделением»**.
  – При необходимости установить флажок для параметра «**Разделять с учетом меток классов».**
  – Сохранить изменения.

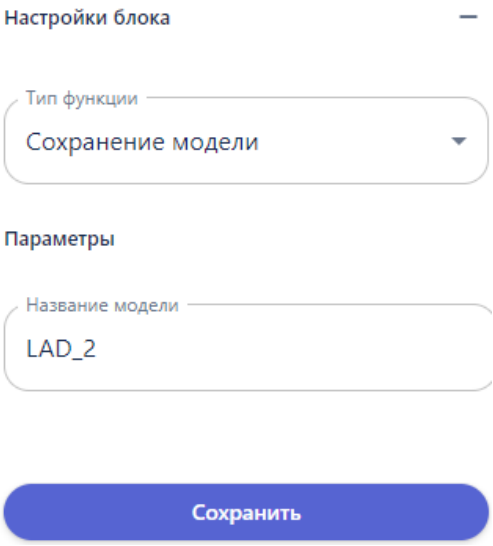


- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Классификация -> Логический анализ данных**.
  – В поле **Минимальная частота правил** ввести 0,95.
  – Сохранить изменения.

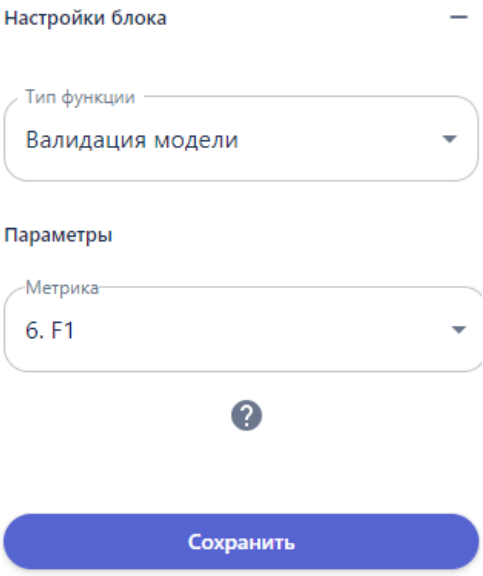


- Настройка блока **Процесс**:

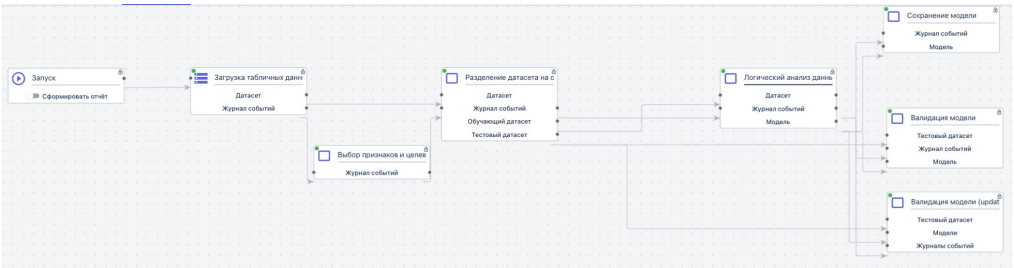
– В поле **Тип функции** выбрать **Управление моделями -> Сохранение модели**.
– В поле **название модели** ввести желаемое название.
– Сохранить изменения.



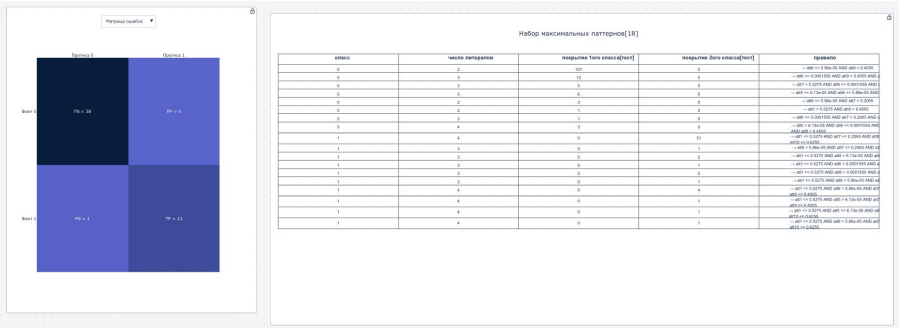
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Валидация модели**.
  – В поле **Метрика** выбрать **6. F1**.
  – Сохранить изменения.



4. **Соединить элементы друг с другом как показано на рисунке ниже**:
   - Для соединения точек элементов используется левая кнопка мыши.



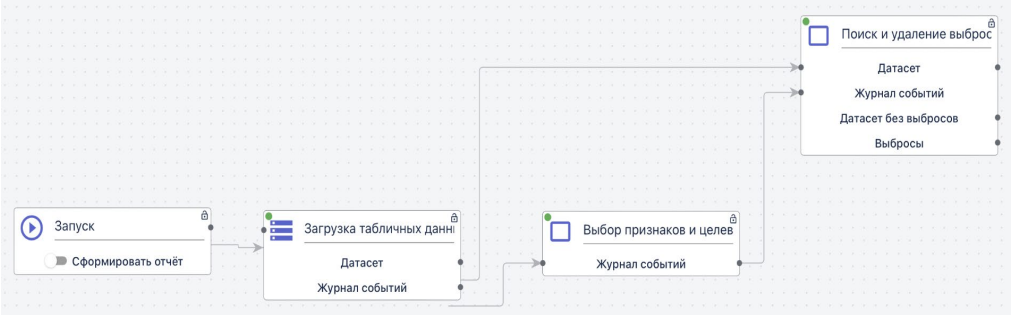
5. **Запустить процесс.**
6. **Отобразить визуализацию результатов:**



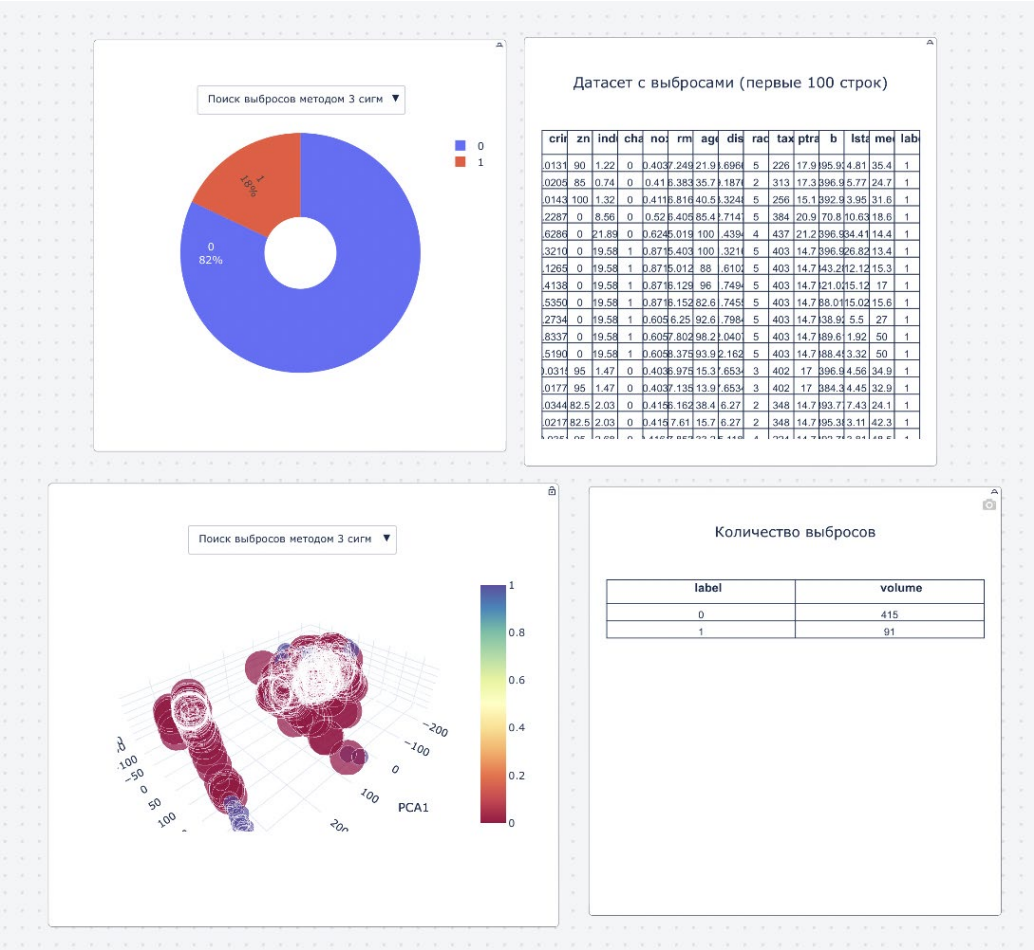
15.11 Поиск и удаление выбросов

1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**.
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка табличных данных**.
     – Выбрать файл для загрузки в формате CSV (например, BostonHousing.csv).
     – Сохранить изменения.
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
     – В поле **Признаки** выбрать нужные признаки для обучения.
     – В поле **Целевые признаки** выбрать нужные целевые признаки.
     – Сохранить изменения.
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Анализ данных -> Поиск и удаление выбросов**.
     – В поле **Метод** выбрать одно из следующих значений:
       1. **Метод 3 сигм**
       2. **Метод 5 сигм**
       3. **Изолирующий лес**
       4. **По квартилям**
     – При выборе метода **Изолирующий лес** указать нужное количество деревьев.
     – Сохранить изменения.
4. **Соединить элементы друг с другом как показано на рисунке ниже**:

- Для соединения точек элементов используется левая кнопка мыши.
- Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи, после чего она выделится голубым цветом и ее можно будет удалить.



5. **Запустить процесс.**
6. **Отобразить визуализацию результатов:**



15.12 Визуализация кластеров и определение ключевых слов в текстовых кластерах

1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».

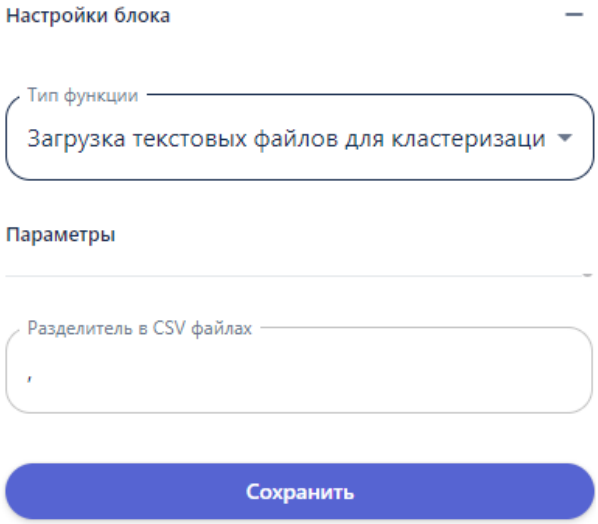
- На панели инструментов отобразится название созданной рабочей области.

2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:

     – В поле **Тип функции** выбрать **Загрузка данных -> Загрузка текстовых файлов для кластеризации**.

     – В поле Разделитель в CSV файлах выбрать, .

     – Сохранить изменения.



   - Настройка блока **Источник данных**:

     – В поле **Тип функции** выбрать **Предобработка данных -> Фильтрация текстового шума**.

     – Сохранить изменения.
   - Настройка блока **Источник данных**:

     – В поле **Тип функции** выбрать **Предобработка данных -> Лемматизация текста**.

     – Сохранить изменения.
   - Настройка блока **Источник данных**:

     – В поле **Тип функции** выбрать **Предобработка данных -> Векторизация текста**.

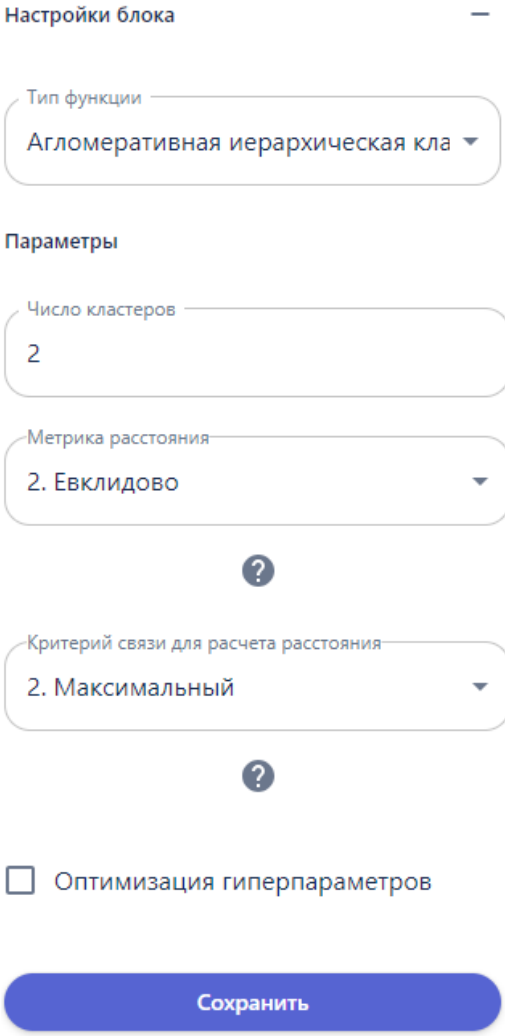
     В поле **Метод векторизации** выбрать нужный метод.

     В поле **Максимальная размерность текста** установить 10000.

     В поле **Количество признаков** введите 10.

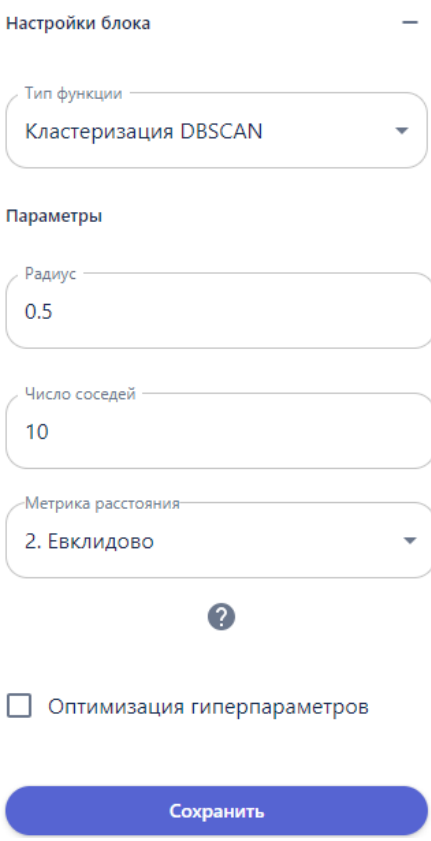
     – Сохранить изменения.
   - Настройка блока **Источник данных**:

– В поле **Тип функции** выбрать **Обучение без учителя -> Агломеративная иерархическая кластеризация**.
– В поле **Число кластеров** ввести 2.
– В поле **Метрика расстояния** выбрать 2. Евклидово.
– В поле **Критерий связи для расчета расстояния** выбрать 2. Максимальный.

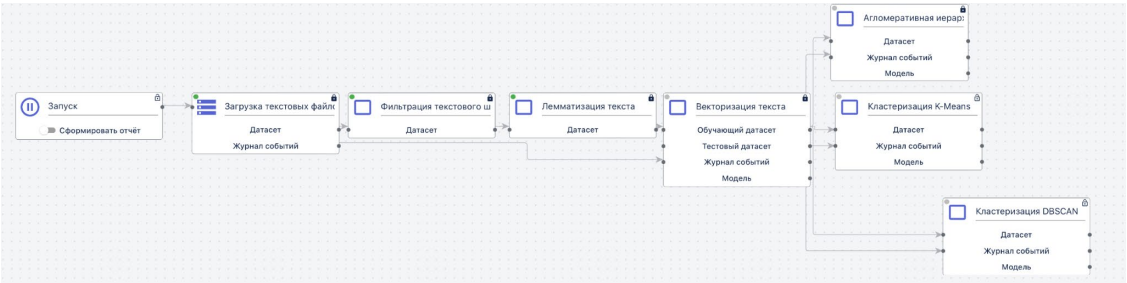


– Сохранить изменения.

- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Обучение без учителя -> Кластеризация K-means**.
  – В поле **Число кластеров** задать 2.
  – Сохранить изменения.
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Обучение без учителя -> Кластеризация DBSCAN**.
  – В поле **Радиус** ввести 0,5.
  – В поле **Число соседей** ввести 10.
  – В поле **Метрика расстояния** выбрать 2. Евклидово.
  – Сохранить изменения.



4. **Соединить элементы друг с другом как показано на рисунке ниже**:
   - Для соединения точек элементов используется левая кнопка мыши.
   - Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи. Когда связь выделится, удалить ее.



5. **Запустить процесс.**
6. **Отобразить нужную визуализацию результатов.**

15.13 Использование горячих клавиш

1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:

- Блок **Запуск**;
- Блок **Источник данных**;
- Блок **Процесс**;

3. **Использование горячих клавиш**. На Платформе в правом нижнем углу можно нажать на  , после чего выпадет подсказка с возможными горячими клавишами. Выбрать один из добавленных элементов и затем нажать на клавиатуре клавиши, показанные на рисунке ниже:

Рисунок … - Использование горячих клавиш

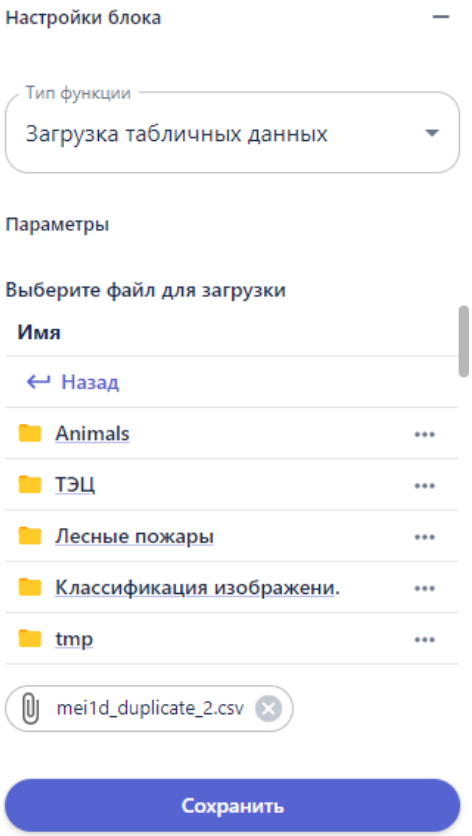
15.14 Использование приложения Docker для классификации текстов

1. В боковом меню выберите пункт **Приложения.**
2. Скачайте приложение Docker на компьютер.
3. Запустите приложение Docker.
4. Укажите в качестве источника текстовые данные.
5. Выполните классификацию и получите прогноз.
6. Сохраните результаты прогноза.

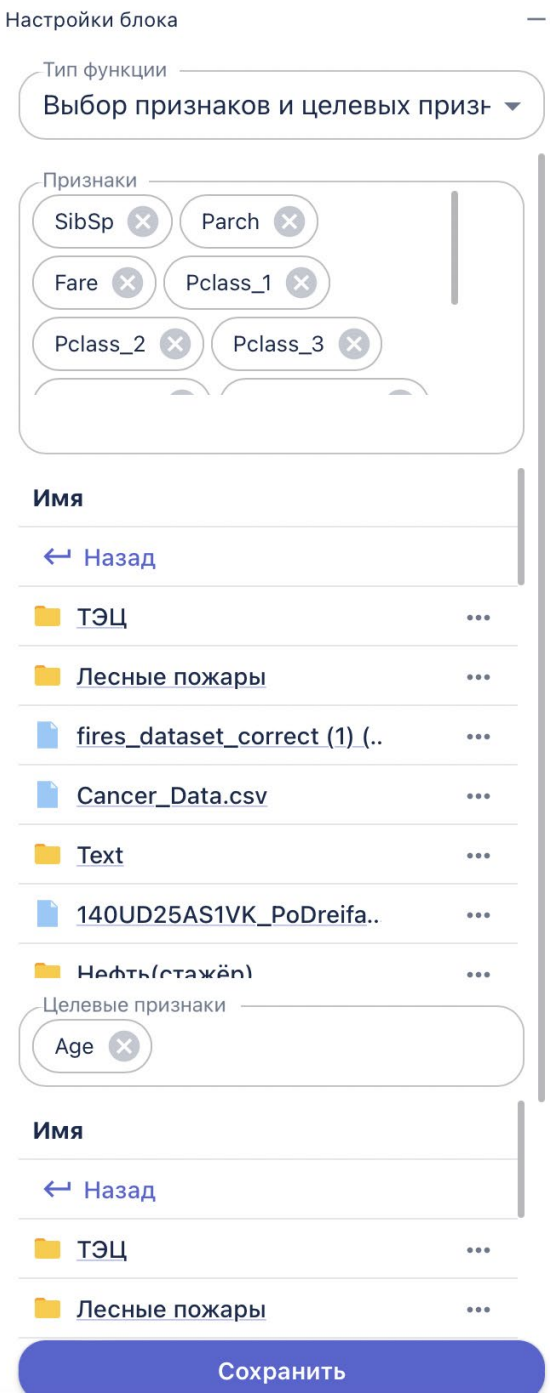
15.15 Формирование сравнительной таблицы обученных моделей

1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.

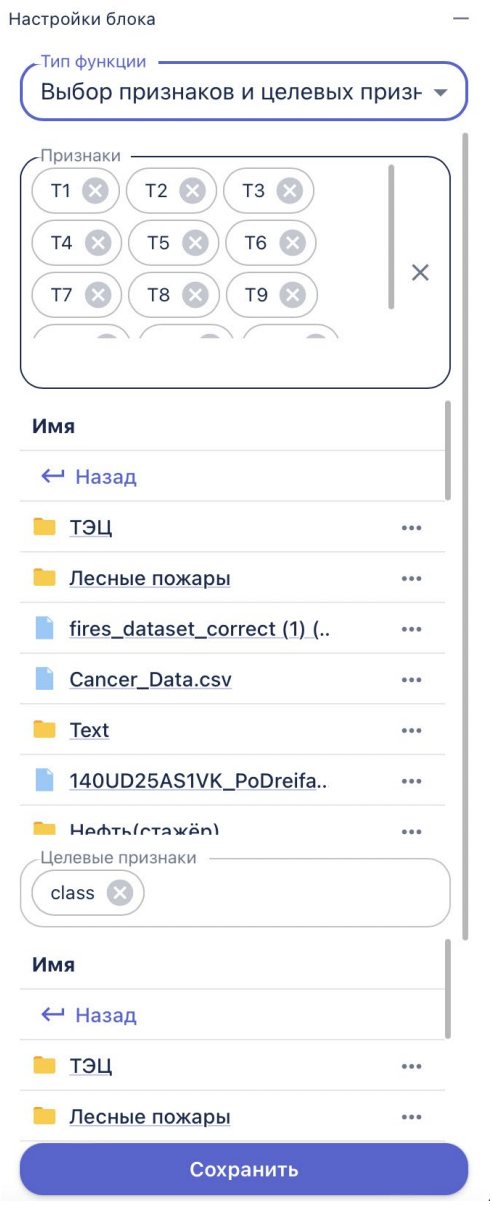
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка табличных данных**.
     – Выбрать файл для загрузки в формате CSV.
     – Сохранить изменения.



   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
     – В поле **Признаки** выбрать нужные признаки для обучения.
     – В поле **Целевые признаки** выбрать нужный целевой признак.
     – Сохранить изменения.



- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
  – В поле **Признаки** выбрать нужные признаки для обучения.
  – В поле **Целевые признаки** выбрать нужные целевые признаки.
  – Сохранить изменения.



- Настройка блока **Процесс**:

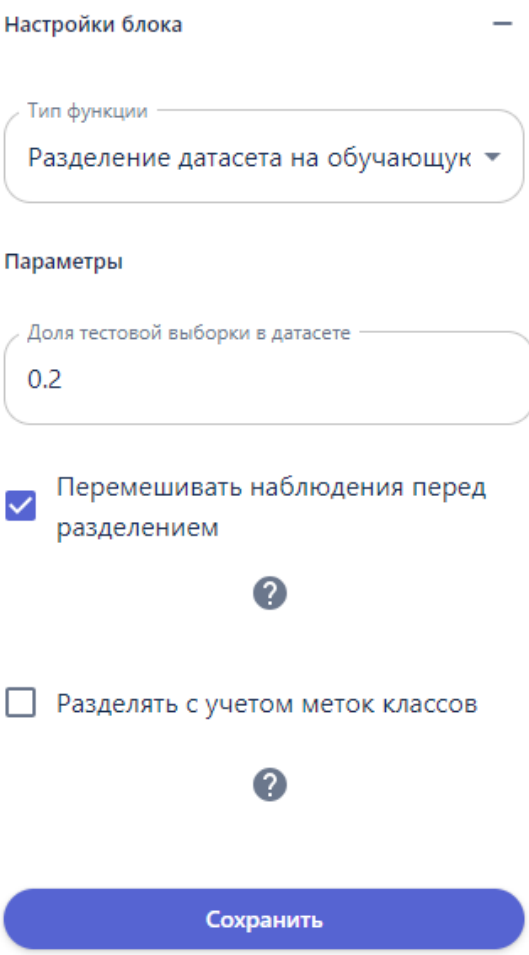
  – В поле **Тип функции** выбрать **Машинное обучение -> Разделение датасета на обучающую и тестовую выборки**.

  – В поле **Доля тестовой выборки в датасете** ввести нужное значение.

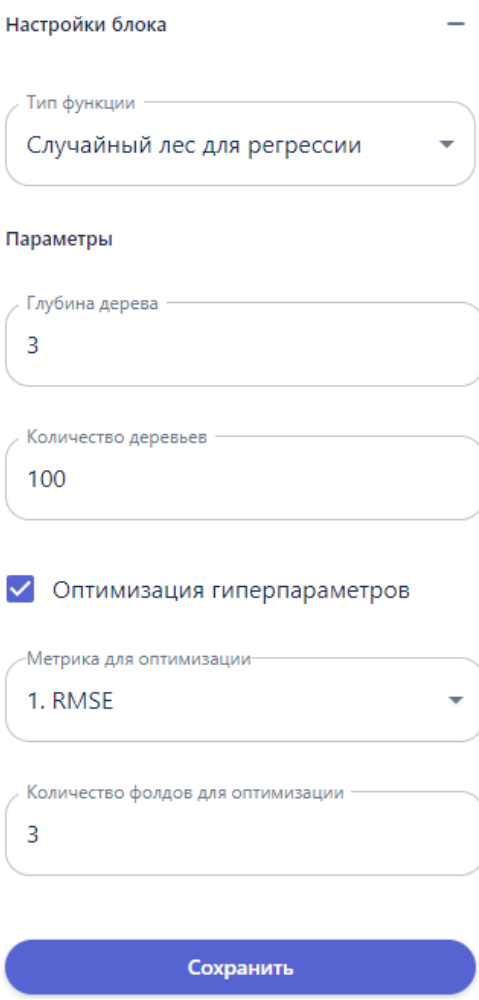
  – При необходимости установить флажок для параметра «**Перемешивать наблюдения перед разделением**».

  – При необходимости установить флажок для параметра «**Разделять с учетом меток классов**».

  – Сохранить изменения.

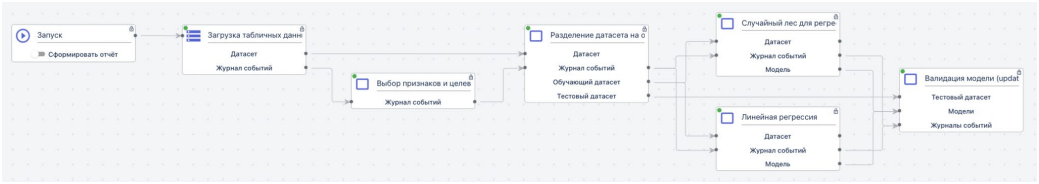


- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Регрессия -> Случайный лес для регрессии**.
  – В поле **Глубина дерева** ввести 3.
  – В поле **Количество деревьев** ввести 100.
  После выбора «Оптимизация гиперпараметров», появятся два поля:
  – В поле «**Метрика для оптимизации»** выбрать 1.RMSE.
  – В поле «**Количество фолдов для оптимизации»** ввести 3.
  – Сохранить изменения.

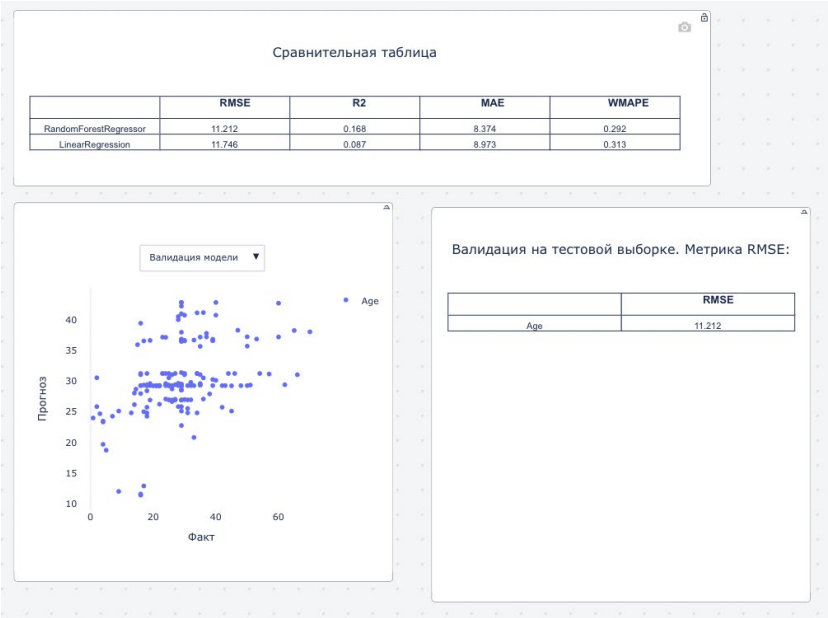


- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Регрессия -> Линейная регрессия**.
  – Сохранить изменения.
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Валидация модели (update)**.
  – В поле «**Метрика»** выбрать 1.RMSE.
  – Сохранить изменения.

4. **Соединить элементы друг с другом как показано на рисунке ниже:**
   - Для соединения точек элементов используется левая кнопка мыши.
   - Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи, после чего она выделится и ее можно будет удалить.



5. **Запустить процесс.**
6. **Отобразить визуализацию результатов:**



15.16 Классификация родинок

Описание методики проверки функционала, реализующего анализ и классификацию родинок на доброкачественные и злокачественные, приводится в следующем методическом пособии:.

15.17 Сегментация изображений

1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы 2-х блок-схем.
   1-я блок схема (сохранение модели):
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.

   2-я блок схема (загрузка модели):
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**.
3. **Настройка элементов (1-ой блок схемы):**
   - Настройка блока **Источник данных**:

– В поле **Тип функции** выбрать **Сегментация (обучение)**.

– В поле **Путь до датасета** выбрать файл для загрузки в формате CSV (**Cars_Segm**).

– В поле **Размер изображения** выбрать **2.224**.

– В поле **Модель сегментации** выбрать **4.deeplabv_resnet50**.

– В поле **Метрика качества** выбрать **1.loU.**

– В поле **Количество эпох** выбрать **2**.

– В поле **Loss функция** выбрать **2. focal.**

– В поле **batch_size** выбрать **16**.

– В поле **Оптимизатор** выбрать 1.AdamW.

– В поле **Шаг при обучении** выбрать 0.001.

– В поле **Вероятность вертикального переворота изображения** выбрать 0.05.

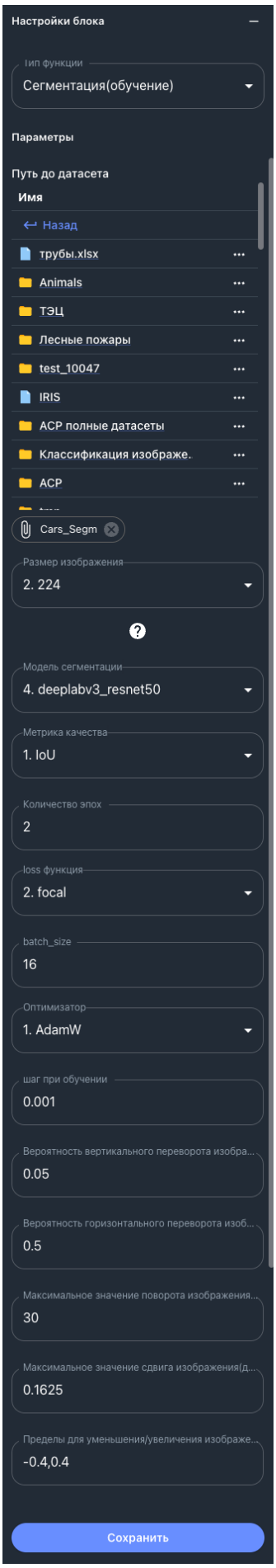
– В поле **Вероятность вертикального переворота изображения** выбрать 0.5.

– В поле **Максимальное значение поворота изображения** выбрать 30.

– В поле **Максимальное значение сдвига изображения** выбрать 0.1625.

– В поле **Пределы для уменьшения/увеличения изображения** выбрать -0.4,0.4.

– Сохранить изменения.

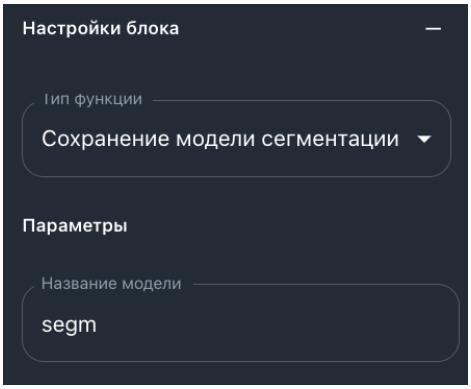


- Настройка блока **Процесс**:

– В поле **Тип функции** выбрать **Сохранение модели сегментации**.

– В поле **Название модели** ввести **segm**.

– Сохранить изменения.



- Настройка блока **Процесс**:

  – В поле **Тип функции** выбрать **Сегментация (прогноз)**.

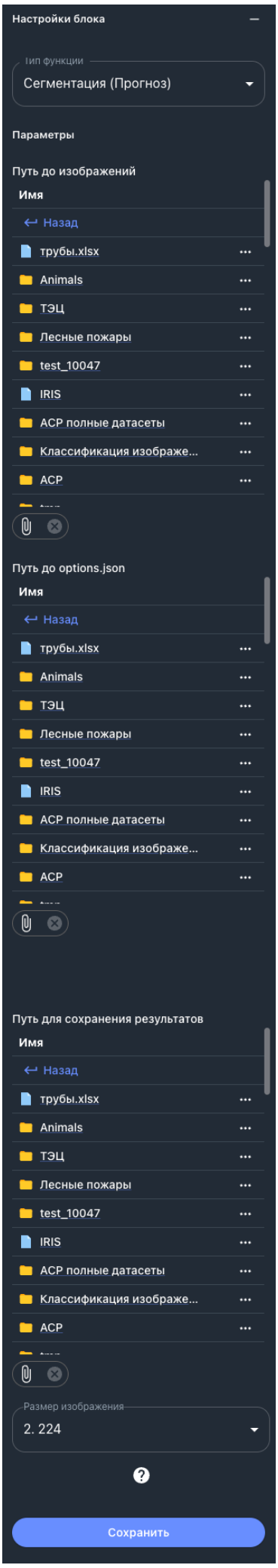
  – В поле **Путь до изображений** указать путь до исходных изображений.

  – В поле **Путь до options.json** указать путь до файла с настройками в формате JSON.

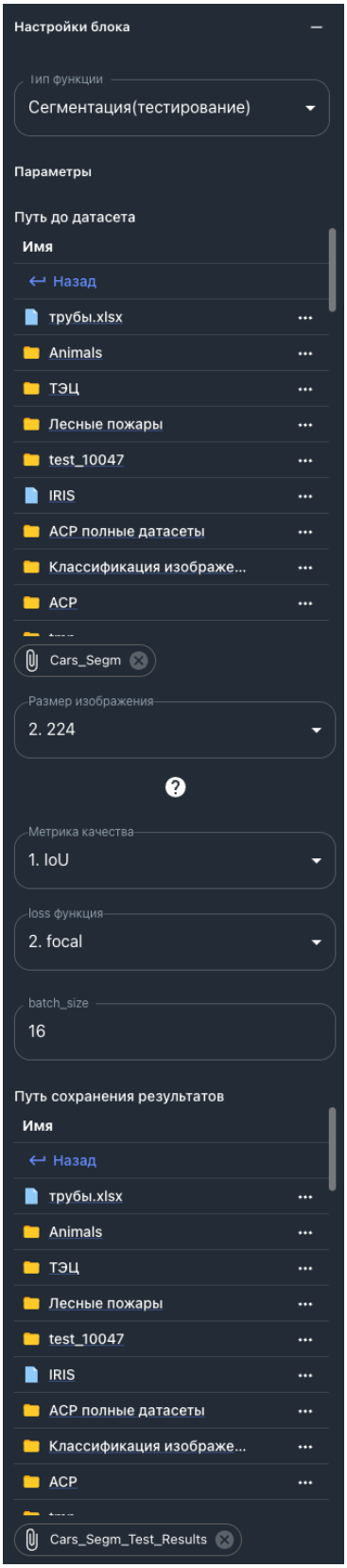
  – В поле **Путь до сохранения результатов** указать путь к папке, в которую будут сохранены результаты работы.

  – В поле **Размер изображени**я выбрать **2.224**.

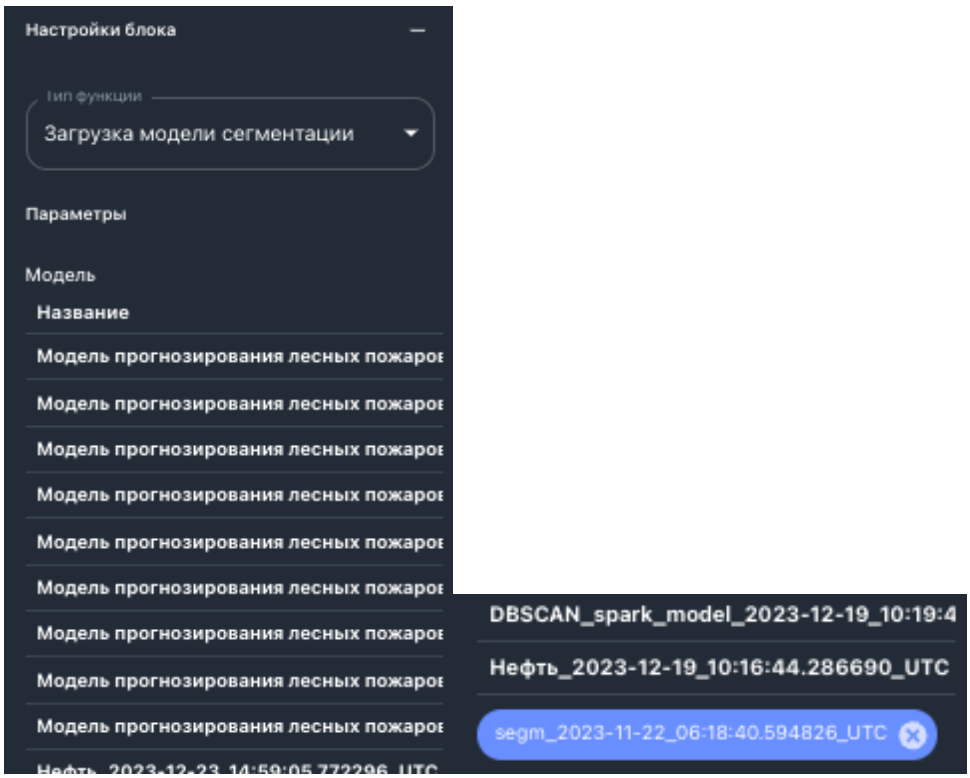
  – Сохранить изменения.


Настройки блока
Тип функции
Сегментация (Прогноз)
Параметры
Путь до изображений
Имя
Назад
трубы.xlsx
Animals
ТЭЦ
Лесные пожары
test_10047
IRIS
ACP полные датасеты
Классификация изображе...
ACP
Путь до options.json
Имя
Назад
трубы.xlsx
Animals
ТЭЦ
Лесные пожары
test_10047
IRIS
ACP полные датасеты
Классификация изображе...
ACP
Путь для сохранения результатов
Имя
Назад
трубы.xlsx
Animals
ТЭЦ
Лесные пожары
test_10047
IRIS
ACP полные датасеты
Классификация изображе...
ACP
Размер изображения
2. 224
Сохранить

- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Сегментация (тестирование)**.
  – В поле **Путь до датасета** выбрать файл для загрузки в формате CSV (**Cars_Segm**).
  – В поле **Размер изображени**я выбрать **2.224**.
  – В поле **Метрика качества** выбрать **1.loU.**
  – В поле **Loss функция** выбрать **2. focal.**
  – В поле **batch_size** выбрать **16**.
  – В поле **Путь до сохранения результатов** указать путь к папке, в которую будут сохранены результаты работы.
  – Сохранить изменения.



4. **Настройка элементов (2-ой блок схемы):**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка модели сегментации**.
     – В поле **Параметры** выбрать модель **segm_2023-11-22_06:18:40.594826_UTC**.
     – Сохранить изменения.



- Настройка блока **Процесс**:

  – В поле **Тип функции** выбрать **Сегментация (прогноз)**.

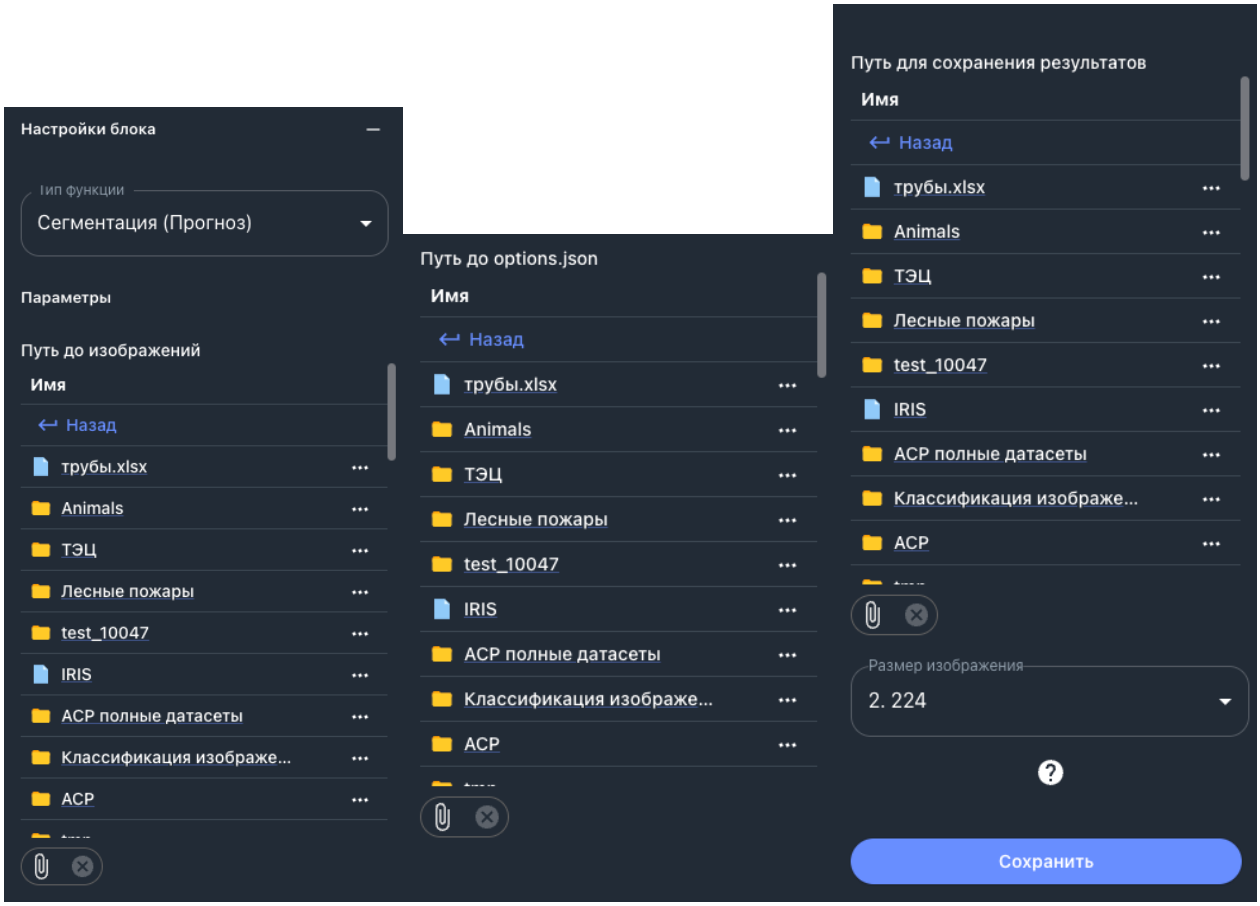
  – В поле **Путь до изображений** указать путь до исходных изображений.

  – В поле **Путь до options.json** указать путь до файла с настройками в формате JSON.

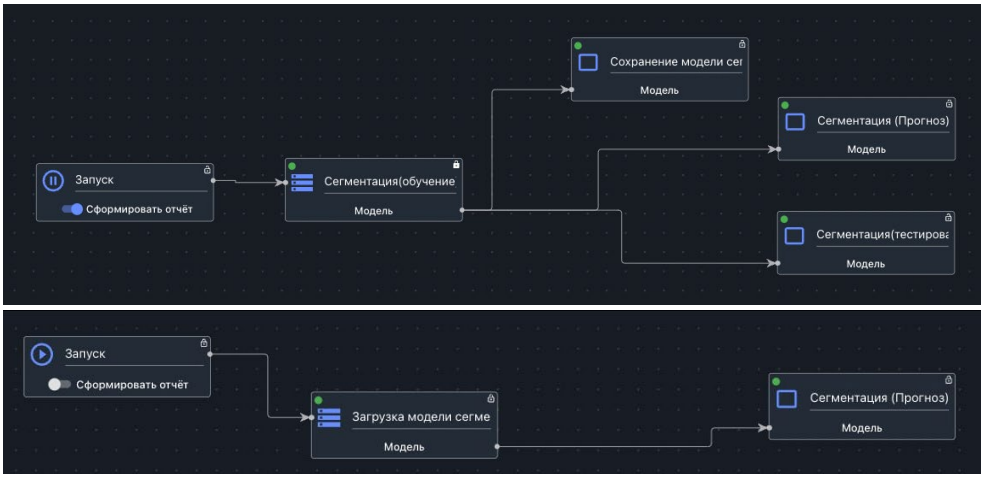
  – В поле **Путь до сохранения результатов** указать путь к папке. в которую будут сохранены результаты работы.

  – В поле **Размер изображени**я выбрать **2.224**.

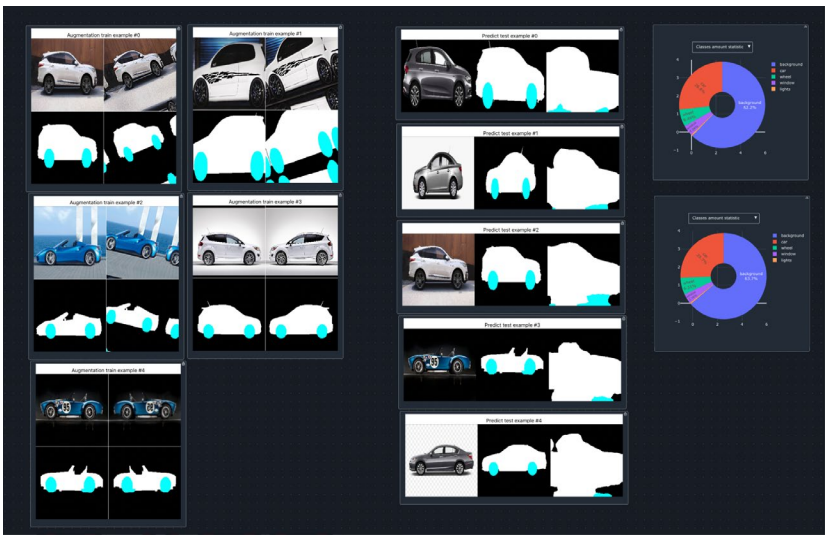
  – Сохранить изменения.



5. **Соединить элементы друг с другом как показано на рисунке ниже**:
   - Для соединения точек элементов используется левая кнопка мыши.
   - Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи, после чего она выделится и ее можно будет удалить.

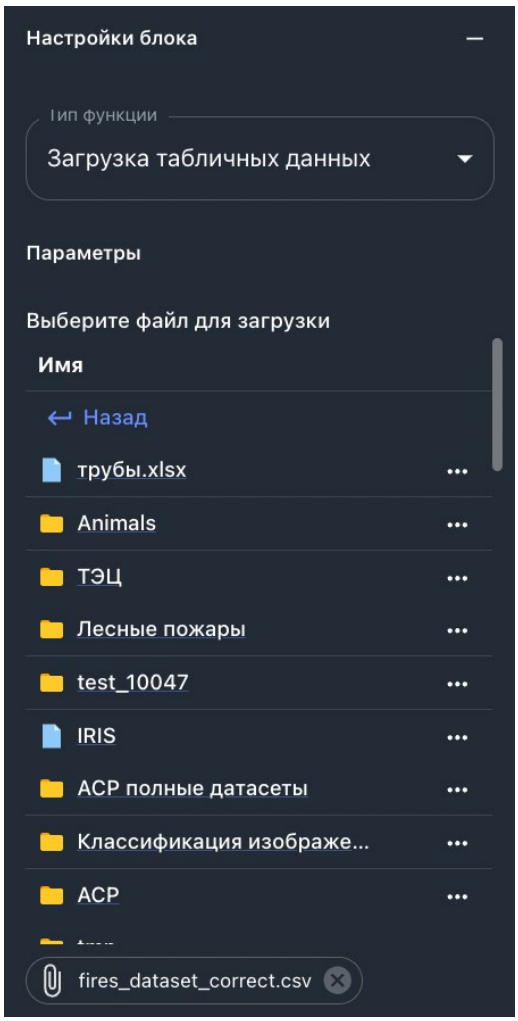


6. **Запустить процесс.**
7. **Отобразить визуализацию результатов:**

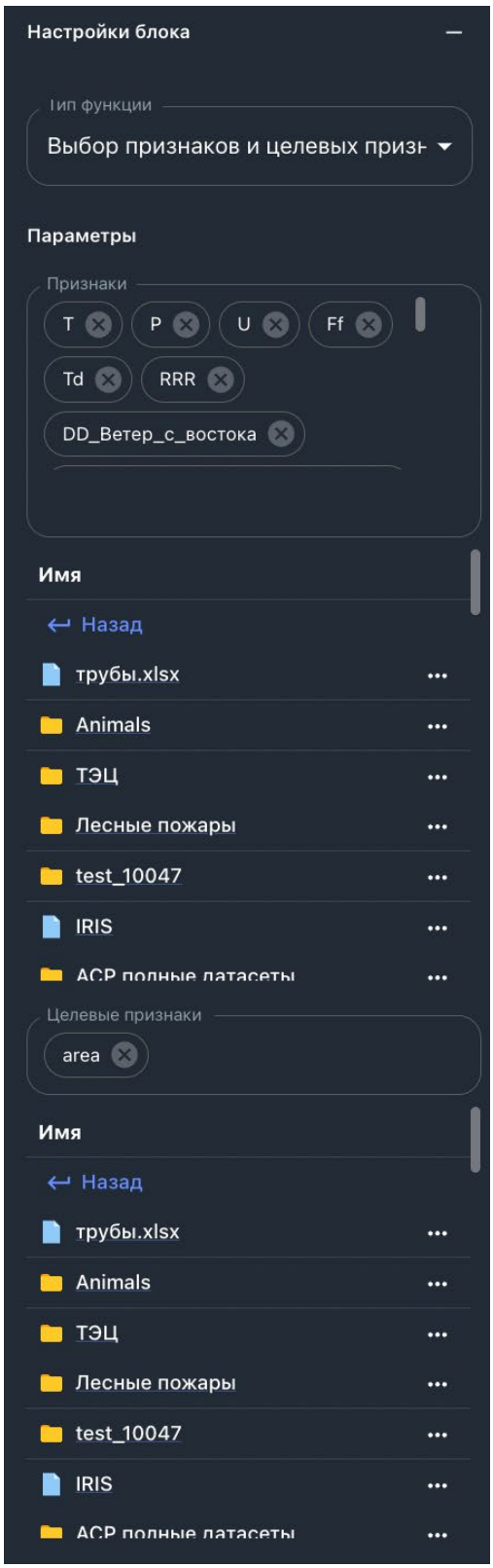


15.18 Стэкинг (классификация)

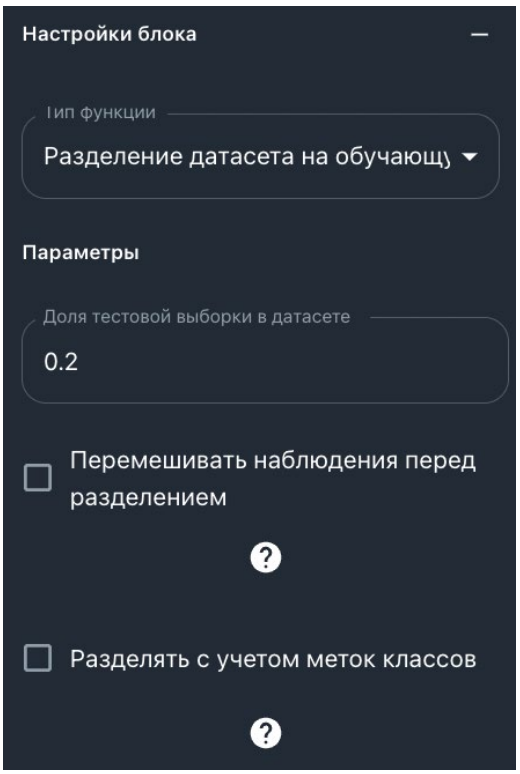
1. **Создание блок-схемы**.
   - Перейти в пункт меню **Моделирование** -> **Рабочая область**.
   - На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
   - В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
   - На панели инструментов отобразится название созданной рабочей области.
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка табличных данных**.
     – Выбрать для загрузки файл **fires_dataset_correct.csv**.
     – Сохранить изменения.



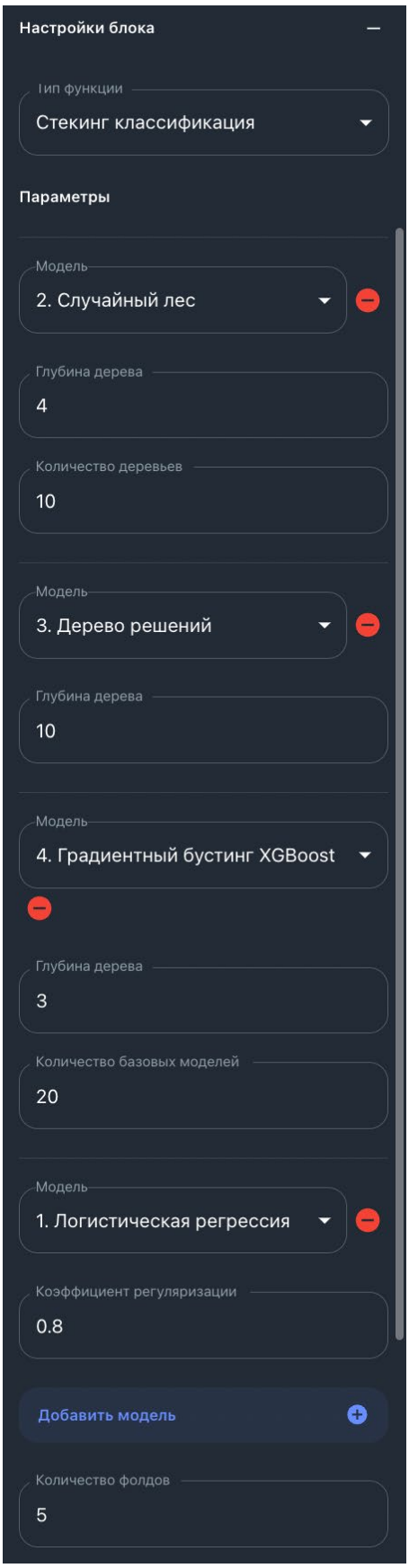
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
  – В поле **Признаки** выбрать нужные признаки для обучения.
  – В поле **Целевые признаки** выбрать нужные целевые признаки.
  – Сохранить изменения.



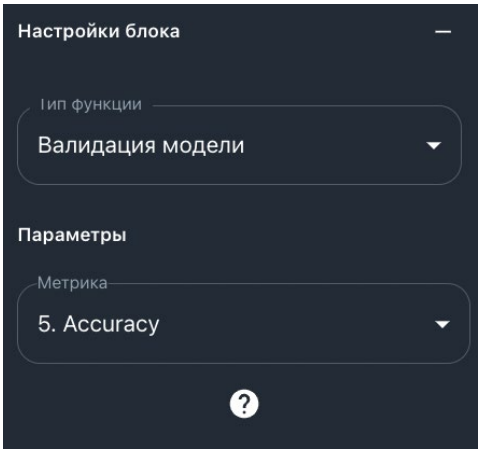
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Разделение датасета на обучающую и тестовую выборки**.
  – В поле **Доля тестовой выборки в датасете** ввести 0.2.
  – Сохранить изменения.



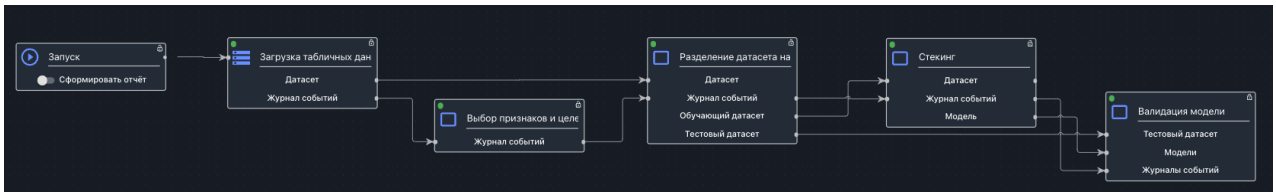
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Классификация -> Стекинг классификация**.
  – В поле **Модель** выбрать **2. Случайный лес**.
  – В поле **Глубина дерева** выбрать **4.**
  – В поле **Количество деревьев** выбрать **10.**
  – В поле **Модель** выбрать **3. Дерево решений**.
  – В поле **Глубина дерева** выбрать **10.**
  – В поле **Модель** выбрать **4. Градиентный бустинг XGBoost**.
  – В поле **Глубина дерева** выбрать **3.**
  – В поле **Количество базовых моделей** выбрать **20.**
  – В поле **Модель** выбрать **1. Логистическая регрессия**.
  – В поле **Коэффициент регуляризации** выбрать **0.8**.
  – В поле **Количество фолдов** выбрать **5**.
  – В поле **Порог классификации** выбрать 0.**5**.
  – Сохранить изменения.



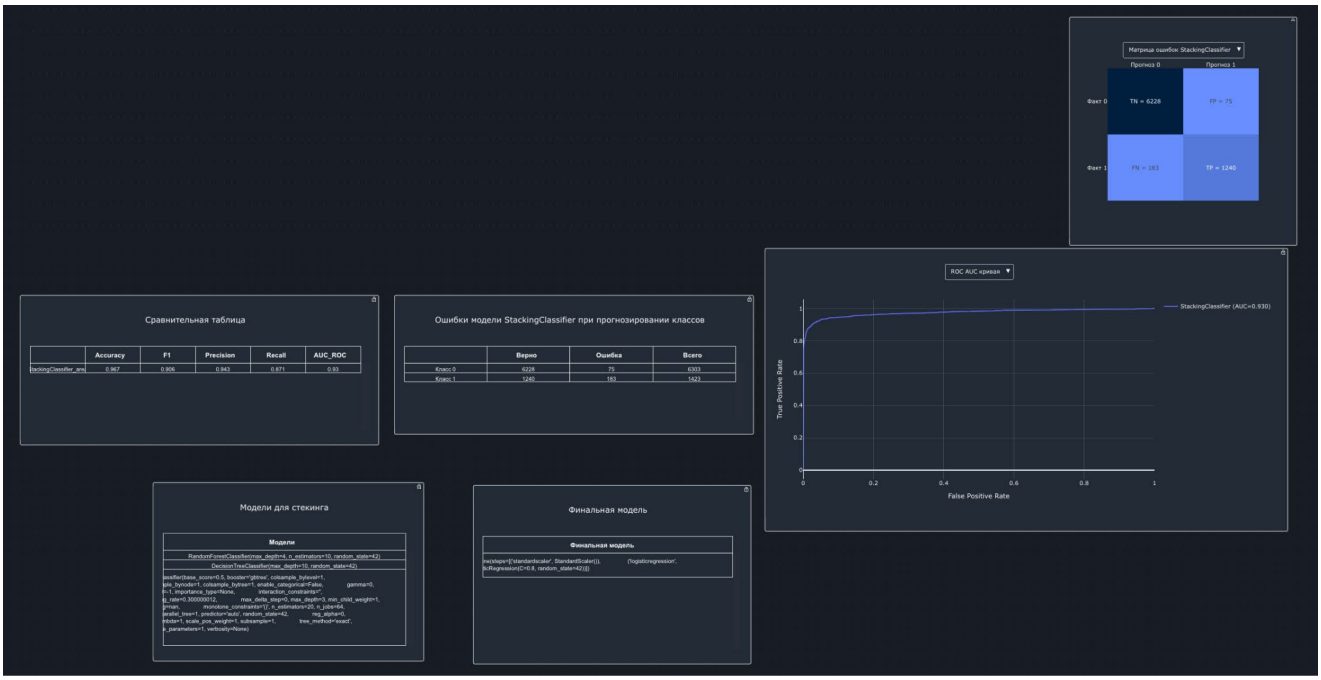
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение-> Валидация модели**.
  – В поле **Метрика** выбрать **5. Accuracy**.
  – Сохранить изменения.



4. **Соединить элементы друг с другом как показано на рисунке ниже**:
    - Для соединения точек элементов используется левая кнопка мыши.
    - Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи, после чего она выделится и ее можно будет удалить.



5. **Запустить процесс.**
6. **Отобразить визуализацию результатов:**

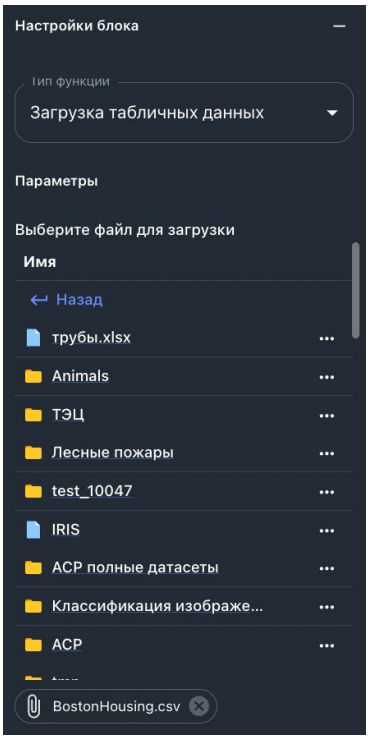


15.19 Стекинг (Регрессия)

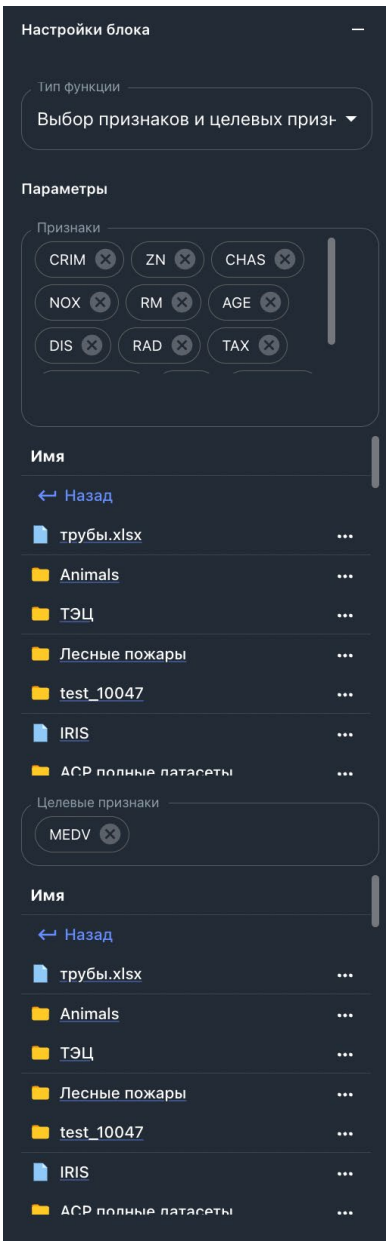
1. **Создание блок-схемы**.
    - Перейти в пункт меню **Моделирование** -> **Рабочая область**.

- На панели инструментов блок-схемы нажать кнопку «Создание рабочей области» (кнопка ⊕).
- В открывшемся окне ввести имя рабочей области и нажать кнопку «Создать».
- На панели инструментов отобразится название созданной рабочей области.

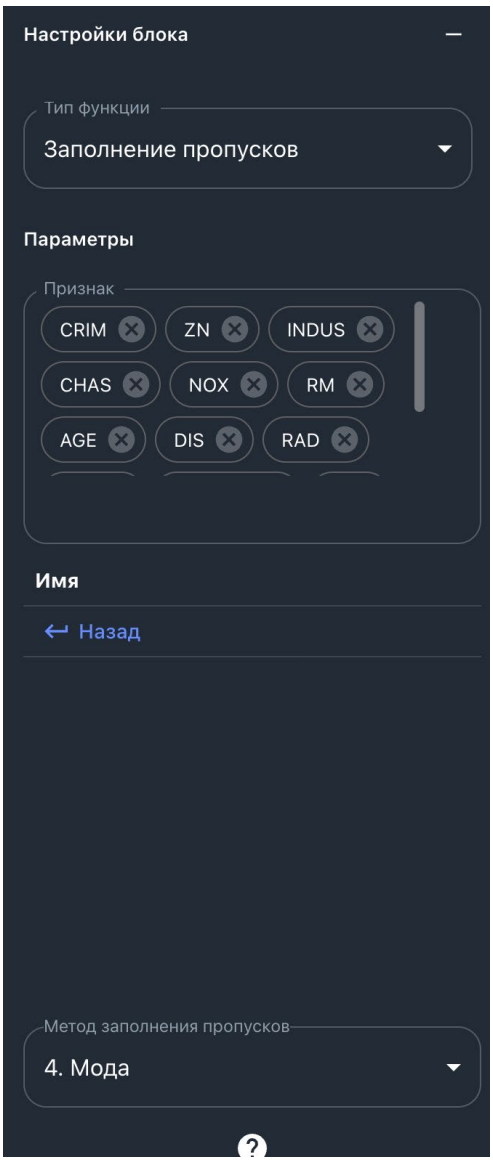
2. **Добавление элементов**. На панели инструментов блок-схемы с помощью кнопки «BPMN» добавить указанные ниже элементы блок-схемы:
   - Блок **Запуск**;
   - Блок **Источник данных**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**;
   - Блок **Процесс**.
3. **Настройка элементов:**
   - Настройка блока **Источник данных**:
     – В поле **Тип функции** выбрать **Загрузка табличных данных**.
     – Выбрать для загрузки файл **BostonHousing.csv**.
     – Сохранить изменения.



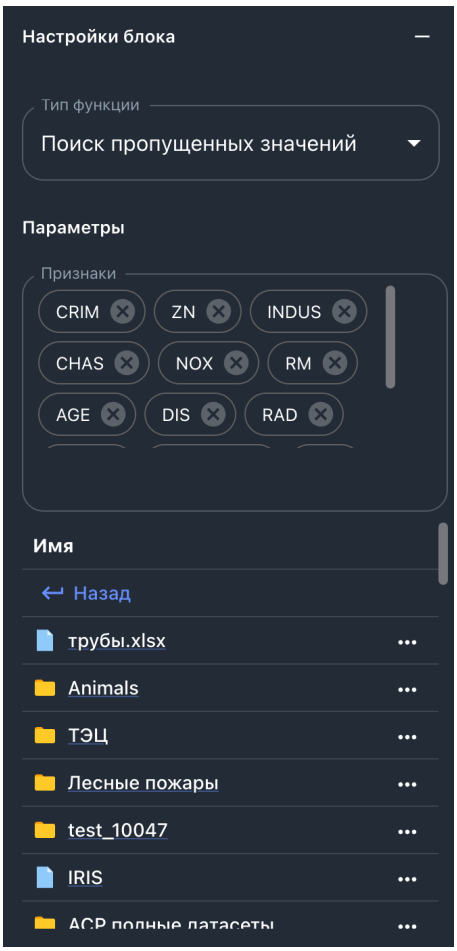
   - Настройка блока **Процесс**:
     – В поле **Тип функции** выбрать **Анализ данных -> Выбор признаков и целевых признаков**.
     – В поле **Признаки** выбрать нужные признаки для обучения.
     – В поле **Целевые признаки** выбрать нужные целевые признаки.
     – Сохранить изменения.



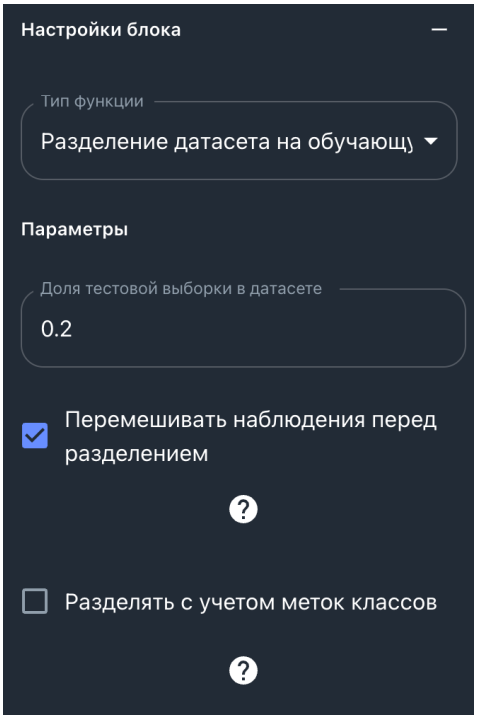
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Предобработка данных -> Заполнение пропусков**.
  – В поле **Параметры** выбрать нужные параметры обработки.
  – В поле **Метод заполнения пропусков** выбрать **4. Мода**.
  – Сохранить изменения.



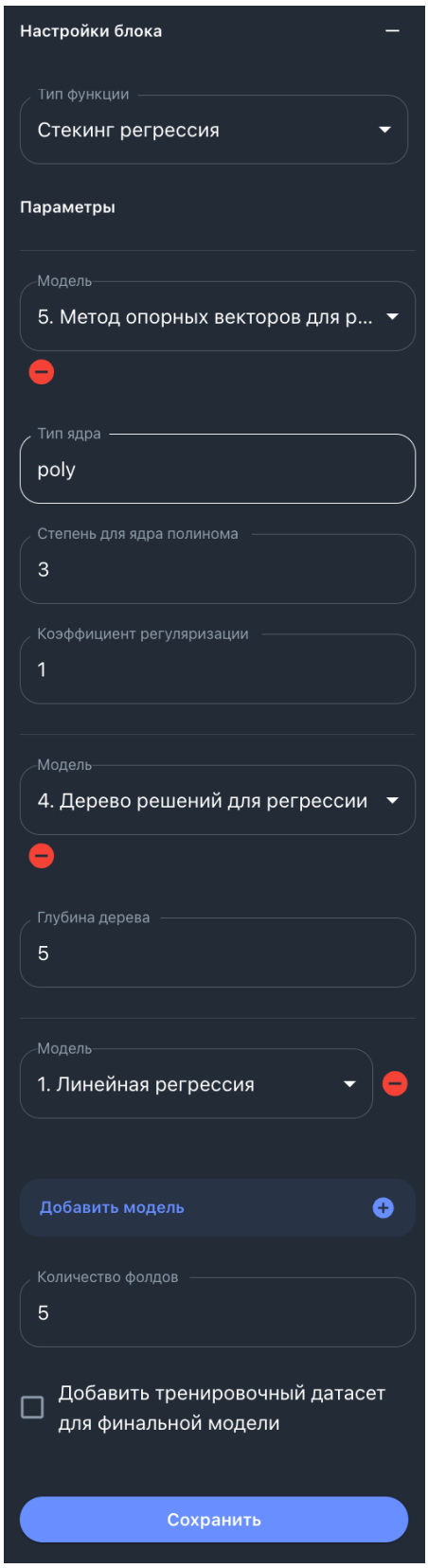
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Анализ данных -> Поиск пропущенных значений**.
  – В поле **Параметры** выбрать нужные параметры обработки.
  – Сохранить изменения.



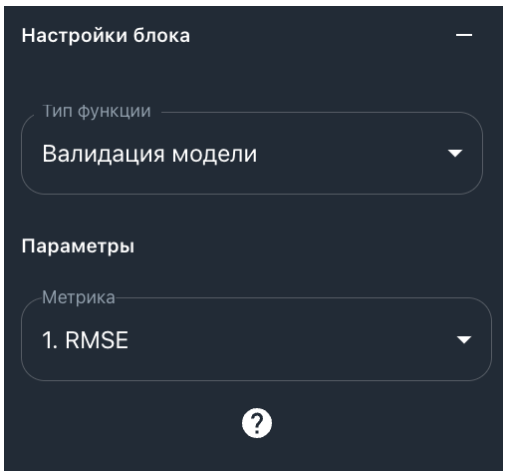
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение -> Разделение датасета на обучающую и тестовую выборки**.
  – В поле **Доля тестовой выборки в датасете** ввести **0.2**.
  – Установить флажок **Перемешивать наблюдения перед разделением**.
  – Сохранить изменения.



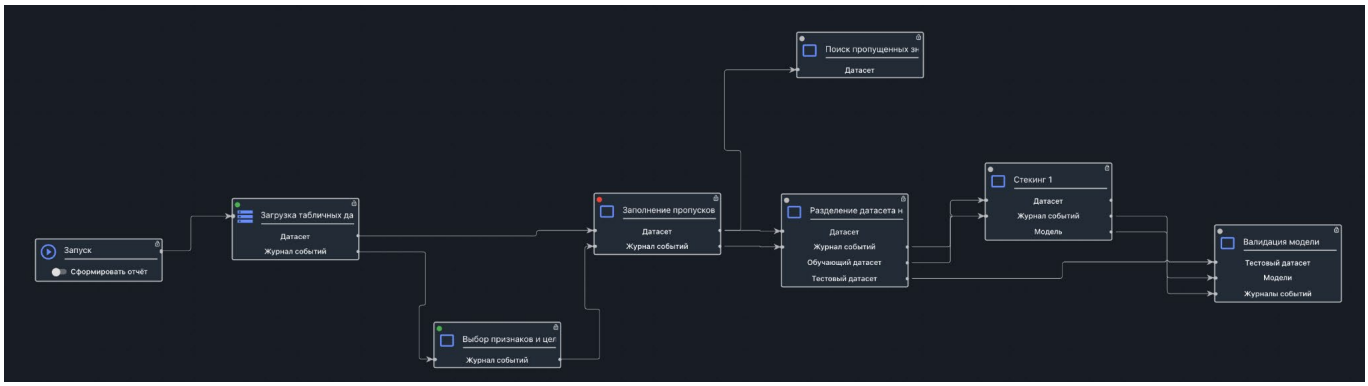
- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Регрессия -> Стекинг регрессия**.
  – В поле **Модель** выбрать **5. Метод опорных векторов для регрессии**.
  – В поле **Тип ядра** выбрать **poly.**
  – В поле **Степень для ядра полинома** выбрать **3.**
  – В поле **Коэффициент регуляризации** выбрать **1.**
  – В поле **Модель** выбрать **3. Дерево решений для регрессии**.
  – В поле **Глубина дерева** выбрать **5.**
  – В поле **Модель** выбрать **1. Линейная регрессия**.
  – В поле **Количество фолдов** выбрать **5**.
  – Сохранить изменения.



- Настройка блока **Процесс**:
  – В поле **Тип функции** выбрать **Машинное обучение-> Валидация модели**.
  – В поле **Метрика** выбрать **1. RMSE**.
  – Сохранить изменения.



4. **Соединить элементы друг с другом как показано на рисунке ниже**:
   - Для соединения точек элементов используется левая кнопка мыши.
   - Для удаления неверно добавленной связи нужно дважды кликнуть по линии связи, после чего она выделится и ее можно будет удалить.



5. **Запустить процесс.**
6. **Отобразить визуализацию результатов:**

16. Администрирование Платформы

16.1 Пользователи и группы

Платформа позволяет разделять уровни доступа к разделам меню для разных пользователей в зависимости от требования проекта. Для этого создаются Группы, которые впоследствии назначаются отдельным пользователям.

Создание новой Группы или редактирование уже созданной группы осуществляется в разделе Администрирование -> Группы:

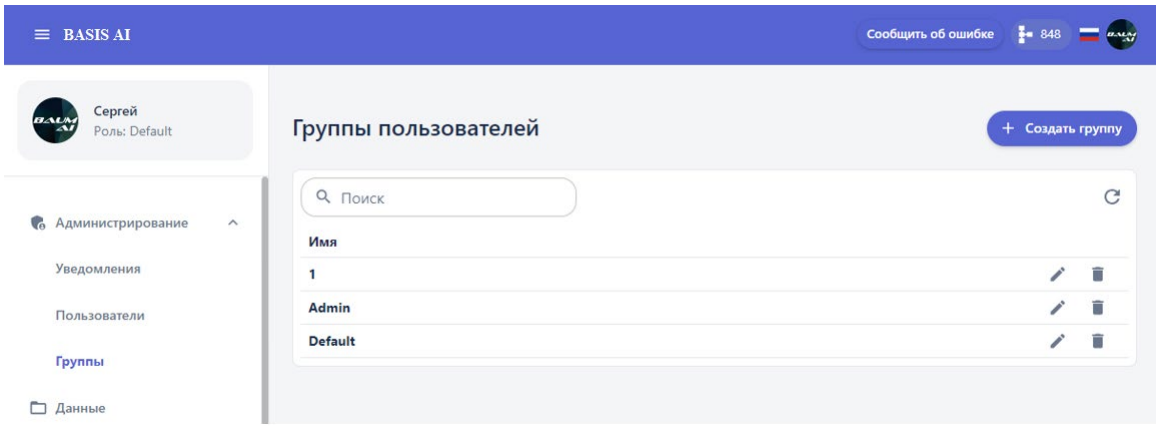


Рисунок 16.1 – Раздел Группы меню Администрирование

В списке отображаются уже созданные группы. Для создания новой группы, нажимается кнопка «Создать группу» в правом верхнем угле, в открывшемся окне отобразятся все доступные для настройки параметров:

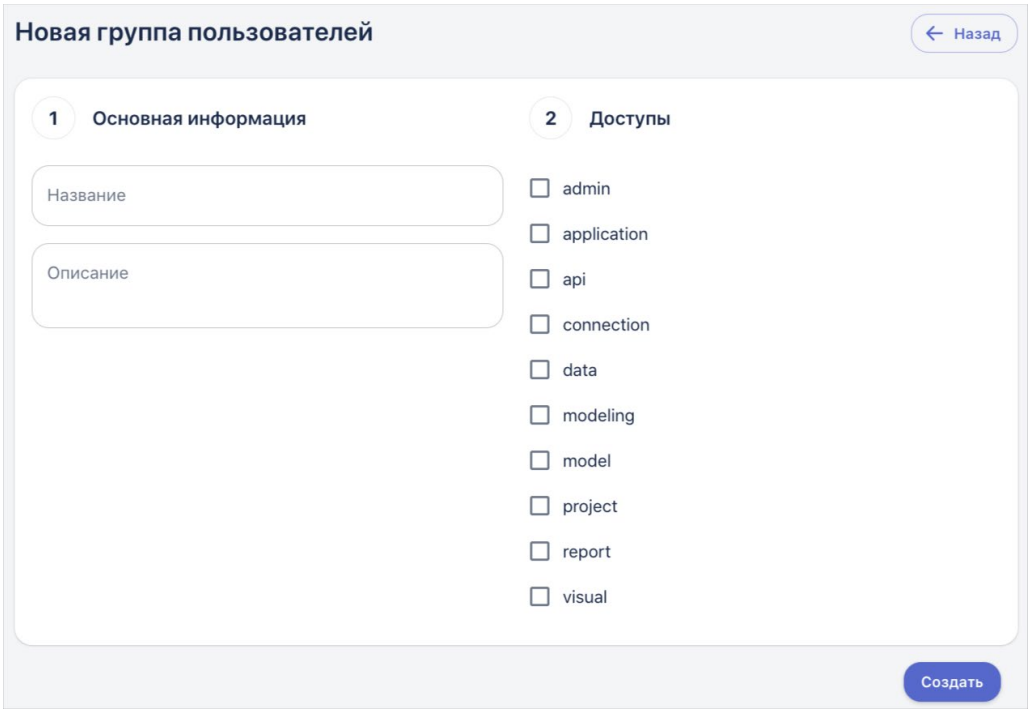


Рисунок 16.2 – Настройка новой группы пользователей

Сначала заполняется основная информация:

- *Название* - задать уникальное название для Группы (обязательное поле)

- *Описание* - внести краткую информацию о применимости данной роли (необязательное поле)

Далее из списка доступов выбираются разделы данных, к которым у пользователей, принадлежащих к этой Группе, должны быть доступны:

- admin - Администрирования
- application - Приложения
- api - API
- connection - Соединения
- data - Данные
- modeling - Моделирование
- model - Модели
- project - Проекты
- report - Отчеты
- visual - Визуализация

После того, как галочки для соответствующих разделов проставлены, нажимается кнопка «Создать». Новая группа появится в списке.

Для того чтобы отредактировать одну из Групп, используется кнопка «Редактировать» в строке с ее названием. Для того чтобы удалить группу, нажимается кнопка «Удалить» :

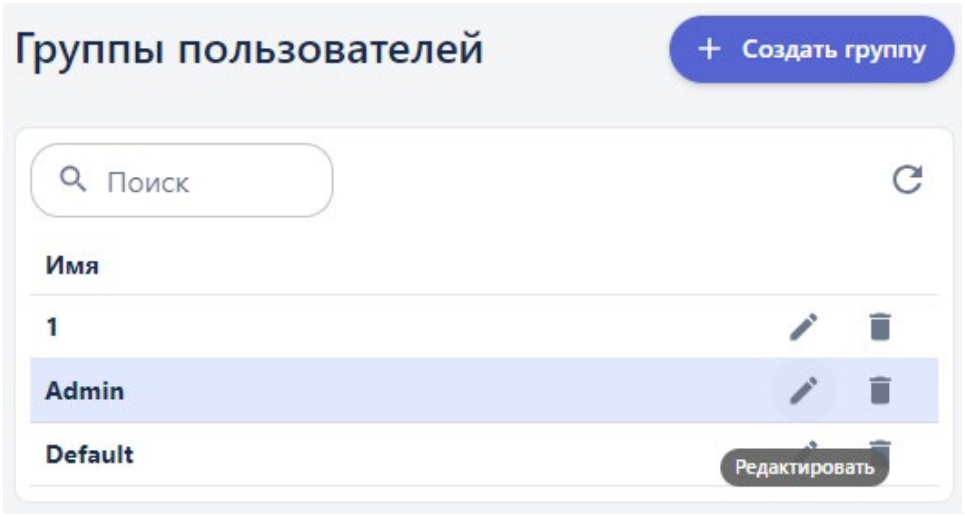


Рисунок 16.3 – Кнопки удаления и редактирования группы пользователей

Список всех пользователей представлен в разделе Администрирование -> Пользователи:

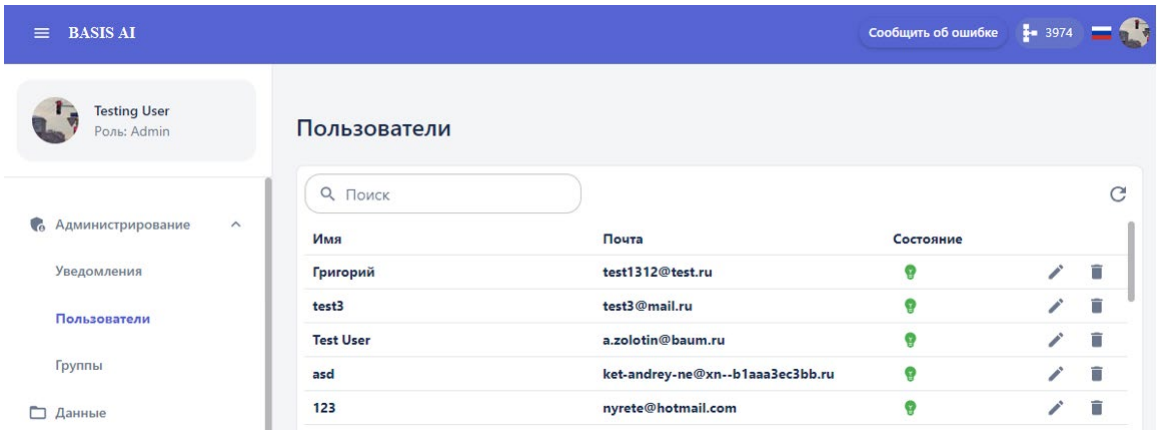


Рисунок 16.4 – Список пользователей в меню Администрирование

Для того чтобы отредактировать настройки пользователя, нажимается кнопка «Редактировать» в строке с именем:

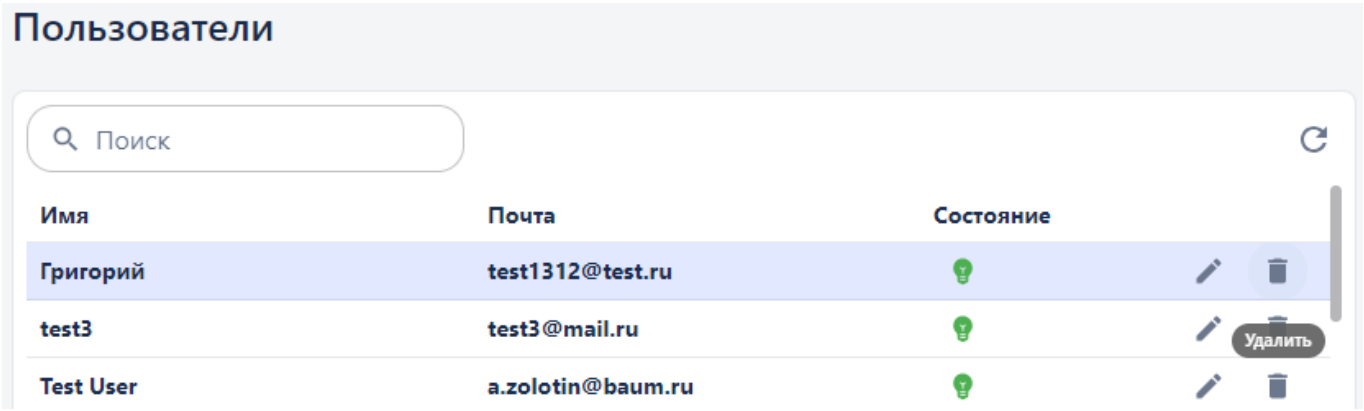


Рисунок 16.5 – Редактирование данных пользователя

В открывшемся окне есть возможность изменить имя пользователя, адрес электронной почты, а также назначить Группу:

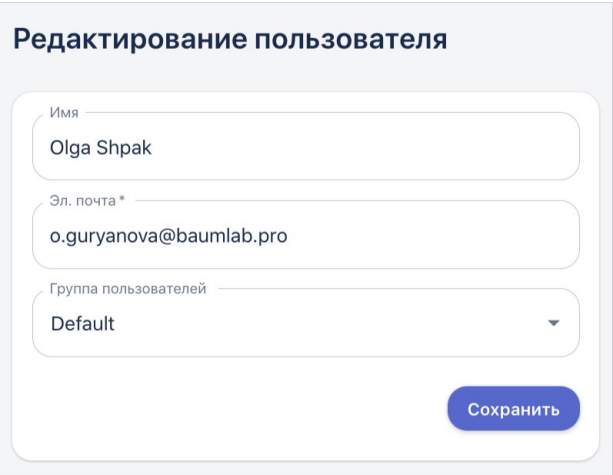


Рисунок 16.6 – Параметры для редактирования

Обратите внимание, что всем новым пользователям по умолчанию присваивается Группа Default (группа по умолчанию).

Для того чтобы удалить пользователя, используется кнопка «Удалить» в строке с его именем.

Рекомендуется оставлять одного-двух пользователей Администраторов, которые смогут управлять доступами и отвечать за настройки.

16.2 Настройка отправки уведомлений

На платформе реализована возможность отправки уведомлений - сообщений в телеграм или на почту. Этот функционал позволяет пользователям получать автоматические оповещения о результатах работы пайплайнов, при выполнении заданных условий. Создание и настройка каналов осуществляется следующим образом:

1. Перейти в раздел Администрирование -> Уведомления:

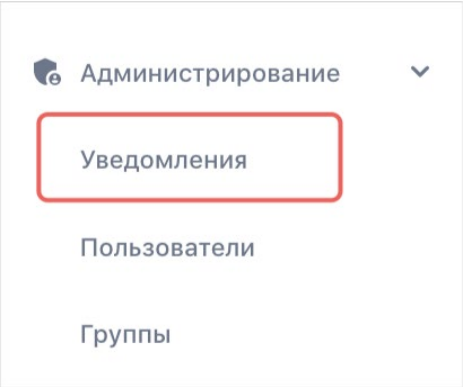


Рисунок 16.6 - Раздел Уведомления

2. В открывшемся окне отобразится список всех существующих каналов уведомлений:

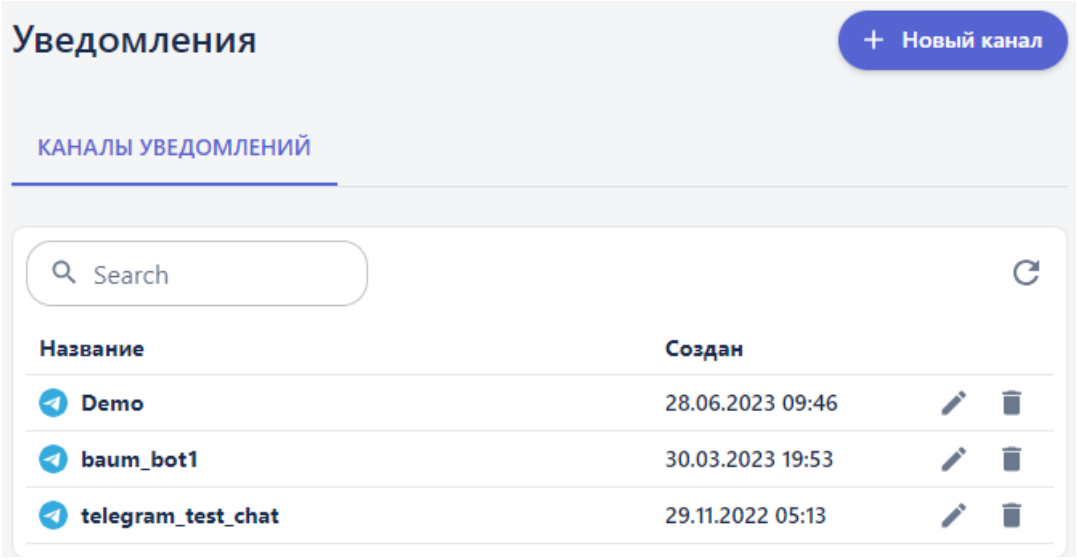


Рисунок 16.6 - Список созданных каналов

3. Для создания нового канала уведомлений нажимается кнопка «Новый канал» в правом верхнем углу
4. Открывается окно создания нового канала уведомлений:

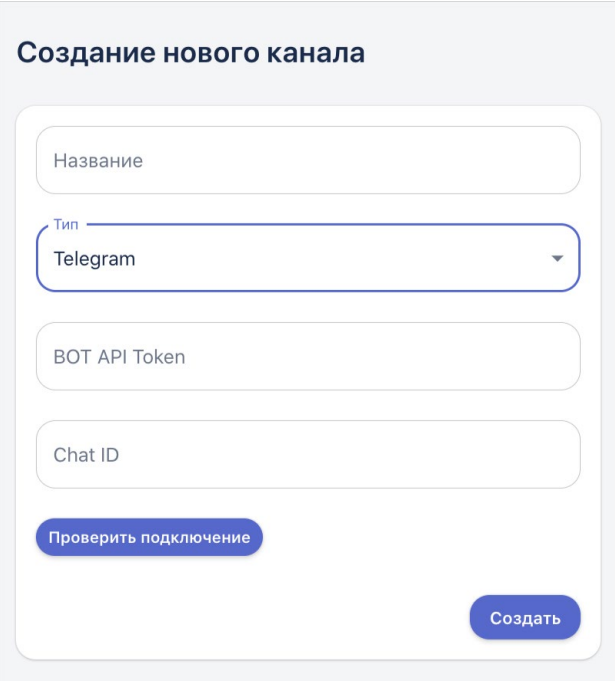


Рисунок 16.7 - Окно создания нового канала

5. Уведомления можно отправлять в телеграм бот или на почту.
   5.1. Для создания уведомления с использованием **телеграм бота** поля заполняются следующим образом:

- *Название* - пользователь задаёт название бота из телеграмм
- *Тип* - Telegram
- *BOT API Token* - токен бота, полученный в телеграм при создании бота
- *Chat ID* - уникальный численный идентификатор чат бота

> ***Примечание**: сначала пользователь должен создать чат бота в телеграм или получить его токен и id для настройки канала уведомлений. Создание бота описано в этой статье: https://tlgrm.ru/docs/bots#botfather*

   5.2. Для создания уведомления **на почту** поля заполняются следующим образом:

- *Название* - пользователь задает название канала
- *Тип* - Почта
- *Почты* - указываются адреса, на которые должны быть отправлены уведомления. При этом после ввода первого адреса, нужно нажать Enter, потом перейти к указанию следующего адреса и т д.

> ***Примечание**: уведомления на почту не реализованы в текущей версии системы*

6. После того, как все поля заполнены, нажимается кнопка «Проверить подключение», если все настройки были указаны верно, система выдаст сообщение от успешности подключения в верхнем правом углу окна. В противном случае, отобразится сообщение «Не удалось подключиться»
7. Завершающим этапом нажимается кнопка «Создать»
8. Новый канал отобразится в списке.

Для того чтобы начать получать уведомления в телеграм, пользователь должен создать пайплайн, который будет содержать блок «Процесс» с функцией «Отправка уведомлений», где указывается необходимый канал:

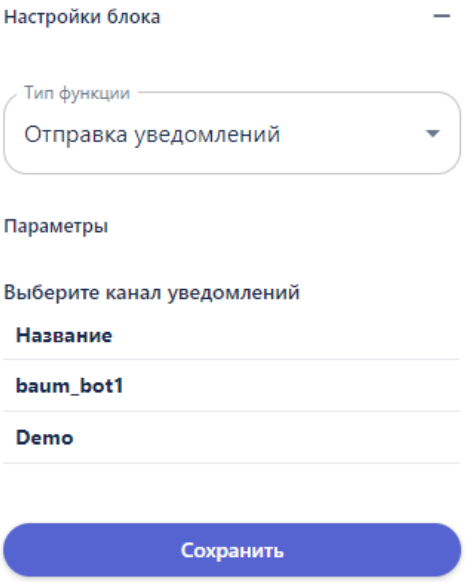


Рисунок 16.8 - Настройка блока «Отправка уведомлений»

Условием для отправки будет служить блок, идущий перед блоком уведомлений. Например, это может быть шлюз, где задаётся параметр, при котором уведомление должно быть отправлено:

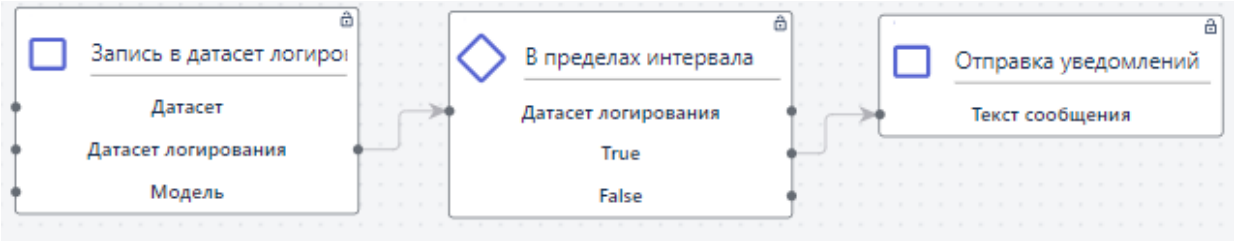


Рисунок 16.9 - Пример построения пайплайна с блоком уведомления

Например, в результате отработки такого пайплайна, пользователи будут получать следующую информацию:

Рисунок 16.10 - Пример уведомлений в телеграм канале

17. Дополнительные возможности Платформы

17.1. Обращение в службу поддержки

В верхнем правом углу окна системы находится кнопка «Сообщить об ошибке»:

Рисунок 17.1 – Кнопка «Сообщить об ошибке»

В случае, если пользователь сталкивается с какой-то проблемой в работе платформы, поведением, не описанным в данном руководстве, есть возможность связаться с командой поддержки системы и сообщить о проблеме.

Пользователю необходимо максимально подробно описать проблему и шаги, которые привели к её появлению в текстовом окне сообщения об ошибке, а после нажать кнопку «Отправить»:

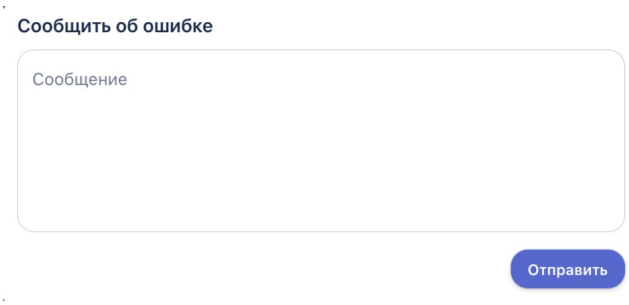


Рисунок 17.2 – Описание найденной в Системе ошибки, и отправка на обработку Разработчику

Ограничение размера сообщения составляет 256 символов.

17.2. История изменений

История изменений - раздел системы, отображающий информацию о том в каком блоке и в какое время было совершено какое либо изменение. Раздел содержит информацию о всех запусках пайплайнов, сформированных отчетах, названия блоков, дату и время каждого изменения.

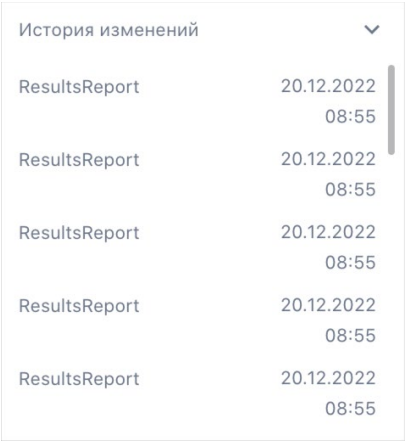


Рисунок 17.3 – История изменений

18. Приложения

Приложение 1. Автоматизированные функции

**Журнал преобразований* – технический объект, который позволяет пробрасывать данные между элементами блок-схемы.
***Выходным параметром для всех функций является *словарь с данными*, который может включать в себя: таблицы, графики, текстовое описание.

Таблица 18.1 – Перечень автоматизированных функций элемента «Источник данных»

Функция	Назначение	Параметры	Выходная информация
Группа «Загрузка данных»			
Загрузка табличных данных	Преобразует загруженные данные во временной ряд. При этом автоматически определяет: формат файла (csv, txt, xls, xlsx), разделитель, размерность, где *разделитель* – это символ в датасете с временным рядом, обозначающий деление на ячейки. Для *ресемплирования* использует агрегирующую функцию	**Выберите файл** – Название файла, содержащего временной ряд для анализа	**
Загрузка изображений для object detection	**** данный функционал находится в разработке, в текущей версии 2.3.3***		В БД сохраняются загруженные датасеты, прошедшие

<table>
<tr>
<td></td>
<td>применение функции недоступно.
Функция предназначена для загрузки в Систему изображений ‘с метками’, с целью дальнейшего решения задачи компьютерного зрения – распознавания отмеченных объектов на новых данных (изображениях/видео).</td>
<td>
<table>
<tr><td>Группа обучающих изображений</td><td>Выбрать папку для обучения нейронной сети с ‘размеченными’ изображениями – на которых с помощью «bounding box» (ограничительной рамки прямоугольной формы) отмечены объекты для распознавания</td></tr>
<tr><td>Группа валидационных изображений</td><td>Выбрать папку для валидации обученной нейронной сети, который также содержит ‘размеченные’ данные. Здесь разметка будет использоваться для оценки качества модели – путем сравнения предсказания и фактического значения</td></tr>
</table>
</td>
<td>предварительную обработку, и вместе с каждым датасетом дополнительно сохраняется файл label – файл разметки.</td>
</tr>
<tr>
<td>Загрузка изображений для классификации</td>
<td>Загружаются изображения с объектами для дальнейшей классификации этих объектов на новых данных. При этом предусмотрено, что изображения с объектами одного класса распределены по отдельным папкам. При загрузке выполняется ресайз изображений – изменение (чаще уменьшение) размера изображений до заданного</td>
<td>
<table>
<tr><td>Группа обучающих изображений</td><td>Выбрать папку для обучения нейронной сети, который представляет собой каталог с папками, где каждая папка содержит изображения с объектами одного класса. Например – папки с изображениями собак, кошек и диких животных (три класса).</td></tr>
<tr><td>Группа тестовых изображений</td><td>Выбрать папку для валидации обученной нейронной сети, который также представлен в виде каталога с папками отдельных классов.</td></tr>
</table>
</td>
<td>В БД сохраняются каталоги датасета обучения и валидации, прошедшие предварительную обработку перед загрузкой.</td>
</tr>
</table>

<table>
<tr><td></td><td>формата. Данные загружаются маленькими порциями, так называемыми мини-батчами (например, за один раз подается два изображения).</td><td><table>
<tr><td>Размер мини-батча</td><td>Указывается количество изображений, которое за один раз подается на вход нейронной сети для её обучения.</td></tr>
<tr><td>Новая высота</td><td>Новая высота изображения</td></tr>
<tr><td>Новая ширина</td><td>Новая ширина изображения</td></tr>
</table></td><td></td></tr>
<tr><td>Загрузка табличных данных из коннектора</td><td>Функция предназначена для подключения к источникам данным в виде баз данных «ClickHouse» или «Postgresql». При этом используется сущность «Коннектор», в которой прописываются настройки для подключения к этим базам данных. Реализована возможность формирования датасета на основании полученных данных. Для этого в настройках функции активируется галочка в поле «Сохранить датасет», и в поле «Название файла» указывается имя для бэкапа таблицы базы данных в настоящий момент времени. В</td><td><table>
<tr><td>Выберите файл</td><td>Из списка выбирается коннектор – источник данных, подключение к которому необходимо выполнить. Источником выступает сторонняя база данных – ClickHouse или Postgresql (соответственно из списка выбирается коннектор с таким типом).</td></tr>
<tr><td>Сохранить датасет</td><td>Чтобы сформировать бэкап таблицы внешней БД в поле «Сохранить датасет» устанавливается галочка.
Иначе, если не установить галочку в поле «Сохранить датасет», выполняется подключение к внешней БД в её состоянии на текущий момент времени, без дополнительного формирования датасета во внутренней БД.</td></tr>
</table></td><td>**</td></tr>
</table>

<table>
<tr>
<td></td>
<td>результате в разделе «Данные» сохраняется файл в формате .csv с данными из коннектора.
Важно – Подключение выполняется к БД в её состоянии на текущий момент времени.</td>
<td>
<table>
<tr><td>Название файла</td><td>Указывается название файла бэкапа таблицы для сохранения во внутренней БД.</td></tr>
<tr><td>Онлайн данные</td><td>Указывается, осуществляется ли получение данных в режиме он лайн.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Загрузка модели</td>
<td>Функция предназначена для использования в качестве источника данных ранее обученной модели. При этом система при обработке данных пайплайна применяет ранее полученные знания для построения прогнозов.</td>
<td>
<table>
<tr><td>Модель</td><td>Выбор из списка ранее сохраненных моделей</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Загрузка текстовых файлов для классификации</td>
<td>Данная функция предназначена для загрузки текстов, принадлежащих к тем или иным классам, для обучения нейронной сети определять эти классы на новых данных. Функция обязательно используется при решении задач классификации текстов.</td>
<td>
<table>
<tr><td>Группа обучающих текстов</td><td>Выбор папки для обучения нейронной сети, которая должна содержать в себе подпапки с названиями классов объектов. Данные подпапки содержат тексты, принадлежащие определенному классу. Например, это могут быть: «Пушкин»,«Лермонтов»,«Толстой».</td></tr>
</table>
</td>
<td></td>
</tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td>Группа тестовых текстов</td><td>Выбор папки для валидации обученной нейронной сети. Папка должна иметь такую же структуру, как и обучающая.</td></tr><tr><td>Группа текстов для классификации</td><td>Здесь можно сразу выбрать файл или папку с файлами, которые необходимо классифицировать с применением обученной модели.</td></tr></table></td><td></td></tr>
<tr><td>Загрузка текстовых файлов для кластеризации</td><td>Функция обязательно используется при решении задач кластеризации текстов, когда необходимо определить кластеры, к которым принадлежат тексты.</td><td><table><tr><td>Группа текстов для кластеризации</td><td>Выбор файла, содержащего однотипные данные, подлежащие разделению на кластеры.</td></tr></table></td><td></td></tr>
<tr><td>Загрузка графа</td><td>Функция предназначена для загрузки и дальнейшего преобразования файлов с форматом .graphml в переменную graph_out с типом данных networkx.MultiDiGraph, предназначенных для решения задач с применением теории графов.</td><td><table><tr><td>Выберите файл с графом</td><td>Выбирается ранее загруженный в систему файл в формате .graphml.</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><td></td><td>Граф — это геометрическая фигура, которая состоит из точек и линий, которые их соединяют. Точки называют вершинами графа, а линии — ребрами. Графы имеют очень широкое применение: с их помощью выбирают наиболее выгодное расположение зданий, графами представлены схемы метро, маршруты, схемы игр, блок схемы процессов и т д.</td><td></td><td></td></tr>
<tr><td colspan="4">Группа «Spark»</td></tr>
<tr><td>Загрузка табличных данных из файла CSV (Spark)</td><td>При помощи данной функции осуществляется загрузка в систему табличных данных с помощью фрйемворка для распределенных вычислений Apache Spark, конкретно, с помощью библиотеки PySpark для Python. Датафрейм в PySpark — это таблица, строки которой хранятся в</td><td><table><tr><td>Выберите файл для загрузки</td><td>Выбор из списка файла для дальнейшего анализа.</td></tr></table></td><td></td></tr>
</table>

<table>
<tr><td></td><td>RDD (Отказоустойчивый распределенный набор данных (англ. Resilient Distributed Dataset, RDD) — тип структуры данных, который можно распределить между несколькими узлами в кластере). Работа с датафреймами ведётся по принципу «ленивых вычислений» (англ. lazy evaluations). Это вычисления, которые откладываются до тех пор, пока пользователь не запросит их результат.
Данная функция работает только для файлов в формате csv, содержащих big data.</td><td></td><td></td></tr>
<tr><td>Загрузка табличных данных из папки CSV (Spark)</td><td>Фреймворк «Apache Spark» распределяет хранимые данные по серверам и директориям. Чтобы обратиться к файлам на уровне папки, в которой они хранятся, используется данный метод.</td><td><table><tr><td>Выберите директорию с датасетом для загрузки</td><td>Выбор из списка папки для дальнейшего анализа.</td></tr></table></td><td></td></tr>
</table>

Загрузка модели	Функция предназначена для использования в качестве источника данных ранее обученной модели ИИ «Spark».	**Модель** — Выбор из списка ранее сохраненных моделей Spark (в разработке отдельное *API* для моделей Spark. Модели будут объединяться в одну группу по семантическому типу).	
Загрузка табличных данных из коннектора (Spark)	Функция предназначена для получения табличных данных через коннектор с типом «ClickHouse», с использованием библиотеки «Spark».	(*в разработке*)	

Таблица 18.2 – Перечень автоматизированных функций элемента «Процесс»

Функция	Назначение	Параметры	Выходная информация
1.Группа «Анализ данных»			
Выбор признаков и целевых признаков	В датасете выбираются: признаки – измеримые характеристики исследуемого объекта/ процесса, и целевые (зависимые) переменные, значения которых предстоит предсказывать модели.	**Признаки** — Характеристики, которые исследуются и выявляется корреляция между ними и рассматриваемым целевым признаком **Целевые признаки** — Предсказываемые переменные	Датасет с размеченными признаками и целевыми признаками
Матрица корреляции	1.Алгоритм сначала рассчитывает		1. Матрица корреляции с топ-k признаков, имеющих

<table>
<tr>
<td></td>
<td>коэффициенты корреляции по всем признакам (общая матрица корреляции).
2.Затем в этой матрице отбираются топ-k максимальных (ближе к 1) значений коэффициентов корреляции.
3.Строится новая матрица корреляции, состоящая из признаков, для которых найдены максимальные значения коэффициентов.</td>
<td><table>
<tr><td>Топ k-значений для корреляции</td><td>Количество максимальных значений корреляции (int)</td></tr>
</table></td>
<td>максимальные значения корреляции.
2.Матрица корреляции по всем признакам.</td>
</tr>
<tr>
<td>Косинусное расстояние</td>
<td>Вычисляется косинусное расстояние между значениями во входном векторе и значениями выбранных столбцов в наблюдениях.</td>
<td><table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Признаки</td><td>Признаки для анализа.</td></tr>
<tr><td>Вектор</td><td>Вектор такой же длины, что и количество выбранных в датасете признаков (1D-array).</td></tr>
</table></td>
<td>Таблица с наблюдениями, наиболее схожими с входным вектором, где первые топ-5 наблюдений выделены жирным шрифтом (по ним рассчитанная косинусная мера имеет значение, наиболее близкое к 0).</td>
</tr>
<tr>
<td>Поиск пропущенных значений</td>
<td>Для каждого выбранного признака метод находит пропущенные значения в наблюдениях.</td>
<td><table>
<tr><td>Признаки</td><td>Выбираются признаки, в которых необходимо найти пропущенные значения.</td></tr>
</table></td>
<td>Словарь, в котором по каждому выбранному признаку отображается количество пропущенных значений в наблюдениях, и процент пропусков.</td>
</tr>
</table>

<table>
<tr>
<td>Анализ временных рядов</td>
<td>Совокупность математико-статистических методов анализа, предназначенных для выявления структуры временных рядов и для их прогнозирования</td>
<td>
<table>
<tr><td>Список графиков</td><td>Выбираются графики, которые будут использованы для визуализации анализа временных рядов и задаются параметры для них</td></tr>
</table>
</td>
<td>Линейный график, ACF/PACF, Декомпозиция, Свечной график, Time profile, Extended, Bollinger Bands, Stochastic Oscillator</td>
</tr>
<tr>
<td>Запись в датасет логирования</td>
<td>Данная функция применяется при построении пайплайнов в режиме реального времени. В процессе логирования записываются все новые поступающие значения в датасет для дальнейшего использования при валидации и работе с моделью. При этом в процессе логирования записываются:
- фактические значения
- промежуточные транформации при препроцессинге до подачи в модель (если в качестве источника</td>
<td>-</td>
<td>Датасет с данными</td>
</tr>
</table>

<table>
<tr><td></td><td>данных выбрана модель)
- прогнозные значения (если в пайплайне настроен прогноз)</td><td></td><td></td></tr>
<tr><td>Визуализация Real-time</td><td>Представление в виде графиков и диаграмм результатов работы пайплайна, содержащего данные, получаемые в режиме реального времени</td><td><table><tr><td>Список графиков</td><td>Выбираются графики, которые будут использованы для визуализации данных в режиме он лайн и задаются параметры для них</td></tr></table></td><td>Линейный график, Свечной график, Time Profile, Extended, Bollinger Bands, Stochastic Oscillator</td></tr>
<tr><td colspan="4">Загрузка данных</td></tr>
<tr><td>Преобразование данных во временной ряд</td><td>Метод редактирует исходные данные, исключая в них аномалии и искаженные наблюдения, которые могли быть зафиксированы в результате помех. Далее выполняется дискретизация – определяются точки (моменты времени), в которых должны быть произведены выборки значений. Дискретизация производится через</td><td><table><tr><td>Шаг ресемплирования</td><td>Дискретность для временного ряда – частота фиксирования наблюдений, значения начиная с нано-, микро-, милли-, секунд и заканчивая годами. Указывается оптимальный интервал дискретности.</td></tr><tr><td>Частота ресемплирования</td><td>Единица измерения, в которой фиксируются наблюдения.</td></tr><tr><td>Агрегирующая функция</td><td>Функция, вычисляющая результат по набору значений группы, где группа – наблюдения в пределах шага ресемплирования. По умолчанию, значение вычисляется функцией</td></tr></table></td><td>Создается временной ряд, с заданным шагом ресемплирования</td></tr>
</table>

<table>
<tr><td></td><td>равные промежутки времени.</td><td><table>
<tr><td></td><td>медианы.</td></tr>
<tr><td>Столбец с временной меткой</td><td>Время фиксирования наблюдения. По умолчанию, нулевой столбец в датасете.</td></tr>
</table></td><td></td></tr>
<tr><td colspan="4">Препроцессинг</td></tr>
<tr><td>Стабилизация дисперсии</td><td>Уменьшает разброс исследуемых данных, чтобы сделать их более компактными и пригодными для работы.</td><td><table>
<tr><td>Замена значений столбцов</td><td>Преобразование оригинального временного ряда, загруженного в систему. Позволяет заменять трансформируемые столбцы или добавлять новые</td></tr>
<tr><td>Стандартизация</td><td>Преобразование значений признака, адаптирующая признаки с разными диапазонами значений к моделям машинного обучения</td></tr>
<tr><td>Метод</td><td>Выбирается метод, с помощью которого проводится стабилизация дисперсии – приведение данных к нормальному распределению. На выбор два метода – yeo-johnson и box-cox. Метод yeo-johnson работает как с отрицательными, так и с положительными значениями, а метод box-cox только с положительными</td></tr>
</table></td><td>Преобразованный датасет. При этом преобразования над целевыми признаками проводятся отдельно.</td></tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td>Флаг признака</td><td>Показатели датасета, значения которых предстоит предсказывать модели машинного обучения</td></tr></table></td><td></td></tr>
<tr><td>Стандартизация</td><td>Чтобы сгладить большие различия между диапазонами признаков датасета и предотвратить искаженное восприятие данных моделью машинного обучения выполняется стандартизация – преобразование и приведение признаков датасета к единому формату</td><td><table><tr><td>Замена значений столбцов</td><td>Подтверждение преобразования оригинального временного ряда (заменой трансформируемых столбцов или добавлением новых)</td></tr><tr><td>Флаг признака</td><td>Выбрать столбцы для преобразования – все, кроме столбцов с датой и целевым признаком</td></tr></table></td><td>Преобразованные значения показателей временного ряда, кроме целевого признака</td></tr>
<tr><td>Дифференцирование временного ряда</td><td>Выполняется дифференцирование целевых признаков (таргетов) временного ряда. При этом временной ряд сдвигается на указанное число шагов в разрезе каждого целевого признака. Если есть сезонность, сначала проводится сезонное дифференцирование. Желательно</td><td><table><tr><td>Шаг дифференцирования для каждого целевого признака</td><td>Есть возможность задать шаг дифференцирования для каждого таргета, в формате [сдвиг для таргета 1, свиг для таргета 2, …], где сдвиг на один шаг применяется для обычного (для избавления от тренда) дифференцирования, сдвиг на несколько шагов – для сезонного, а сдвиг, равный нулю означает, что дифференцирование для данного таргета не проводится. Например, [1, 0, 3].</td></tr></table></td><td>К датасету временного ряда добавляются новые столбцы с окончанием '_diff' для каждого указанного таргета. При этом замена колонок не предусмотрена – оригинальные колонки сохраняются для задачи обратного дифференцирования. Отображаются графики настоящего и сдвинутого временных рядов.</td></tr>
</table>

<table>
<tr><td></td><td>дифференцировать ряд как можно меньше раз, потому что с увеличением количества дифференцирований растет дисперсия ошибки прогноза</td><td></td><td></td></tr>
<tr><td>One-Hot Encoding</td><td>Метод One Hot Encoding (OHE) применяется, когда в датасете необходимо закодировать категориальные признаки (текстовые), перед подачей в модель. Для кодируемого категориального признака создаются N новых столбцов в датасете, где N – количество уникальных категорий. Значения в новых столбцах – 0 или 1, в зависимости от принадлежности к категории. Так каждый новый признак – бинарный характеристический признак категории.</td><td><table>
<tr><td>Флаг удаления первого признака</td><td>Устанавливается, чтобы удалить из итоговой таблицы столбец с признаком, над которым были выполнены преобразования. Так как новые столбцы отражают принадлежность наблюдения к той или иной категории признака, удаление первого признака не повлияет на результат.</td></tr>
<tr><td>Флаг признака</td><td>Выбираются столбцы, над которыми будут осуществляться преобразования. Значения на выбор – признаки или таргеты.</td></tr>
</table></td><td>Таблица с новыми столбцами, в которых отражается принадлежность наблюдений к тем или иным категориям преобразованных признаков.</td></tr>
</table>

<table>
<tr>
<td>Создание признаков для временного ряда</td>
<td>Для временного ряда создаются новые признаки, в которых значения таргетов сдвигаются на указанное число шагов. Например, если для одномерного (с одним таргетом) временного ряда задать сдвиг в один шаг, создается новая колонка y, в которой значение первой строки равно значению второй строки в колонке с таргетом x (категориальный признак), т.е. значения сдвигаются на один шаг вперед. Если ряд многомерный – состоящий из нескольких таргетов, то для каждого из них передается общий массив признаков, с учетом лагов всех таргет-рядов.
Такое действие является предварительным перед тем, как подавать данные в модель ИИ, чтобы у</td>
<td>
<table>
<tr><td>Максимальное количество лагов</td><td>Указывается, на какое количество шагов может быть сдвинут временной ряд.</td></tr>
</table>
</td>
<td>Создается таблица, график со смещенным временным рядом.
Процесс создания признаков-лагов сохраняется в журнале преобразований, и далее отрабатывается при препроцессинге.</td>
</tr>
</table>

<table>
<tr><td></td><td>модели были не только фактические значения таргета, но и прогнозные.</td><td></td><td></td></tr>
<tr><td>Препроцессинг текстовых данных</td><td>Алгоритмы машинного обучения не работают с «сырыми данными». Большая часть процесса – это подготовка текста, преобразование ее в вид, доступный для восприятия компьютером.
В первую очередь выполняется очистка текста. Из текста удаляются бесполезные для машины данные – это большинство знаков пунктуации, особые символы, скобки, теги и т.д.
Дальше наступает большой этап предварительной</td><td><table><tr><td>Метод векторизации</td><td>1. TF-IDF. С англ. TF – term frequency (частота слова), IDF – inverse document frequency (обратная частота документа). Это мера, используемая для оценки важности слова в контексте документа, являющегося частью коллекции документов. Вес некоторого слова пропорционален частоте употребления этого слова в документе, и обратно пропорционален частоте употребления слова во всех документах коллекции.
Метод TF-IDF используется в задачах анализа текстов, и представляет собой линейный классификатор с разреженными признаками, взвешенными по частоте.
Этот метод выбирается по умолчанию.
2. Word2vec. Принимает большой текстовый корпус в качестве входных</td></tr></table></td><td>Числовые векторы, созданные на основе исходной текстовой информации, которые отражают важность использования каждого слова.</td></tr>
</table>

	обработки – *препроцессинга*. Это приведение информации к виду, в котором она более понятна алгоритму. Методы препроцессинга: -приведение символов к одному регистру; -*токенизация* – разбиение текста на токены. Так называют отдельные компоненты – слова, предложения или фразы; -*лемматизация* – приведение слов к изначальным словоформам, часто с учетом контекста; -удаление *стоп-слов* – артиклей, междометий и пр.; После предподготовки на выходе получается набор подготовленных слов. Но алгоритмы работают с числовыми данными, поэтому из входящей информации создают *векторы* – представляют		данных, и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он генерирует словарь корпуса, а затем вычисляет векторное представление слов, ‘обучаясь’ на входных тестах. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте с одинаковыми словами (а следовательно, имеющие схожий смысл), будут иметь близкие (по *косинусному расстоянию*) векторы. Полученные векторные представления слов используются для обработки естественного языка и машинного обучения.	
		Использование GPU и нейронной сети	В данном поле устанавливается галочка, когда обучение модели ИИ происходит с использованием нейронной сети.	

	ее как набор числовых значений. На Платформе используются следующие методы векторизации – TF-IDF, и Word2vec.		
Кодирование целевого признака	Данная функция применяется, когда необходимо преобразовать *категориальный целевой признак в датасете* в числовое значение. Такое преобразование выполняется перед подачей входных данных в алгоритм. Правила перевода категориальный признаков в числовые прописываются в *кодировщике*. Данная функция представляет собой первый тип кодирования – **Label Encoder**. Выполняется *порядковое кодирование* всех уникальных значений категориального	Для данной функции не предусмотрен ручной ввод параметров пользователем. На вход функции подается *датасет* с входными данными, над которыми уже выявлены некоторые преобразования (очистка, выделение целевого признака, и т.д.), и файл *vars_dict*, в котором содержится информация о выполненных преобразованиях над данными.	Закодированный целевой признак в датасете.

	признака: первое (выбранное каким-то образом) уникальное значение кодируется нулем, второе единицей, и так далее, последнее кодируется числом, равным количеству уникальных значений минус единица.		
Порядковое кодирование категориальных признаков	Отличие данной функции в том, что она выполняет преобразование всех *категориальных признаков датасета* в числовые значения. При этом кодировщик используется тот же, что и в предыдущей функции – Label Encoder, но кодируются признаки. Выполняется порядковое кодирование каждой категориальной переменной (кроме целевой).	Для данной функции не предусмотрен ручной ввод параметров.	Закодированные категориальные признаки в датасете.
Предобработка данных			

<table>
<tr>
<td>Заполнение пропусков</td>
<td>Позволяет заполнять пропущенные значения в датасете одним из следующих способов: среднее, мода, медиана, квантили 0.25, 0.5, 0.75 или по выбору, min, max. Функция применяется для датасетов, созданных из табличных данных. При этом при загрузке файла в систему можно посмотреть количество пропусков (пустых ячеек) в датасете, если для одного из признаков пропусков слишком много, предусмотрена возможность удаления его полностью.</td>
<td>
<table>
<tr><td>Индекс столбца</td><td>номер столбца</td></tr>
<tr><td>Способ заполнения пропусков</td><td>mean, mode, median, max, min, quantile25, quantile50, quantile75
drop - удаление</td></tr>
</table>
</td>
<td>Датасет с заполненными пропусками</td>
</tr>
<tr>
<td>Сглаживание временного ряда</td>
<td>Позволяет исключить влияние шума в данных и увидеть структуру временного ряда. Для сглаживания применяется метод центрированного скользящего среднего: по временному ряду «скользит окно»</td>
<td>
<table>
<tr><td>Список признаков</td><td>Выбираются признаки для расчета скользящего среднего</td></tr>
<tr><td>Размер окна для сглаживания</td><td>Временное окно анализа, определяется количеством входящих в него наблюдений. Например, размер окна три, тогда берутся первые три наблюдения и по ним считаются</td></tr>
</table>
</td>
<td>1. Таблица, содержащая сглаженные значения признаков.
2. График исходных данных и сглаженного временного ряда.</td>
</tr>
</table>

	определенного размера, в рамках окна значения группируются и по ним рассчитываются средние значения.	<table><tr><td></td><td>средние значения признаков</td></tr></table>	
Срез временного ряда по индексу	Позволяет создавать выборки данных за период времени, используя временные метки или временные диапазоны.	<table><tr><td>**Дата начала**</td><td>Дата начала среза</td></tr><tr><td>**Дата окончания**</td><td>Дата окончания среза</td></tr></table>	Временной ряд после применения фильтра.
Фильтрация текстового шума	Данная функция позволяет очистить текст от шумов: из текста убираются знаки препинания, заглавные буквы (они заменяются на строчные) и стоп-слова (различные служебные части речи - союзы, предлоги, частицы и т.д.)	-	Текст без шумов
Лемматизация текста	Лемматизация - это процесс приведения всех встречающихся форм слова к одной, нормальной словарной форме. В процессе лемматизации платформа использует словарь и морфологический анализ,	-	Нормализованный текст

<table>
<tr><td></td><td>чтобы привести слово к его канонической форме – т.н. «лемме», в итоге получается текст, состоящий из слов приведенных к единственному числу, мужскому роду, именительному падежу и инфинитиву (в зависимости от части речи).</td><td></td><td></td></tr>
<tr><td>Векторизация текста</td><td>Векторизация текста - это процесс преобразования слов в векторы (числа), которые являются «читаемым» форматом для алгоритмов машинного обучения.</td><td><table>
<tr><td>Метод векторизации</td><td>1.TD IDF - метод, используемый для оценки важности слова в контексте документа. Вес некоторого слова пропорционален частоте употребления этого слова в документе и обратно пропорционален частоте употребления слова во всех документах коллекции.
2.Word to Vec - данный метод использует контекст, чтобы сформировать численные представления слов, в результате слова, используемые в одном и том же контексте, имеют похожие векторы.</td></tr>
<tr><td>Максимальная размерность</td><td>Указывается примерное количество уникальных слов в тексте.</td></tr>
</table></td><td></td></tr>
</table>

<table>
<tr><td></td><td></td><td><table>
<tr><td>текста</td><td></td></tr>
<tr><td>Количество признаков</td><td>Указывается количество столбцов таблицы, которая получится в результате преобразования текста в числовой вид</td></tr>
<tr><td>Сгенерировать тензор для GPU</td><td>Выбирается в случае, если предполагается что дальше будет использоваться графический процессор. Тензор - это просто таблица особого вида</td></tr>
</table></td><td></td></tr>
<tr><td colspan="4">Тесты на нормальность распределения</td></tr>
<tr><td>Коэффициент асимметрии Skewness</td><td>Данный метод проверяет выборку на нормальность распределения путем расчета асимметрии данных. Если правый хвост асимметрии длиннее левого, то коэффициент положителен, иначе – отрицателен. Если распределение симметрично (в форме ‘колокола’), коэффициент равен нулю.</td><td><table>
<tr><td>Признаки</td><td>Выбираются все признаки в датасете для расчета коэффициента асимметрии.</td></tr>
</table></td><td>Словарь с данными.</td></tr>
<tr><td colspan="4">Тесты на стационарность временного ряда</td></tr>
<tr><td>Тест Дики-Фуллера</td><td>Проверяется, является ли временной ряд стационарным – не</td><td><table>
<tr><td>Пороговое</td><td>Задается пороговое значение p из теста</td></tr>
</table></td><td>Результаты теста Дики-Фуллера.</td></tr>
</table>

	влияют ли на него тренды и сезонность. Для такого ряда суммарные статистические данные согласованы по времени, например, *среднее значение* и *дисперсия наблюдений*. Стационарность влияет на легкость моделирования – часто требуется, чтобы временной ряд был стационарным, чтобы быть эффективным.	**значение alpha**	Дики-Фуллера, с использованием которого интерпретируются результаты гипотез: ● *Нулевая гипотеза* – временной ряд имеет единичный корень, то есть он нестационарный; ● *Альтернативная гипотеза* – нулевая гипотеза отвергается, и предполагается, что временной ряд не имеет единичного корня, то есть он является стационарным. Значение p ниже порогового значения означает, что отвергается нулевая гипотеза и временной ряд *стационарный*. Значение p выше порогового значения означает, что подтверждается нулевая гипотеза и временной ряд *нестационарный*. Значение *p* задается в формате числа с плавающей точкой (float).	
2.Группа «Машинное обучение»					
Валидация модели	На тестовой выборке данных (обычно это 20% датасета) проверяется правильность работы (предсказательная способность) модели ИИ,	**Метрика**	Из списка выбирается название метрики для валидации. Для задачи *классификации*: Accuracy, F1, Precision, Recall, AUC_ROC. Для задачи *регрессии*: RMSE, MAE, WMAPE.		Таблица со значением выбранной метрики, отражающей количество правильных ответов обученной модели на тестовой выборке данных

<table>
<tr><td></td><td>построенной на основе машинного обучения.</td><td><table><tr><td>–</td><td>Журнал преобразований над данными.</td></tr><tr><td>–</td><td>Обученная модель ИИ.</td></tr></table></td><td>(максимальное значение метрики равно 1).</td></tr>
<tr><td>Прогноз модели</td><td>Выполняется последовательность действий по прогнозированию будущих значений целевых признаков.</td><td>-</td><td>1. Точность прогноза.
2. Словарь с данными.
3. Датасет логирования.</td></tr>
<tr><td>Разделение датасета на обучающую и тестовую выборки</td><td>Разделение выборки данных на две категории: для обучения модели ИИ, и для проверки результатов обучения.</td><td><table><tr><td>Доля тестовой выборки в датасете</td><td>Обычно на 80% датасета выполняется обучение модели, а на оставшихся 20% – ее валидация. Значение указывается в формате 0.2.</td></tr><tr><td>Перемешивать наблюдения перед разделением</td><td>Необходимо установить галочку в поле, чтобы случайным образом поменять местами наблюдения в датасете. Не рекомендуется перемешивать временные ряды, т.к. наблюдения в них упорядочены и зафиксированы последовательно по времени.</td></tr><tr><td>Разделять с учетом меток классов</td><td>Выбирается, учитывать ли долю таргетов при разделении датасета. Используется только для задач классификации, когда объекты распределяются по категориям согласно определенным и заданным заранее</td></tr></table></td><td>1. Отдельно обучающая и тестовая выборки.
2. Журнал преобразований.</td></tr>
</table>

<table>
<tr><td></td><td></td><td><table><tr><td></td><td>признакам.</td></tr></table></td><td></td></tr>
<tr><td colspan="4">Классификация решает задачу разделения множества наблюдений (объектов) на группы, называемые классами, на основе анализа их формального описания. При классификации каждое наблюдение относится к определенной группе на основе некоторого качественного свойства. Пусть X – множество описаний объектов, Y – конечное множество номеров/имен/меток классов. Существует неизвестная целевая зависимость отображения $y^*: X \to Y$, значения которой известны только на объектах обучающей выборки $X^m = (x_1, y_1), \ldots, (x_m, y_m)$. Строится алгоритм, способный классифицировать произвольный объект $x \in X$.</td></tr>
<tr><td>Логистическая Регрессия</td><td>Используется логистическая функция для моделирования зависимости выходной переменной y от набора входных переменных x, в случае, когда первая является бинарной. Например, с помощью логистической регрессии можно оценивать вероятность наступления/или не наступления некоторого события.
Предсказывается непрерывная переменная – коэффициент логистической регрессии, принимающий значение от 0 до 1:</td><td><table><tr><td>Коэффициент регуляризации</td><td>Указывается значение строго больше нуля – положительное вещественное число, с помощью которого добавляется дополнительное ограничение к условию с целью предотвратить переобучение модели.</td></tr><tr><td>Порог классификации</td><td>Значение вещественного типа от 0 до 1, определяющее принадлежность объекта к тому или иному классу.</td></tr><tr><td>Флаг возврата вероятности при прогнозе</td><td>Возвращает вероятность или метки классов для дальнейшего прогноза после обучения модели. Используется для решения задач бинарной классификации, когда выходная переменная может принимать только два значения – решается вопрос о принадлежности объекта к одному из двух классов.</td></tr></table></td><td>1. Модель бинарной классификации.
2. Словарь с данными.
3. Точность модели.
4. Журнал преобразований.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>● если значение коэффициента больше порогового значения, то вероятность наступления события равна 1;
● иначе вероятность наступления события равна 0.</td>
<td>
<table>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров. Флаг активируется, когда указывается несколько гиперпараметров.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Критерий остановки итераций. Настройка, позволяющая определить точность нахождения минимума функции ошибки.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Датасет делится на фолды – на указанное количество равных частей. При обучении модели каждый фолд становится валидационным один раз, при этом на остальных фолдах выполняется обучение. Каждый раз рассчитывается значение метрики. Затем рассчитывается усредненная метрика, которая характеризует точность модели.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Модель XGB Classifier</td>
<td>Алгоритм XGBClassifier анализирует связь между признаками и целевым признаком. На обучающей выборке модель обучается соотносить наблюдение к аномалиям, а на тестовой выборке выполняется</td>
<td>
<table>
<tr><td>Глубина дерева</td><td>Заданное максимальное число разбиений в ветвях, по достижению которого обучение модели ИИ останавливается.</td></tr>
<tr><td>Количество базовых моделей</td><td>Определяет сколько независимых моделей будет работать над обучением.</td></tr>
</table>
</td>
<td>1. Таблица с матрицей ошибок.
2. Таблица верных и ошибочных прогнозов модели в разрезе классов.
3. Модель бинарной классификации.</td>
</tr>
</table>

	валидация ответов обученной модели.		**Порог классификации**	Значение от 0 до 1, указывающее на верхнюю границу вероятности причисления объекта к классу.		
			Флаг возврата вероятности при прогнозе	Возвращает вероятность или метки классов для дальнейшего прогноза после обучения модели. Используется для решения задач бинарной классификации, когда выходная переменная может принимать только два значения – решается вопрос о принадлежности объекта к одному из двух классов.		
			Оптимизация гиперпараметров	Флаг подбора гиперпараметров. Флаг активируется, когда указывается несколько гиперпараметров.		
			Метрика для оптимизации	Критерий остановки итераций. Настройка, позволяющая определить точность нахождения минимума функции ошибки.		
			Количество фолдов для оптимизации.	Датасет делится на фолды – на указанное количество равных частей. При обучении модели каждый фолд становится валидационным один раз, при этом на остальных фолдах выполняется обучение. Каждый раз рассчитывается значение метрики.		

<table>
<tr><td></td><td></td><td><table><tr><td></td><td>Затем рассчитывается усредненная метрика, которая характеризует точность модели.</td></tr></table></td><td></td></tr>
<tr><td>Дерево решений для классификации</td><td>Предсказывает, к какому классу принадлежит объект из обучающего массива данных. Для этого строится дерево решений: древовидная структура, где моменты принятий решений соответствуют узлам, в узлах происходит ветвление процесса на ветки в зависимости от сделанного выбора, и конечные узлы (листья) – конечные результаты последовательного принятия решений. В узлах, начиная с корневого, выбирается признак, значение которого используется для разбиения всех данных на два класса. Процесс продолжается до тех пор, пока не выполнится критерий</td><td><table>
<tr><td>Глубина дерева</td><td>Заданное максимальное число разбиений в ветвях, по достижению которого обучение модели ИИ останавливается.</td></tr>
<tr><td>Порог классификации</td><td>Значение порога определяет принадлежность объекта к одному из классов: к положительному – если порог выше указанного значения, к отрицательному – если порог ниже.</td></tr>
<tr><td>Флаг возврата вероятности при прогнозе</td><td>Возвращает вероятность или метки классов для дальнейшего прогноза после обучения модели.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>При выборе флага оптимизации не нужно вручную задавать глубину дерева, или можно задать несколько значений на выбор. Алгоритм подбирает глубину дерева из расчета получения максимального значения метрики.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Метод, который рассчитывает точность обученной модели.</td></tr>
</table></td><td>1. Датасет с меткой класса, определяющей принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.
4. Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td></tr>
</table>

<table>
<tr>
<td></td>
<td>остановки – дерево превысило заранее заданный «лимит роста» (достигнута глубина дерева). При этом разбиения выполняются таким образом, чтобы уменьшить выбранный критерий, например энтропию – степень неопределенности в разбиении на классы.</td>
<td>
<table>
<tr><td></td><td>Выбирается один из предлагаемых методов.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Датасет делится на фолды – на указанное количество равных частей. При обучении модели каждый фолд становится валидационным один раз, при этом на остальных фолдах выполняется обучение. Каждый раз рассчитывается значение метрики. Затем рассчитывается усредненная метрика, которая характеризует точность модели.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Случайный лес для классификации</td>
<td>Строится множество решающих деревьев, и в задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по схеме:
1.Выбирается подвыборка обучающей выборки и по ней строится дерево.
2.Для построения каждого расщепления в дереве просматривается максимальное количество случайных признаков.</td>
<td>
<table>
<tr><td>Глубина дерева</td><td>Максимальная глубина для деревьев решений.</td></tr>
<tr><td>Количество деревьев</td><td>Число деревьев в «лесу».</td></tr>
<tr><td>Порог классификации</td><td>–</td></tr>
<tr><td>Флаг возврата вероятности при прогнозе</td><td>–</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>3.Выбирается наилучший признак и расщепление по нему (по заранее заданному критерию). Дерево строится, до достижения параметра, ограничивающего его высоту.
Таким образом деревья обучаются не только на разных наборах данных, но и используют разные признаки для принятия решений – это создает некоррелированные деревья, которые и защищают друг друга от своих ошибок. Прогноз получается точнее, чем у любого отдельного дерева.</td>
<td>
<table>
<tr><td>Оптимизация гиперпараметров</td><td>Необходимо активировать галочку в поле, чтобы подобрать гиперпараметры – глубину и количество деревьев. Гиперпараметры подбираются таким образом, чтобы получить максимальное значение метрики.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Categorical Naive Bayes</td>
<td>Группа байесовских классификаторов позволяет определить к какому классу принадлежит объект на основе теоремы Байеса с допущением о независимости признаков.</td>
<td>
<table>
<tr><td>Параметр сглаживания Лапласа</td><td>Значение для аддитивного сглаживания Лапласа во избежание проблемы нулевой вероятности. Если равен нулю, то сглаживания нет.</td></tr>
</table>
</td>
<td>1. Датасет с меткой класса, определяющей принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>Категориальный наивный байесовский классификатор применяется для признаков с категориальным распределением.</td>
<td>
<table>
<tr><td>Априорные вероятности классов</td><td>Определяет, будут ли взяты в расчет априорные вероятности классов. Если не активирован, то применяются значения вероятностей для равномерного распределения.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Определяет, будет ли проводиться оптимизация гиперпараметров при обучении модели</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr>
</table>
</td>
<td>4. Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td>
</tr>
<tr>
<td>Multinomial Naive Bayes</td>
<td>Мультиномиальный классификатор применяется для признаков с полиномиальным распределением. Пример: классификация текстов, где каждый текст представлен вектором слов (например, мешок слов или tf-idf).</td>
<td>
<table>
<tr><td>Параметр сглаживания Лапласа</td><td>Значение для аддитивного сглаживания Лапласа во избежание проблемы нулевой вероятности. Если равен нулю, то сглаживания нет.</td></tr>
<tr><td>Априорные вероятности классов</td><td>Определяет, будут ли взяты в расчет априорные вероятности классов. Если не активирован, то применяются значения вероятностей для равномерного распределения.</td></tr>
</table>
</td>
<td>1. Датасет с меткой класса, определяющей принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.
4.Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td><table>
<tr><td>Оптимизация гиперпарамет ров</td><td>Определяет, будет ли проводиться оптимизация гиперпараметров при обучении модели</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr>
</table></td><td></td></tr>
<tr><td>Complement Naive Bayes</td><td>Представляет собой вариант адаптации Multinomial Naive Bayes для датасетов с несбалансированными классами.
Вместо вычисления вероятностей принадлежности объекта к конкретному классу для каждого класса вычисляются вероятности того, что объект им не принадлежит.
Выбирается наименьшая вероятность "непринадлежности" к</td><td><table>
<tr><td>Параметр сглаживания Лапласа</td><td>Значение для аддитивного сглаживания Лапласа во избежание проблемы нулевой вероятности. Если равен нулю, то сглаживания нет.</td></tr>
<tr><td>Априорные вероятности классов</td><td>Определяет, будут ли взяты в расчет априорные вероятности классов. Если не активирован, то применяются значения вероятностей для равномерного распределения.</td></tr>
<tr><td>Оптимизация гиперпарамет ров</td><td>Определяет, будет ли проводиться оптимизация гиперпараметров при обучении модели</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
</table></td><td>1. Датасет с меткой класса, определяющей принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.
4. Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td></tr>
</table>

<table>
<tr><td></td><td>классу, так как это означает, что объект с наибольшей вероятностью принадлежит к данному классу.</td><td><table><tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr></table></td><td></td></tr>
<tr><td>Gaussian Naive Bayes</td><td>Для значений признаков для каждого класса строится распределение Гаусса. В качестве значений правдоподобия для признаков берутся значения функции Гаусса из конкретного распределения (соответствующее признаку и классу).</td><td><table><tr><td>Параметр сглаживания Лапласа</td><td>Значение для аддитивного сглаживания Лапласа во избежание проблемы нулевой вероятности. Если равен нулю, то сглаживания нет.</td></tr><tr><td>Априорные вероятности классов</td><td>Определяет, будут ли взяты в расчет априорные вероятности классов. Если не активирован, то применяются значения вероятностей для равномерного распределения.</td></tr><tr><td>Оптимизация гиперпараметров</td><td>Определяет, будет ли проводиться оптимизация гиперпараметров при обучении модели</td></tr><tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr><tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr></table></td><td>1. Датасет с меткой класса, определяющей принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.
4. Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td></tr>
<tr><td>Bernoulli Naive Bayes</td><td>Применяется для признаков с</td><td></td><td>1. Датасет с меткой класса, определяющей</td></tr>
</table>

<table>
<tr>
<td></td>
<td>биномиальным распределением. Пример: классификация текстов, где каждый текст представлен вектором наличия слов из словаря (1 - есть слово, 0 - нет).</td>
<td>
<table>
<tr><td>Параметр сглаживания Лапласа</td><td>Значение для аддитивного сглаживания Лапласа во избежание проблемы нулевой вероятности. Если равен нулю, то сглаживания нет.</td></tr>
<tr><td>Априорные вероятности классов</td><td>Определяет, будут ли взяты в расчет априорные вероятности классов. Если не активирован, то применяются значения вероятностей для равномерного распределения.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Определяет, будет ли проводиться оптимизация гиперпараметров при обучении модели</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr>
</table>
</td>
<td></td>
<td>принадлежность объекта к одному из классов.
2. Модель ИИ, обученная классифицировать данные по заданным критериям.
3. Журнал преобразований над данными.
4. Словарь с переменными (описание модели, таблицы, графики) для отображения в интерфейсе Программы.</td>
</tr>
<tr>
<td colspan="5">Кластеризация – это задача группировки множества объектов на подмножества (кластеры) так, чтобы объекты одного кластера были более похожи друг на друга, чем на объекты других кластеров по какому-либо критерию. Относится к классу задач обучения без учителя.</td>
</tr>
</table>

<table>
<tr>
<td>Алгоритм кластеризации DBSCAN</td>
<td>Алгоритм DBScan формирует группы коренных соседей/кластеры, объединяя точки, расположенные рядом. А точки, которые не попадают ни в одну из групп, отмечаются меткой -1 и приравниваются к аномалиям.</td>
<td>
<table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Журнал преобразований</td><td>–</td></tr>
<tr><td>Радиус</td><td>Радиус в единицах расстояния, в рамках которого выполняется поиск потенциальных соседей (float/list/tuple).</td></tr>
<tr><td>Число соседей</td><td>Минимальное число ближайших соседей в указанном радиусе для формирования группы коренных соседей (int/list/tuple).</td></tr>
<tr><td>Метрика расстояния</td><td>Метрика расстояния (str/list): расстояние Евклида, косинусное расстояние. По умолчанию «Евклидово расстояние» – используется при кластеризации данных в текущем датасете, а также при отнесении нового объекта к кластеру.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров. При значении «true» выполняется ручной ввод следующих гиперпараметров: радиус, число соседей, метрика расстояния. При значении «false» эти</td></tr>
</table>
</td>
<td>1. Модель кластеризации.
2. Выходной датасет, дополненный меткой кластера и/или флагом аномалии.
3. Словарь, содержащий информацию (графики, таблицы, текст) для отображения в пользовательском интерфейсе.
4. Журнал преобразований.
**Для алгоритмов кластеризации, регрессии, классификации наборы выходных параметров идентичны, отличие заключается в их содержимом.</td>
</tr>
</table>

			гиперпараметры подбираются автоматически.	
		* Параметры: *датасет* и *журнал преобразований* являются входными и выходными параметрами для всех алгоритмов.		

<table>
<tr>
<td>Изоляционный лес</td>
<td>Алгоритм поиска аномалий (выбросов) методом «Изоляционный лес»:
Изолирует наблюдения, случайным образом выбирая объект, а затем случайным образом выбирая разделения между максимальным и минимальным значениями объекта. Разбиение представлено древовидной структурой, количество разбиений, необходимое для изоляции выборки, равно длине пути от корневого до конечного узла. Эта длина пути является мерой нормальности и функции принятия решений. Когда лес случайных деревьев создает более короткие пути для отдельных объектов, они, скорее всего, являются аномалиями.</td>
<td>
<table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Журнал преобразований</td><td>–</td></tr>
<tr><td>Количество деревьев</td><td>По умолчанию устанавливается значение, равное 2</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td>Кластеризация K-Means</td>
<td>Алгоритм кластеризации K-средних:
1. Из исходного множества случайным образом выбирается K наблюдений, равное заданному количеству кластеров.
2. Для каждого наблюдения определяется ближайший к нему центр кластера (измеряется Евклидового расстояние до центра). Образуются начальные кластеры.
3. Вычисляются центры тяжести кластеров – вектора, элементы которых представляют собой среднее арифметическое значение признаков кластера.
4. Центры кластеров смещаются и объединяют вокруг себя наблюдения, пока центры и границы кластеров не перестанут изменяться.</td>
<td>
<table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Журнал преобразований</td><td>–</td></tr>
<tr><td>Число кластеров</td><td>По умолчанию устанавливается значение, равное 2.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров.</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td>Агломеративная иерархическая кластеризация</td>
<td>Последовательно объединяет объекты во все более крупные подмножества, в результате образуется древовидная структура. Отдельные версии иерархии отличаются правилами вычисления расстояния между кластерами. Например, алгоритм средней связи на каждом шаге объединяет два ближайших кластера, рассчитывая среднюю арифметическую дистанцию между всеми парами объектов.</td>
<td>
<table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Журнал преобразований</td><td>–</td></tr>
<tr><td>Число кластеров</td><td>Задается оптимальное количество кластеров.</td></tr>
<tr><td>Метрика расчета расстояния</td><td>Используется при кластеризации данных в текущем датасете, а также при отнесении нового объекта к кластеру. Значения: евклидово расстояние, косинусная мера, расстояние городских кварталов, расстояние Чебышева.</td></tr>
<tr><td>Критерий связи для расчета расстояния</td><td>Правила вычисления расстояния между кластерами при каждой итерации. Значения:
-алгоритм средней связи,
-алгоритм одиночной связи или ближайшего соседа,
-алгоритм полной связи или дальнего соседа,
-метод минимума дисперсии Уорда.
Дисперсия объединяет кластеры с минимальной общей внутрикластерной дисперсией после слияния, в качестве метрики</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
<table>
<tr><td></td><td>расстояния используется евклидово расстояние. Минимальный использует самые близкие точки в обоих кластерах для расчета расстояния.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Метод локтя K-Means</td>
<td>Метод локтя позволяет вычислить правильное значение k (количество кластеров) и повысить производительность модели. Вычисляется сумма квадратов расстояний между точками, и среднее арифметическое значение (Mean) – сумма элементов датасета, разделенная на их количество. Когда значение k равно 1, сумма квадрата внутри кластера будет большой. По мере увеличения значения k сумма квадратов внутри кластера будет уменьшаться. Наконец будет построен график</td>
<td>
<table>
<tr><td>Число кластеров</td><td>Задается оптимальное количество кластеров.</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr><td></td><td>между значениями k и суммой квадрата внутри кластера. В момент, когда значение k резко уменьшится будет считаться оптимальным числом кластеров.</td><td></td><td></td></tr>
<tr><td colspan="4">Регрессия – математическое выражение, отражающее связь между зависимой переменной y и независимыми переменными x. Алгоритмы регрессии используются для контролируемого обучения моделей ИИ – так называемого обучения «с учителем», когда данные размечаются для помощи в прогнозировании. Сопоставляя входные данные и полученные результаты на точность, модель постепенно обучается прогнозировать числовые значения целевых переменных.</td></tr>
<tr><td>Линейная регрессия</td><td>Прогнозирует целевую переменную Y на основе одной или нескольких независимых переменных X. Для этого между X и Y строится линейная связь.</td><td><table><tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr><tr><td>Журнал преобразований</td><td>–</td></tr></table></td><td>**</td></tr>
<tr><td>Дерево решений для регрессии</td><td>Предсказывает значение целевой переменной, изучая простые правила принятия решений, выведенные из характеристик данных. Представляет собой древовидный граф с узлами, где атрибут – вопрос, ребро – ответ на вопрос, а листья – фактический результат. Наблюдения</td><td><table><tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr><tr><td>Журнал преобразований</td><td>–</td></tr><tr><td>Глубина дерева</td><td>Заданное максимальное число разбиений в ветвях, по достижению которого обучение останавливается.</td></tr><tr><td>Оптимизация гиперпараметров</td><td>При выборе флага оптимизации не нужно вручную задавать глубину</td></tr></table></td><td>**</td></tr>
</table>

<table>
<tr>
<td></td>
<td>классифицируются сверху вниз от корня до листьев.</td>
<td>
<table>
<tr><td></td><td>дерева, или можно задать несколько значений на выбор. Алгоритм подбирает глубину дерева из расчета получить максимальное значение метрики.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Метод, который рассчитывает точность обученной модели. Выбирается один из предлагаемых методов.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Датасет делится на фолды – на указанное количество равных частей. При обучении модели регрессии каждый фолд становится валидационным один раз, при этом на остальных фолдах выполняется обучение. Каждый раз рассчитывается значение метрики. Затем рассчитывается усредненная метрика, которая характеризует точность модели.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Случайный лес для регрессии</td>
<td>В отличие от предыдущего алгоритма здесь строится ансамбль решающих деревьев. При этом большое количество некоррелированных моделей (деревьев)</td>
<td>
<table>
<tr><td>Датасет</td><td>Датасет с исходными данными.</td></tr>
<tr><td>Журнал преобразований</td><td>–</td></tr>
<tr><td>Глубина дерева</td><td>–</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>превосходит любую из отдельных моделей.</td>
<td>
<table>
<tr><td>Количество деревьев</td><td>–</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Алгоритм подбирает гиперпараметры: глубину и количество деревьев из расчета получить максимальное значение метрики.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Полиномиальная регрессия</td>
<td>Метод регрессионного анализа, в которой взаимосвязь между независимыми переменными x и зависимой переменной y моделируется как полином n-ой степени от x. Полиномиальная регрессия соответствует нелинейной зависимости между значением x и соответствующим</td>
<td>
<table>
<tr><td>Степень полинома</td><td>Степень уравнения полиномиальной регрессии, которая определяет линию наилучшего соответствия. При неправильном выборе степени, модель может быть перенасыщена. Значение по умолчанию – 2.</td></tr>
<tr><td>Только произведение</td><td>Если установить галочку в поле, то не выполняется возведение в степень, а только перемножение.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Нужно активировать галочку в поле, когда выбирается наиболее</td></tr>
</table>
</td>
<td>1. Модель полиномиальной регрессии.
2. Словарь с переменными для отображения в интерфейсе.
3. Словарь с преобразованиями данных.
4. Выходной датасет.</td>
</tr>
</table>

	условным средним *y*, обозначающим E(y|x). В отличие от линейной регрессии моделирует нелинейно разделенные данные – более гибкая и может моделировать сложные взаимосвязи.		подходящая степень полинома из нескольких предложенных. A подбирается гиперпараметр так, чтобы получить наилучшее значение метрики.	
		Метрика для оптимизации	Значения на выбор: RMSE, MAE, WMAPE, где RMSE – среднеквадратическая ошибка, MAE – средняя абсолютная ошибка, а ошибка – разница между значениями, предсказанными моделью, и фактическими значениями переменной. Эти метрики используются для оценки работы модели регрессии – проверяют точность прогноза и измеряют величину отклонения от фактических значений.	
		Количество фолдов для оптимизации	Указывается, на сколько равных частей разбивается входной датасет при обучении модели.	
Метод опорных векторов для регрессии	В основе регрессии опорных векторов (с англ. **SVR** – Support Vector Regression) лежит поиск гиперплоскости, при которой риск в	**Тип ядра**	Функция ядра (**kernel**) может принимать значения: {‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’}.	**
		Степень для	Если в качестве функции ядра	

	многомерном пространстве будет минимальным. SVR оценивает коэффициенты путем минимизации квадратичных потерь: считается сумма квадратов ошибок (между прогнозом и фактом), и к ней прибавляется штраф в виде произведения *коэффициента регуляризации* и суммы квадратов весов. *Вместо квадратичной функции используется кусочно-линейная, и задается отступ *eps* (по умолчанию, равная 0.1): Если разница между прогнозируемым и истинным значением меньше eps (прогнозное значение попадает в пространство гиперплоскости), модель не считает это за ошибку, иначе – берется модуль разницы.	**ядра полинома**	используется полиномиальная функция (‘poly’), которая является методом нелинейной регрессии, то зависимая переменная связана с независимыми переменными n-ой степени. В поле указывается степень этого ядра.	
		Коэффициент регуляризации	Мера степени наказания модели за каждую неверно спрогнозированную точку.	
		Оптимизация гиперпараметров	Флаг подбора гиперпараметров.	
		Метрика для оптимизации	Выбирается одна из предлагаемых метрик для оценки работы модели.	
		Количество фолдов для оптимизации	Указывается, на сколько равных частей разбивается входной датасет при обучении модели.	

<table>
<tr>
<td>Байесовская гребневая регрессия</td>
<td>В основе метода лежит формула Байеса, которая дает возможность оценить вероятность событий эмпирическим путем.
Гребневая регрессия – один из методов снижения размерности. Для гребневой регрессии к функции потерь прибавляется параметр lambda, обозначающий размер штрафа. Чем меньше lambda, тем выше *дисперсия* и ниже *смещение*.
Смещение – это погрешность оценки, возникающая в результате ошибочного предположения в алгоритме обучения. В результате большого смещения алгоритм может пропустить связь между признаками и выводом (недообучение).
Дисперсия – это ошибка чувствительности к</td>
<td>
<table>
<tr><td>alpha_1, alpha_2</td><td>Допустимые максимальные расстояния графика регрессии до верхнего и нижнего доверительного интервала.</td></tr>
<tr><td>lambda_1, lambda_2</td><td>Размеры штрафов при выходе прогнозируемых значений за пределы верхнего и нижнего доверительного интервала.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров.</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>–</td></tr>
</table>
</td>
<td>**</td>
</tr>
</table>

<table>
<tr><td></td><td>малым отклонениям в тренировочном наборе. При высокой дисперсии алгоритм может трактовать случайный шум в тренировочном наборе, а не желаемый результат (переобучение).</td><td></td><td></td></tr>
<tr><td>Метод k-ближайших соседей для регрессии</td><td>Для регрессии объекту присваивается среднее значение по k ближайшим к нему объектам, значения которых уже известны.
Алгоритм применяется к выборке с большим количеством атрибутов (многомерной). Для этого перед применением определяется функция расстояния, классический вариант такой функции – евклидова метрика.
Разные признаки могут иметь разный диапазон представленных значений в выборке, поэтому выполняется нормализация данных.</td><td>
<table>
<tr><td>Количество ближайших соседей</td><td>Число k, характеризующее количество соседей в кластере.</td></tr>
<tr><td>Тип веса для соседей</td><td>Задается одно из значений: ‘uniform’ (единый – всем признакам присваивается единый вес), или ‘distance’ (по расстоянию).
Значение по умолчанию – единый.</td></tr>
<tr><td>Метрика расстояния</td><td>Задается одно из значений: ‘chebyshev’ (Чебышева), ‘euclidean’ (Евклидова), ‘cosine’ (Косинусное), ‘cityblock’ (Манхэттенское).
Значение по умолчанию – евклидово расстояние, когда вычисляется расстояние между всеми точками попарно.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров.</td></tr>
</table>
</td><td>**</td></tr>
</table>

<table>
<tr><td></td><td>Некоторые значимые признаки могут быть важнее остальных, поэтому для каждого признака задается определенный вес.
Алгоритм предполагает, что похожие наблюдения существуют в непосредственной близости: улавливается идея сходства (иногда называемого расстоянием или близостью) благодаря вычислению Евклидова расстояния между точками.</td><td><table><tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr><tr><td>Количество фолдов для оптимизации</td><td>Указывается, на сколько равных частей разбивается входной датасет при обучении модели.</td></tr></table></td><td></td></tr>
<tr><td colspan="4">Авторегрессия – модель временного ряда, в которой ее текущее значение линейно зависит от предыдущих (ретроспективных) значений этого же ряда. Линейная зависимость означает, что текущее значение равно взвешенной сумме нескольких предыдущих значений ряда.
Зная параметры модели и соответствующие ретроспективные значения временного ряда, можно предсказать его будущие значения. Основное назначение авторегрессионной модели – прогнозирование. Также с ее помощью можно производить анализ временных рядов – выявлять тенденции, сезонность, и другие особенности.</td></tr>
<tr><td>ARIMA/ SARIMAX</td><td>Авторегрессионное интегрированное скользящее среднее (с англ. ARIMA – autoregressive integrated moving average) используется при работе с временными рядами для</td><td><table><tr><td>Число шагов для прогноза</td><td>Количество шагов, на которые модель будет предсказывать.</td></tr><tr><td>Порядок авторегрессии, p</td><td>Количество запаздывающих наблюдений, включенных в модель, также называется лаговый порядок. P</td></tr></table></td><td>**</td></tr>
</table>

	более глубокого понимания данных, или предсказания будущих точек ряда. Упоминается как **ARIMA (p, d, q)**, где *p, d* и *q* – целые неотрицательные числа, характеризующие порядок для частей модели (соответственно – авторегрессионной, интегрированной и скользящего среднего). **Авторегрессия**. Модель, использующая зависимую связь между наблюдением и некоторым количеством запаздывающих наблюдений. **Интегрированный**. Использование разности необработанных наблюдений (например, вычитание наблюдения из наблюдения на предыдущем временном шаге), чтобы сделать временной ряд стационарным.			

	помогает настроить линию для прогнозирования серии. Чисто авторегрессионные модели напоминают линейную регрессию, где прогностическими переменными являются p числа предыдущих периодов.
Порядок интегрировани я, d	Число обычных дифференцирований – количество раз, когда необработанные наблюдения различаются, также называется степенью различия. В модели ARIMA временные ряды преобразуются в *стационарные* (серии без тренда и сезонности), используя дифференцирование. Стационарный ряд – это когда среднее значение и дисперсия постоянны во времени.
Порядок скользящего среднего, q	Размер окна скользящей средней.
Параметры модели SARIMAX:	
Порядок авторегрессии	–
Порядок	Число сезонных дифференцирований.

<table>
<tr><td></td><td>Скользящая средняя. Модель, в которой используется зависимость между наблюдением и остаточной ошибкой из модели скользящего среднего, применяемая к запаздывающим наблюдениям.
Модель SARIMAX используется для временных рядов с учетом сезонности.</td><td><table>
<tr><td>интегрировани я</td><td></td></tr>
<tr><td>Порядок скользящего среднего</td><td>–</td></tr>
<tr><td>Сезонный период</td><td>Число наблюдений за сезон.</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Флаг подбора гиперпараметров.</td></tr>
</table></td><td></td></tr>
<tr><td colspan="4">Группа «Работа с текстами»</td></tr>
<tr><td>Автореферирование текста</td><td>Данная функция представляет собой автоматический процесс выделения краткого содержания текста с помощью модели машинного обучения. На выходе получается датасет заданного объема, который можно представить в виде таблицы.</td><td><table>
<tr><td>Объем автореферата</td><td>Максимальное количество символов в выходном результате</td></tr>
</table></td><td>Таблица с кратким содержанием</td></tr>
<tr><td colspan="4">Группа «Управление моделями»</td></tr>
</table>

Сохранение модели	Сохраняет модель по настроенному в системе пути и названию файла, а также сохраняет словарь с переменными. В этом словаре содержится отдельно список независимых переменных и список целевых признаков, с указанием выполненных над ними преобразований. Сохраняет шаг ресемплирования	**Название модели** — Пользователь задает название для обучаемой модели.	Созданная модель сохраняется в пункте меню системы **Модели –> Сохранённые модели**
Классификация			
Сохранение модели классификации изображений	Функция предназначена для сохранения в системе модели классификации изображений.	**Название модели** — Пользователь задает название для обучаемой модели.	**
Обнаружение объектов			
Сохранение модели YOLO v5	**** данный функционал находится в разработке, в текущей версии 2.3.3 применение функции недоступно.*** Функция предназначена для сохранения в системе модели распознавания изображений «YOLO v5».	**Название модели** — Пользователь задает название для обучаемой модели.	**

<table>
<tr><td></td><td>Сохранив модель, можно создать на ее основе приложение для последующей интеграции со сторонними системами. Также обученная модель может использоваться повторно для анализа онлайн данных.</td><td></td><td></td></tr>
<tr><td colspan="4">Spark</td></tr>
<tr><td>Сохранение модели Spark</td><td>Функция предназначена для сохранения в системе моделей, собранных с применением технологии Spark.</td><td><table><tr><td>Название модели</td><td>Пользователь задает название для обучаемой модели.</td></tr></table></td><td></td></tr>
<tr><td colspan="4">Группа «Глубокое обучение»</td></tr>
<tr><td>Валидация модели классификации изображений</td><td>После того, как модель нейронной сети обучена, натренирована и для нее выбраны оптимальные гиперпараметры, необходимо проверить ее точность и адекватность. Для этого выполняется валидация итоговой модели нейронной сети на тестовой выборке.</td><td><table><tr><td>Метрика</td><td>Метрика, которая оценивает работу обученной модели нейронной сети. Применяется к типу данных – изображения. Чтобы оценить качество модели классификации используются следующие метрики:
1. Accuracy – оценивает долю правильных ответов модели.</td></tr></table></td><td>1.Строится матрица ошибок. Пример для бинарной классификации:<table><tr><td></td><td colspan="2">Истинный класс</td></tr><tr><td></td><td>0</td><td>1</td></tr><tr><td>0</td><td>TN</td><td>FN</td></tr><tr><td>1</td><td>FP</td><td>TP</td></tr></table></td></tr>
</table>

	В качестве *входных данных* для функции используются: -тестовый датасет с изображениями; - обученная модель; -словарь с преобразованиями данных; -выбранная метрика валидации. Рассчитывается сколько изображений тестовой выборки попадают в каждую ячейку матрицы ошибок. Оценивается качество классификации.			2. **F1** (среднее гармоническое) – агрегированная функция, которая позволяет вместо точности и полноты использовать только один параметр качества классификации. Формула: $$F_1 = \frac{2 \cdot precision \cdot recall}{precision + recall}$$ Чем ближе F1 к 1, тем лучше. *В задачах, в которых точность и полнота не равноценны, применяется взвешенное значение F_β. 3. **Precision**. Метрика, которая оценивает *точность модели*. Рассчитывается по формуле: $$precision = \frac{TP}{TP + FP},$$ здесь считается точность для класса 1 (для класса 0 считается аналогично). 4. **Recall** – оценивает *полноту модели*: $$recall = \frac{TP}{TP + FN}.$$ Идеально, чтобы точность и полнота были равны 1 (100%). 5. **AUC_ROC**. Для анализа качества модели применяется ROC-анализ: строится ROC-кривая, которая наиболее часто используется для представления результатов	где столбцы – истинные классы, а строки – предсказанные классы. Обозначения: TN – true negative TP – true positive FN – false negative FP – false positive Например, ячейка **TP** означает, что объект действительно принадлежит классу 1, и для него предсказан класс 1. А **FN** – объект неправильно отнесли к классу 0, хотя он принадлежит к классу 1. 2.Отображается значение выбранной метрики.

			бинарной классификации. Классов два: один называется классом с положительными исходами, второй – с отрицательными исходами. ROC-кривая показывает зависимость количества верно классифицированных положительных примеров от количества неверно классифицированных отрицательных примеров. Чем выше показатель AUC_ROC, тем качественнее классификатор, при этом значение 0,5 демонстрирует непригодность выбранного метода классификации. AUC_ROC = 0,9-1,0 означает отличное качество модели.	
Классификация				
Классификация (табличные данные)	Типы решаемых задач: анализ тональности, классификация текста по категориям, распознавание речи и многое другое. Рассмотрим функцию на примере задачи по отнесению документов к определенной категории на основании его содежимого. Процесс			**

Количество эпох	Параметр, который показывает сколько раз *модель* подвергается воздействию обучения.
Размер мини-батча	Количество обучающих примеров за одну итерацию. Под примерами имеются в виду *наблюдения* – строки в табличных данных.

	классификации осуществляется с помощью применения методов машинного обучения, в частности *сверточных нейронных сетей*. Задача классификации текстов применима в решении следующих задач: борьба с массовой рассылкой рекламы, распознавание тональности текстов, сортировка документов и т.д. Задача определяется следующим образом: пусть существует конечное множество категорий, на вход алгоритма подается конечное количество документов, и есть целевая функция, которая определяет соответствие для каждой пары (документ, категория). Задача состоит в нахождении этой			

Метрика для обучения	Метрика «Accuracy» (точность) показывает долю правильных ответов алгоритма
Алгоритм градиентного спуска	Метод нахождения минимального значения *функции потерь*. Алгоритмы на выбор: *SGD*, *Adam*, и др.
Шаг градиентного спуска	Параметр, регулирующий скорость обучения модели. Значения – 0.001 или 0.1.
Функция потерь	Выбирается одна из функций, в зависимости от задачи: бинарная классификация или многоклассовая.
Добавить слой	Задается одно из трех значений: Conv2D, Flatten, Dense (последний – полносвязный слой).
Перемешивать выборку перед обучением	Необходимо установить галочку в поле, чтобы случайным образом поменять местами наблюдения, соответствующие строкам в таблице
Порог классификации	Для бинарной классификации значение по умолчанию 0.5, для

<table>
<tr>
<td></td>
<td>функции, называемой классификатором.
Строится многослойная нейронная сеть, состоящая из слоев:
-входной, на который поступают входные признаки;
-скрытый, на котором рассчитываются промежуточные результаты;
- выходной, на котором выводятся окончательные значения, вычисленные по гипотезе.
*Сверточными искусственные нейронные сети называются из-за специальной архитектуры, с наличием операций сверки.</td>
<td>
<table>
<tr><td></td><td>многоклассовой – параметр не заполняется</td></tr>
<tr><td>Флаг возврата вероятности при прогнозе</td><td>Возвращает вероятность или метки классов для дальнейшего прогноза после обучения модели. Для бинарной классификации может использоваться одно поле с метками 0 и 1, обозначающими принадлежность к тому или иному классу. Для многоклассовой классификации используется несколько полей – каждое поле соответствует отдельному классу (0, 1, 2 и т.д.), и записываются вероятности (от 0 до 1), с которыми наблюдения принадлежат классам</td></tr>
<tr><td>Оптимизация гиперпараметров</td><td>Алгоритм подбирает гиперпараметры</td></tr>
<tr><td>Метрика для оптимизации</td><td>Выбирается одна из предлагаемых метрик для оценки работы модели.</td></tr>
<tr><td>Количество фолдов для оптимизации</td><td>–</td></tr>
</table>
</td>
<td></td>
</tr>
<tr>
<td>Классификация изображений</td>
<td>Алгоритм решения задачи классификации:</td>
<td></td>
<td>**</td>
</tr>
</table>

	1.Берется *тренировочная выборка* – набор изображений с известными значениями целевого признака Y. Нейронная сеть должна восстановить зависимость между нецелевыми признаками и целевым. 2.Задаются основные параметры нейронной сети. 3.Выписываются выражения для вероятностей принадлежности наблюдения к тому или иному классу (Y = 0, 1, 2, 3, и т.д.). 4.По тренировочной выборке составляется функция потерь. 5. Функция потерь L(w) содержит вхождения весов нейронной сети. Относительно этих переменных находится *точка минимума функции L(w)*.		**Количество эпох**	Это гиперпараметр, который определяет сколько раз *алгоритм обучения* будет обрабатывать весь *набор обучающих данных*. То есть *эпоха* – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества		
			Метрика для обучения	Метрика «Accuracy» (точность) показывает долю правильных ответов алгоритма		
			Алгоритм градиентного спуска	Метод нахождения минимального значения *функции потерь*. Минимизация любой функции означает поиск самой глубокой впадины в этой функции. Функция используется, чтобы контролировать ошибку в прогнозах модели. Поиск минимума означает получение наименьшей возможности ошибки или повышение точности модели. Точность увеличивается перебором *учебных* данных при настройке параметров модели (весов и смещений). Суть алгоритма – процесс получения наименьшего значения ошибки. Аналогично это можно рассматривать		

	6.Точка минимума определяет оптимальные веса нейронной сети. 7.Весам нейронной сети присваиваются найденные оптимальные значения. Пусть изображение А не принадлежит тренировочной выборке. Объект А прогоняется через нейронную сеть и на выходе получаются вероятности – они и являются предсказаниями для объекта А (по максимальной вероятности определяется принадлежность объекта к классу).			как спуск во впадину в попытке найти самое низкое значение ошибки. Можно выбрать один из следующих алгоритмов: *SGD* (Stochastic gradient descent, с англ. Стохастический градиентный спуск), *Adam*, и др. Алгоритм **Adam** является *модифицированным*, в нем также выполняется минимизация функции потерь. Рассчитываются векторы *частных* в текущей точке функции, и определяются координаты следующей точки. Частные производные вычисляются, чтобы определить, какой был вклад в ошибку по каждому весу.		
			Шаг градиентного спуска	Параметр, регулирующий скорость обучения модели – насколько быстро функция потерь спускается к своему минимуму (скорость спуска/поиска). Выбирается значение: 0.001, или 0.1.		
			Функция потерь	Функция потерь находится в центре нейронной сети. Она используется для расчета ошибки между *реальными* и *полученными* ответами. Главная цель – минимизировать эту ошибку. Или: **максимизировать вероятность принадлежности к истинному**		

				классу для каждого объекта из тренировочной выборки. Она также может зависеть от таких переменных, как веса и смещения, где *смещения* – это веса, добавленные к скрытым слоям. Выбирается одна из функций, в зависимости от задачи: бинарная классификация или многоклассовая.		
			Добавить слой	Основой алгоритмов распознавания изображений являются сверточные нейронные сети. Для их построения используются три главных типа слоев: сверточный слой, слой подвыборки и полносвязный слой. Соответственно пользователь задает одно из трех значений: Conv2D, Flatten, Dense (последний – полносвязный). В сверточных нейронных сетях одно изображение является одним наблюдением. Таким образом, исходное изображение преобразуется, слой за слоем, от начального значения пикселя до итоговой оценки класса. Слои, идущие до полносвязного, являются средствами предобработки изображения, и используются для выделения различных признаков,		

<table>
<tr><td></td><td></td><td><table>
<tr><td></td><td>которые затем подаются на вход классификатору.</td></tr>
<tr><td>Порог классификации</td><td>Важно! Параметр заполняется только для задачи бинарной классификации, значение по умолчанию 0.5. Для множественной классификации это поле остается пустым.</td></tr>
</table></td><td></td></tr>
<tr><td colspan="4">Регрессия
При обучении алгоритма регрессии на платформе существует возможность использовать распределенные вычисления Spark CPU.</td></tr>
<tr><td>Регрессия (табличные данные)</td><td>Для обучения нейронной сети данные делятся на части меньшего размера, загружают их по очереди и обновляют веса нейросети в конце каждого шага, подстраивая их под данные.</td><td><table>
<tr><td>Количество эпох</td><td>Указывается количество эпох для обучения модели.
Одна эпоха – весь датасет прошел через нейронную сеть в прямом и обратном направлении только один раз. Так как одна эпоха слишком велика для компьютера, датасет делят на партии – батчи.
С увеличением числа эпох, веса нейронной сети изменяются все большее количество раз. Кривая с каждым разом лучше подстраивается под данные, переходя последовательно из плохо обученного состояния в оптимальное. Если вовремя не остановиться, то может произойти переобучение.</td></tr>
</table></td><td>**</td></tr>
</table>

			Размер мини-батча	Общее число тренировочных объектов, представленных в одном батче. Нельзя пропустить через нейронную сеть разом весь датасет. Поэтому делим данные на пакеты, сеты или партии. *Итерации* – это число батчей, необходимое для завершения одной эпохи.		
			Метрика для обучения	Для регрессии: ['MAE', 'MAPE', 'MSE'].		
			Алгоритм градиентного спуска	Алгоритм итеративной оптимизации, используемой в машинном обучении для получения более точного результата. *Градиент* показывает скорость убывания или возрастания функции. *Спуск* говорит о том, что мы имеем дело с убыванием. Алгоритм **итеративный**, процедура проводится несколько раз, чтобы добиться оптимального результата. На каждом шаге результат получается лучше.		
			Шаг градиентного спуска	Скорость обучения модели алгоритмом градиентного спуска.		

			Функция потерь	Функция, которая используется для оптимизации алгоритма машинного обучения. Значение, вычисленное такой функцией, называется ‘потерей’. Потери регрессии рассчитываются путем прямого сравнения выходного и истинного значения. Самая популярная функция для регрессионных моделей – это среднеквадратическая ошибка, MSE. Функция потерь определяет, как именно выходные данные связаны с исходными. По сути вычисляется насколько хорошо работает модель – сравнивается то, что модель прогнозирует, с фактическим значением. Сохраняется функция потерь, которая может эффективно наказывать модель, пока та обучается на тренировочных данных.		
			Добавить слой	Из списка выбирается дополнительный слой для нейросети.		
			Перемешивать выборку перед обучением	Необходимо установить галочку в поле, чтобы случайным образом поменять местами наблюдения.		
			Оптимизация	Если установить галочку в поле, то		

		гиперпараметров	алгоритм выберет наилучшие гиперпараметры для создания модели из списка предложенных	
		Метрика для оптимизации	Выбирается одна из предлагаемых метрик для оценки работы модели.	
		Количество фолдов для оптимизации	Указывается, на сколько равных частей разбивается входной датасет при обучении модели.	
Обнаружение объектов				
YOLOv5	**** данный функционал находится в разработке, в текущей версии 2.3.3 применение функции недоступно.*** Датасет должен быть разделен на две папки: *train* (тренировочная выборка) и *val* (валидационная). В каждой папке лежат еще две папки: *images* (изображения) и *labels* – папка с текстовыми файлами, содержащими метки объектов на этих изображениях в формате YOLO.	**Размер мини-батча**	Указывается количество изображений, которое одновременно подается на вход YOLO. Например, если задать размер 2, за один подход подается два изображения.	В результате получаются изображения с обозначением детектированных объектов и значениями *confidence*. Где confidence – число от 0 до 1, характеризующее ‘уверенность’ модели в том, что детектирован объект или детектирован объект определенного класса. Еще один параметр *conf-thres* позволяет установить пороговое значение для confidence модели. Все объекты, confidence которых ниже этого значения не считаются объектами.
		Количество эпох	Это гиперпараметр, который определяет сколько раз *алгоритм обучения* будет обрабатывать весь *набор обучающих данных*. То есть *эпоха* – одна итерация в процессе обучения, включающая предъявление всех примеров из обучающего множества. Например, для выборки в десять изображений и размера мини-	

	Для этой функции предварительно выполняется разметка изображений на тренировочной и валидационной выборках. Пользователь с помощью ‘bounding box’ отмечает объекты на изображениях. А алгоритм по точкам объектов находит функцию их обнаружения. На валидационной выборке проверяется точность обученной модели. После запуска функции на выходе получается обученная нейронная сеть, результаты обучения которой сохраняются в БД.		батча два, эпоха равна пяти прохождениям.	Также отображаются: описание модели, графики обучения модели, матрица ошибок на валидационных данных. Где описание модели содержит информацию об оптимизаторе, тренировочном и валидационном датасетах, а также параметры обучения.
Отправка уведомлений				
Отправка уведомлений	Функция предназначена для осуществления отправки уведомлений в настроенный канал телеграм или по	**Канал уведомлений**	Выбирается ранее настроенный в разделе Администрирование -> Уведомления канал, на который будет осуществляться	Оповещение в телеграм канал

<table>
<tr><td></td><td>электронной почте (не реализовано в текущей версии). Данная функция выступает в паре с блоком шлюз, где необходимо задать условия, при которых будет отправлено уведомление.</td><td><table><tr><td></td><td>отправка сообщений.</td></tr></table></td><td></td></tr>
<tr><td colspan="4">Spark. Группа функций для фреймворка Apache Spark. Названия функций дублируются с теми, что были описаны ранее, разница заключается в использовании другого модуля машинного обучения в Apache Spark (в следующих версиях Платформы планируется сделать все функции универсальными).</td></tr>
<tr><td>Сохранение датасета Spark в CSV</td><td>Функция распределяет входные данные в несколько файлов в одну директорию. Для этого выбираем: куда сохранить данные, как их назвать, и по необходимости можем подгрузить новую порцию данных.</td><td><table><tr><td>Путь до директории для датасета</td><td>Выбирается путь до папки, в которую будут сохраняться данные.</td></tr><tr><td>Название датасета</td><td>В этом поле задается название для датасета. По умолчанию датасеты создаются с названиями формата pySpark.csv.</td></tr><tr><td>Добавить данные к датасету</td><td>Если преобразованные данные необходимо сохранять не в виде отдельного файла, а добавить к уже существующему и загруженному на платформе, необходимо установить галочку у данного признака. По умолчанию файл перезаписывается.</td></tr><tr><td>Название</td><td>Указывается название, с которым</td></tr></table></td><td>Таблица в формате csv с датасетом. Сохраняется в раздел данные</td></tr>
</table>

		датасета для валидации	будет сохранен датасет для валидации при активации параметра «Сохранить датасет для валидации»	
		Сохранить датасет для валидации	В процессе работы пайплайна, исходный вид набора данных теряется, поэтому его нужно передать из блока "Загрузка данных" в конец пайплайна в блок сохранени. Датасет для валидации это и есть нетронутый набор данных в первоначальном виде, к нему только добавляется столбец с результатами.	
		Загрузка датасета для валидации в БД	Позволяет загрузить датасет для вализации напрямую в базу данных ClickHouse	
Косинусное расстояние	На вход функция получает новые данные для анализа (датасет в формате csv), обученную модель, и числовой вектор. Выполняется поиск объектов, наиболее схожих с заданным вектором, и в качестве	–		Таблица «Косинусное расстояние»

<table>
<tr><td></td><td>меры схожести используется косинусное расстояние - расстояние между значениями во входном векторе и значениями выбранных столбцов в наблюдениях.</td><td></td><td></td></tr>
<tr><td>Выбор признаков и целевых признаков</td><td>Аналогично стандартной функции.</td><td></td><td></td></tr>
<tr><td>Разделение датасета на обучающую и тестовую выборки</td><td>Аналогично стандартной функции.</td><td></td><td></td></tr>
<tr><td>Валидация модели</td><td>Аналогично стандартной функции.</td><td></td><td></td></tr>
<tr><td>Прогноз модели</td><td>Аналогично стандартной функции.</td><td></td><td></td></tr>
<tr><td>Порядковое кодирование признаков</td><td>Порядковое кодирование - это метод преобразования категориальных данных в цифровой вид. Применяются, когда в датасете существуют НЕ числовые признаки, которые заданы словами и для дальнейшего анализа</td><td><table><tr><td>Выбранные признаки</td><td>Указываются признаки, над которыми необходимо провести операцию порядкового кодирования.</td></tr></table></td><td>-</td></tr>
</table>

	их нужно преобразовать в числа. Порядковое кодирование позволяет пронумеровать признаки по порядку.		
Нормализация признаков	Нормализация - это приведение числовых признаков к единой шкале. Бывает, что числовой признак имеет минимальное и максимального значение в очень широком диапазоне и это плохо для машинного обучения. Например, есть числовой признак, чье минимальное значение равно 0,001, а максимальное - 100000, нормализация преобразовывает их к диапазону от 0 до 1, то есть 0.001 становится 0, а 100000 становится 1, значения между ними также преобразуются, 50 000 станет примерно равным 0.5. Данная функция позволяет	–	–

<table>
<tr><td></td><td>оптимизировать дальнейшие вычисления.</td><td></td><td></td></tr>
<tr><td>Модель градиентного бустинга Spark для бинарной классификации</td><td>Градиентный бустинг представляет собой ансамбль деревьев решений. В основе данного алгоритма лежит итеративное обучение деревьев решений с целью минимизировать функцию потерь. Благодаря особенностям деревьев решений градиентный бустинг способен работать с категориальными признаками, справляться с нелинейностями. Бустинг – это метод преобразования слабообученных моделей в хорошообученные. В бустинге каждое новое дерево обучается на модифицированной версии исходного датасета.</td><td><table><tr><td>Количество базовых моделей</td><td>Указывается число последовательных итераций по оптимизации модели, то есть количество попыток уменьшить ошибку.</td></tr></table></td><td>–</td></tr>
<tr><td>Кластеризация Spark DBSCAN</td><td>Алгоритм DBScan формирует группы коренных</td><td><table><tr><td>Порог для</td><td>Указывается граница, которая</td></tr></table></td><td></td></tr>
</table>

	соседей/кластеры, объединяя точки, расположенные рядом. А точки, которые не попадают ни в одну из групп, отмечаются меткой -1 и приравниваются к аномалиям.		**отнесения кластера к аномалиям**	определяет, когда нужно отнести кластер к аномалии. ество попыток уменьшить ошибку.	
			Радиус	Радиус в единицах расстояния, в рамках которого выполняется поиск потенциальных соседей (float/list/tuple).	
			Число соседей	Минимальное число ближайших соседей в указанном радиусе для формирования группы коренных соседей (int/list/tuple).	
			Метрика расстояния	Метрика расстояния (str/list): расстояние Евклида, косинусное расстояние. По умолчанию «Евклидово расстояние» – используется при кластеризации данных в текущем датасете, а также при отнесении нового объекта к кластеру.	
			Флаг векторизации признаков	Параметр определяет будет ли проводиться векторизация или нет	
			Столбец для группировки перед	Указывается номер столбца	

		векторизацией	
Дополнительные функции			
Расчет параметров графа	С помощью данной функции осуществляется расчет количества вершин и ребер в выбранном графе.		Таблица «Количество вершин и ребер графа»
Поиск ближайших вершин	С помощью данной функции осуществляется определение id вершин, ближайших к указанным координатам, и расстояний между ними. Может быть вычислено как для одной точки, так и для нескольких точек.	**Введите координаты долготы**: Координата долготы точки, для которой производится поиск ближайшей вершины в графе: восточная долгота до 180, западная долгота до -180. **Введите координаты широты**: Координата широты точки, для которой производится поиск ближайшей вершины в графе: северная широта до +90, южная широта до -90.	Таблица «Ближайшие вершины»
Вычисление k кратчайших путей в графе	Определение k кратчайших путей между парой заданных пользователем вершин в графе	**Количество базовых моделей**: Указывается число последовательных итераций по оптимизации модели, то есть количество попыток уменьшить	Изображение «Кратчайшие пути в графе»

			ошибку.		
		Введите номер точки отправления	Планируемый номер вершины отправления в графе — индекс вершины графа		
		Введите номер точки прибытия	Планируемый номер вершины прибытия в графе — индекс вершины графа		

Приложение 2. Содержимое файла с погодными условиями (пример табличных данных для подачи в блок-схему)

datetime	area	T	P	U	Ff	Td	RRR	DD_Ветер, дующий с востока	DD_Ветер, дующий с востоко-северо-востока	DD_Ветер, дующий с востоко-юго-востока	DD_Ветер, дующий с запада	DD_Ветер, дующий с западо-северо-запада
2018-02-01	0.0	2.8625000000000003	767.675	72.625	2.0	-1.925	0.0	0.0	0.0	0.0	0.25	0.0
2018-02-02	0.0	5.75	765.4625	73.375	2.375	1.0	0.0	0.0	0.0	0.375	0.0	0.0

DD_Ветер, дующий с западо-юго-запада	DD_Ветер, дующий с севера	DD_Ветер, дующий с северо-востока	DD_Ветер, дующий с северо-запада	DD_Ветер, дующий с северо-северо-востока	DD_Ветер, дующий с северо-северо-запада	DD_Ветер, дующий с юга	DD_Ветер, дующий с юго-востока	DD_Ветер, дующий с юго-запада	DD_Ветер, дующий с юго-юго-востока	DD_Ветер, дующий с юго-юго-запада
0.375	0.0	0.0	0.0	0.0	0.0	0.125	0.125	0.125	0.0	0.0
0.0	0.0	0.0	0.0	0.0	0.0	0.125	0.0	0.0	0.125	0.375

DD_Штиль, безветрие	N_10% или менее, но не 0	N_100%.	N_20–30%.	N_40%.	N_60%.	N_70 – 80%.	N_90 или более, но не 100%	N_Небо не видно из-за тумана и/или других метеорологических явлений.	N_Облаков нет.	day	month	thermo_area
0.0	0.0	0.125	0.375	0.125	0.125	0.25	0.0	0.0	0.0	1	2	0.0
0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.125	0.0	0.375	2	2	0.0

Таблица 18.3 – Обучение модели классификации изображений

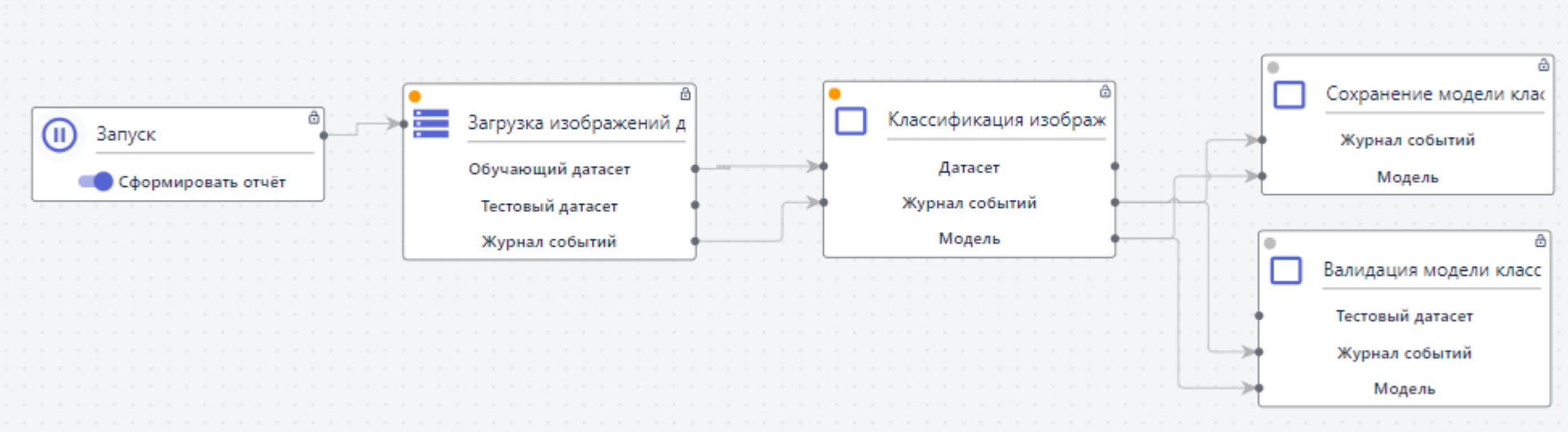


Таблица 18.4 – Обучение модели прогнозирования температуры воды и газов в котле

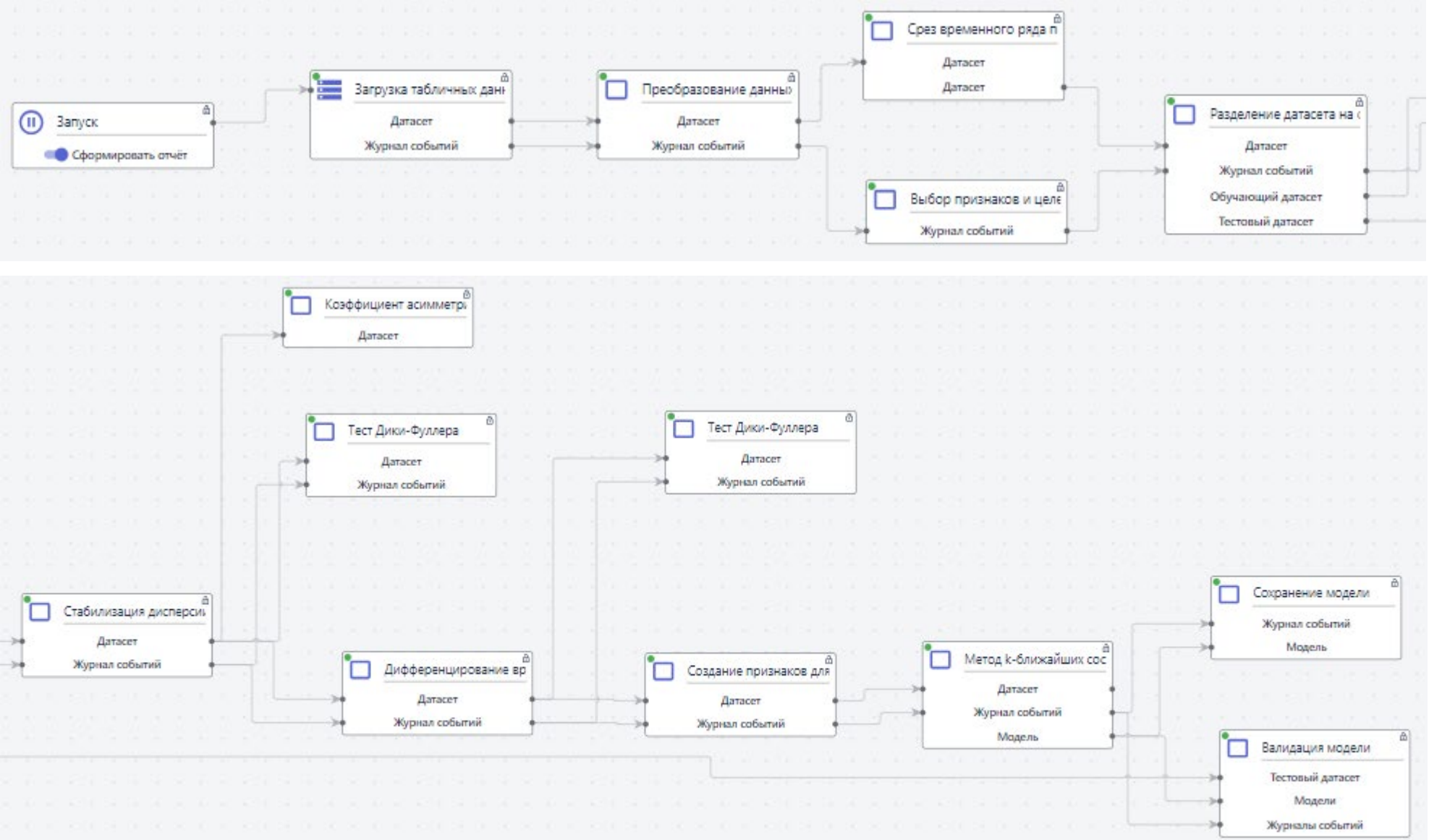


Таблица 18.5 – Пайплайн, обрабатывающий и прогнозирующие данные в режиме реального времени

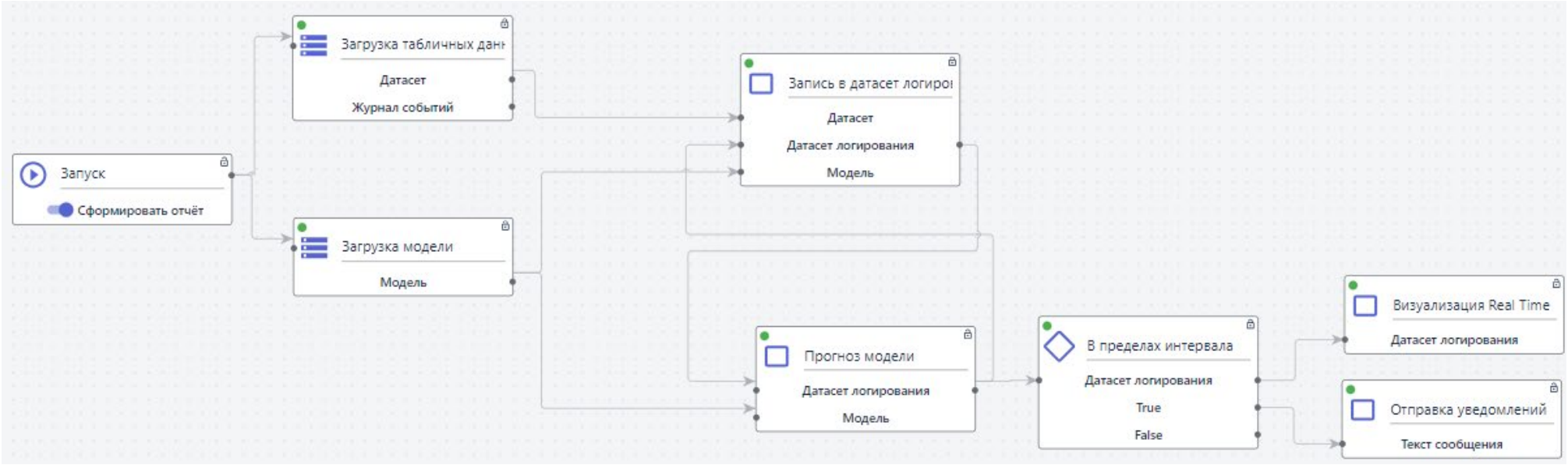


Таблица 18.6 – Обучение модели классификации текстов

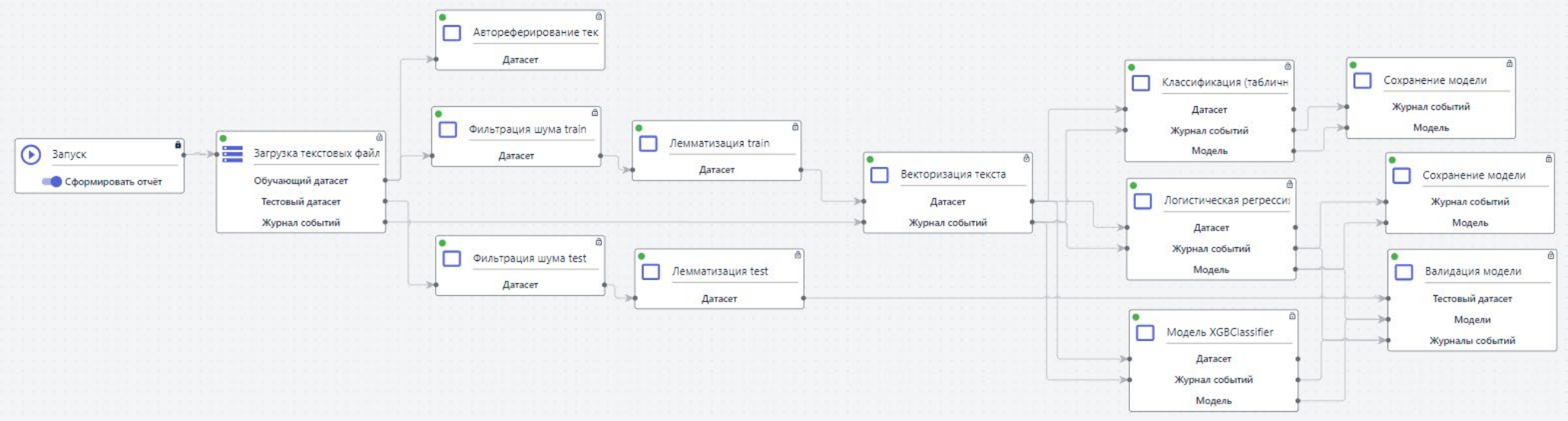
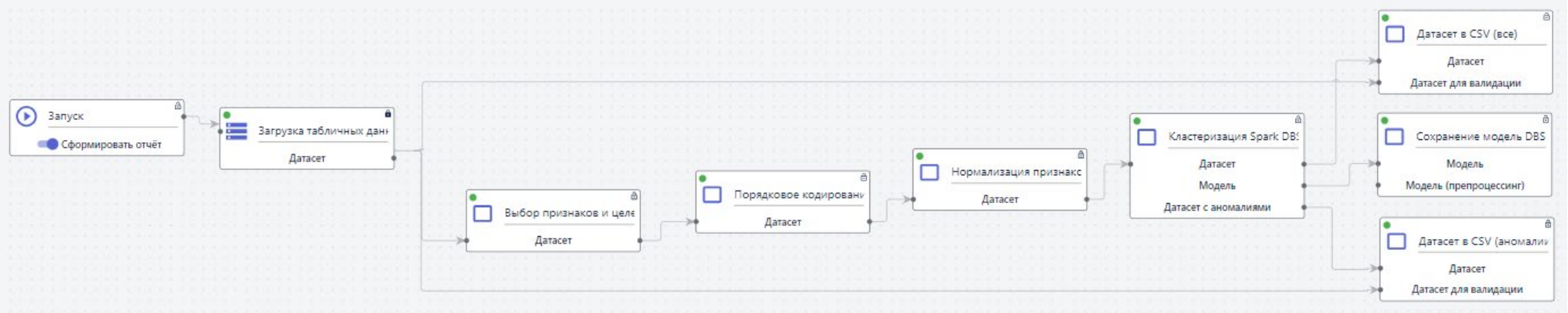


Таблица 18.7 – Обучение модели классификации Spark



Лист изменений

Таблица 19.1 – Лист изменений в версии Платформы 2.3.7

Наименование раздела	Содержание изменения	Обоснование
Раздел 15. Примеры работы с платформой	К примерам работы с платформой добавлено описание функции сегментации изображений	Новая информация
Раздел 15. Примеры работы с платформой	К примерам работы с платформой добавлено описание функции стекинга для классификации	Новая информация
Раздел 15. Примеры работы с платформой	К примерам работы с платформой добавлено описание функции стекинга для регрессии	Новая информация
Раздел 5.4 Встроенные функции	Обновлено описание блока “Процесс”	Актуализация информации

Таблица 19.2 – Лист изменений в версии Платформы 2.3.6

Наименование раздела	Содержание изменения	Обоснование
Раздел 4. Личный кабинет	В описание работы с личным кабинетом добавлено описание функций по установке аватара пользователя и изменению параметров авторизации	Актуализация информации
Приложение 1.	В описание регрессии добавлена информация о поддержке использования распределенных вычислений Spark CPU.	
Раздел 15.3.3. Классификация родинок	Обновлен раздел “Классификация изображений”. Добавлен подраздел 15.3.3. Классификация родинок	Актуализация информации
Раздел 13.4	Добавлен раздел 13.4. Автоматическая сборка и тестирование проектов	Новая информация

Таблица 19.3 – Лист изменений в версии Платформы 2.3.5

Наименование раздела	Содержание изменения	Обоснование
Раздел 5.4.2 Функции элемента «Процесс»	В подгруппу “Классификация” добавлена функция “Логический анализ данных”.	Актуализация информации
Раздел 15.5 Кластеризация Spark	Добавлено описание графика «Сформированные кластеры»	Актуализация информации
Раздел 5.4. Встроенные функции	В подгруппу “Классификация” добавлена функция “Поиск и удаление выбросов”.	Актуализация информации

Таблица 19.4 – Лист изменений в версии Платформы 2.3.4

Наименование раздела	Содержание изменения	Обоснование
Раздел 15. Примеры работы с платформой	Добавлен пример работы для классификации текстовых данных с использованием слоя нейронной сети LSTM	Актуализация информации
Раздел 15. Примеры работы с платформой	Добавлен пример работы для извлечения текстового слоя из текстовых данных	Актуализация информации
Раздел 15. Примеры работы с платформой	Добавлен пример работы для заполнения и работы с пропусками в табличных данных	Актуализация информации