RAZUM AI

Техническое описание

СОДЕРЖАНИЕ

1. Введение

Программное обеспечение RAZUM AI - это платформа (далее – платформа, платформа RAZUM AI), предназначенная для обработки структурированных и неструктурированных массивов данных и обучения моделей искусственного интеллекта для задач создания баз знаний, предиктивной аналитики в промышленности и медицине.

Решение задач с использованием платформы не требует кодирования, специалисту достаточно создать блок схему с последовательностью преобразований над исходными данными, запустить её и получить результат. При этом платформа RAZUM AI предоставляет широкий спектр возможных визуализаций результатов: различные графики, диаграммы, таблицы, изображения. Благодаря визуализации специалист может легко проанализировать процесс, а также представить результаты наглядно.

Платформа RAZUM AI ориентирована на пользователя: начиная с понятного и дружелюбного интерфейса, заканчивая большим списком возможных функций, применяемых для обработки данных: анализ данных (препроцессинг, предобработка, различные тесты и т.д.), машинное обучение (прогноз, классификация, кластеризация и т.д.), глубокое обучение (классификация, регрессия, распознавание объектов и т.д.), также платформа поддерживает ленивые вычисления с помощью Spark.

Платформа RAZUM AI, благодаря своей гибкости, эффективности и ориентированности на клиента, способна решать большой спектр задач, позволяя быстро и наглядно создавать блок схемы для анализа и прогнозирования различных данных, визуализировать результаты и применять обученные модели для дальнейших процессов работы с данными.

Все больше заказчиков из различных сфер задумываются о применении методов искусственного интеллекта (ИИ) для своих задач. Естественно, что любой заказчик перед лицом такой прорывающей инновации хочет максимально подстраховаться и по возможности снизить капитальные инвестиции, «запуститься» как можно быстрее и ожидает адекватного жизненного цикла решения как минимум в течение 3-5 лет.

Компания проанализировала пожелания своих заказчиков и предлагает оптимальное решение, в основу которого были заложены следующие парадигмы:

- Для любой технологии ИИ требуются не просто «большие», а уже «сверхбольшие» наборы данных, их необходимо хранить в течение многих лет (как де-факто для накопления статистики, так и де-юре для выполнения различных законодательных актов);
- Требуется гибкий баланс между масштабированием объема хранимых данных (до 100 и 1000 раз за 2-3-5 лет), возможностью быстро (порой в реальном времени) проводить на этих массивах данных процедуры нормализации и пр. этапы пред- и пост-обработки, необходимостью обеспечивать загрузку ядра данных и моделей в систему ИИ в «реальном времени»;
- Все это должно одинаково оптимально поддерживать как структурированные данные (включая объектное хранение), так и неструктурированные (выборки из соцсетей и т.п. источников);
- Обязательна поддержка различных высокопроизводительных специализированных процессорных платформ (поскольку заранее не всегда можно предсказать все задачи заказчика на несколько лет вперед и, тем более, прогресс аппаратных решений);

- «Эластичность» производительности по отношению к разным задачам – возможность при необходимости отдавать нужную мощность на процедуры хранения/дедупликации/предобработки/загрузки данных либо непосредственно на сами задачи ИИ – обладать разумным резервом мощности, но без неиспользуемых излишков;
- Не просто «фреймворк» для работы, но наполненный заранее разработанными (а в ряде случаев уже настроенными и предобученными) моделями обработки данных и собственно программных модулей ИИ (машинного обучения и нейронных сетей);
- Не только ПО для хранения данных и обучения нейросетей, но и инструменты (веб-сервисы) для очистки, разметки и прочей подготовки обучающих данных (датасетов) для использования при обучении нейронных сетей и других алгоритмов машинного обучения;
- Обеспечение услуг по разработке и сопровождению «под ключ», начиная от сайзинга аппаратной платформы до выбора алгоритмов, консультаций по формированию датасетов и обучению алгоритмов, сбору и разметке датасетов;
- Открытость решения, в котором заказчик волен как воспользоваться услугами внешних консультантов, так и получить исчерпывающую документацию и все делать самостоятельно;
- Понятная заказчику (возможность выбрать только нужные аппаратные блоки и программные модули) и конкурентоспособная ценовая политика, не в ущерб надежности и производительности;
- Поддержка горячих клавиш при работе с объектами на рабочей области.

2. Типы обрабатываемых данных с применением RAZUM AI

Анализ изображения/видео

Классификация изображений	Поиск лиц царской семьи на дореволюционных фото
	Распознавание и классификация лиц на дореволюционных фото. Точность классификатора 85-90%
Диагностика заболеваний в медицине	Распознавание злокачественных родинок
	Создание нейронной сети для автоматической диагностики родинок на злокачественность. Пациент имеет возможность отправить фото своей родинки и получить оценку онлайн, далее оформить запись на прием. Точность модели нейронной сети до 93,21%
Промышленность	Распознавание дефектов на металлических изделиях
	Раннее распознавание микродефектов на металлических поверхностях, с помощью нейронных сетей. Точность классификатора больше 90%
Снижение количества ошибок на производстве	Фиксирование отломанных зубьев на производственной линии
	Анализ входящего видеопотока через API. При фиксировании дефекта изделия отправка уведомления инженеру

Анализ текста

Классификация текстов по автору	Определение автора по тексту, с помощью рекуррентных сетей и одномерной свертки. Высокая точность распознавания текста
Кластеризация текстов	Определение аномалий в тексте, выявление и распознавание кластеров в текстах
Классификация текстов	Извлечение текстового слоя из текстовых данных
Определение ключевых слов	Определение ключевых слов в текстовых кластерах

Анализ числовых данных/временных рядов

Предиктивная аналитика показателей систем СХД	Предиктивно – статистический анализ временных рядов. Раннее предсказание показателей нагрузки на центральный процессор, чтения дисков/ задержки записи
AI мониторинг и предиктивная аналитика для ТЭЦ	Считывание фактических показателей оборудования и комплексный мониторинг состояния системы в реальном времени
	Интеллектуальная система принятия решений на основе глубокой аналитики и машинного обучения

	Решаемые задачи – Уменьшение времени планового простоя, предиктивное предупреждение выхода из строя агрегатов, проведение только фактически необходимого технического обслуживания, оптимизация работы системы
	Ожидаемый срок окупаемости от 3 до 7 лет в зависимости от типа системы и специфики деятельности
Оптимизация ресурсов на складе	По историческим данным определить ожидаемое движение по складу (приход товара и отгрузка). Спрогнозировать возможную пиковую нагрузку в течение месяца
AI анализ сердечно-сосудистых заболеваний	Сбор данных по анализам с учетом дополнительных демографических признаков (пол, возраст)
	Постановка диагноза в момент лабораторных испытаний (задача бинарной классификации)
	Отслеживание показателей анализов во времени и формирование рекомендаций пациенту
Предсказание лесных пожаров	Предсказание вероятности возникновения лесных пожаров, их координаты и площадь, используя исторические погодные данные, а также косвенные признаки (наличие инфраструктурных объектов, близость к транспортным сетям, социально демографические характеристики ближайших населенных пунктов)

Анализ графов

Определение оптимального маршрута между заданными вершинами	По загруженным данным определить кратчайший маршрут между вершинами графа (точками на карте города)

Анализ табличных данных

Работа с пропущенными значениями или пропусками в табличных данных	Возможность работы с пропущенными значениями или пропусками в табличных данных в формате CSV, используя метод частичного заполнения и удаления пропусков в виде удаления строк
Поиск и удаление выбросов	Обработки выбросов в датасете с использованием графика boxplot, визуальным определением и удалением выбросов методом 3-х сиг, отображением таблицы с выбросами и сохранением результатов в отдельный датасет

Эволюционный алгоритм

Генетический алгоритм	Применяет метод оптимизации, использующий принципы естественного отбора и генетического перехода в популяции организмов. Основная идея генетических алгоритмов заключается в создании популяции, которая представляет собой набор индивидуумов, каждый из которых представляет собой потенциальное решение задачи оптимизации

Сравнительная таблица обученных моделей	Формирования сравнительной таблицы обученных моделей в рамках исследования с возможностью отображения наилучшей модели и построения графика roc-auc кривой для каждой модели по отдельности, а также всех моделей на одном графике

3. Архитектура

Платформа RAZUM AI представляет собой контейнер-ориентированную архитектуру, связывающую микросервисы, упакованные в docker контейнеры. Платформа поделена на модули, которые представляют из себя несколько контейнеров, объединенных в pod’ы. Оркестрация контейнерами выполняется с использованием Kubernetes – Программного обеспечения с открытым исходным кодом.

Платформа поддерживает распределенные вычисления, а также вычисления с использованием TPU и GPGPU.

4. Модельный ряд

Конфигурация модели № 1:

Однонодовый кластер

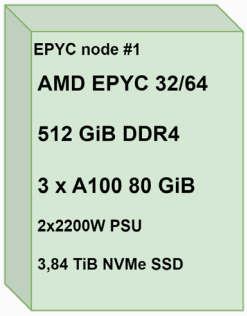
Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD



Конфигурация модели № 2:

Двухнодовый кластер

№1

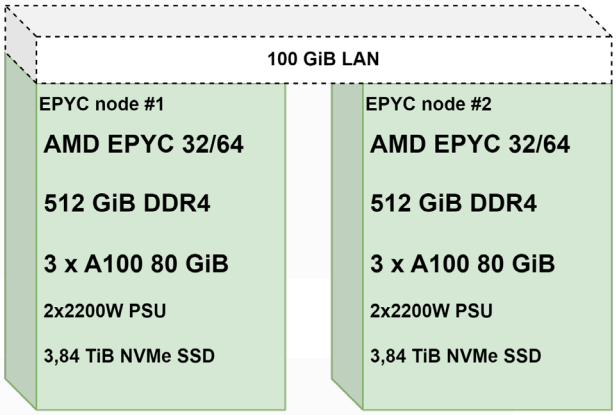
Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD



№2

Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD

Конфигурация модели № 3:

Трёхнодовый кластер с СХД с отказоустойчивой архитектурой

№1

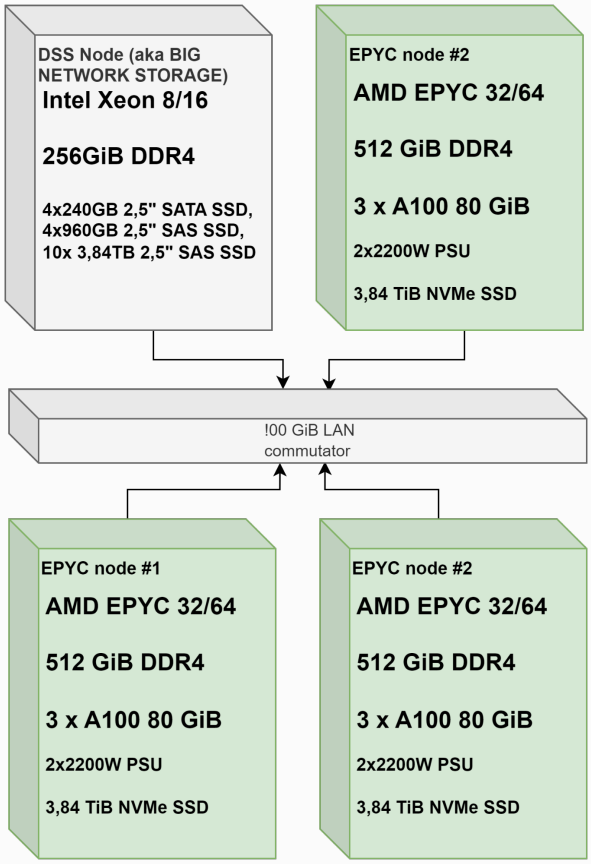
Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD



№2

Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD

№3

Процессор: AMD EPYC 7513 (32 cores /64 threads)

ОЗУ: 512 GiB DDR4

Графический ускоритель: 3 шт. x NVidia Tesla A100 80 GiB

Питание: 2x2200W PSU

Хранилище: 3,84 TiB NVMe SSD

СХД

Процессор: 4 x Intel Xeon Silver 4215R - 8 cores/16 threads

ОЗУ: 8x32GiB (256 GiB) DDR4-3200

Хранилище: 4x240GB 2,5" SATA SSD,

4x960GB 2,5" SAS SSD,

10x 3,84TB 2,5" SAS SSDЫ

5. Основные свойства RAZUM AI

Ключевые преимущества RAZUM AI

Эффективность

−Быстрое развертывание всей инфраструктуры - за один день от распаковки до загрузки пользовательских данных и начала обучения моделей;

−Поддержка оптимального набора библиотек и правил предобработки данных “из коробки” и различных модулей аппаратного ускорения гарантируют быстрое решение всех типовых задач;

−Возможность использовать экспертов как для решения задачи в целом, так и для обучения персонала заказчика позволяет одновременно достичь как быстрого эффекта в конкретных задачах, так и нужного уровня готовности сотрудников заказчика к самостоятельной работе;

−Поддержка большого набора наиболее оптимальных библиотек и всех доступных специализированных процессоров позволяет достичь максимальной скорости как обучения моделей, так и их работы над текущими данными, либо выбрать баланс между скоростью работы/энергопотреблением/стоимостью системы;

−Заказчик на этапе сайзинга прозрачно видит состав системы – нет неявных избыточных компонентов, вся функциональность включена в цену – нет скрытых дополнительных модулей или активаторов пакетов функций. Максимальное использование СПО минимизирует лицензионные отчисления третьим сторонам, заказчик может дозаказать емкость СХД и/или специализированные процессоры AI по мере необходимости, достигая этим баланс между производительностью системы/стоимостью/энергопотреблением, выбирать оптимальный вариант поддержки и сопровождения.

Масштабирование

−От терабайт до петабайт – подсистема СХД масштабируется вертикально добавлением дисковых полок, поддерживая рост объема данных на два порядка и более и скорость обработки путем добавления контроллеров (от 2 до 8).

−Гибкое использование твердотельных и вращающихся дисков дает оптимальный баланс между емкостью и скоростью.

−В сервер AI можно добавлять дополнительные спецпроцессоры по мере необходимости.

−Использование уникальной контейнерной DevOps-ориентированной инфраструктуры позволяет одновременно работать с сотнями разных моделей и разных версий одинаковых моделей, выбирая оптимальные комбинации библиотек и/или вариаций предподготовки данных.

−При необходимости можно добавить дополнительные сервера AI.

−При появлении более быстродействующих компонентов СХД и сервера AI поддерживается их установка/ замена устаревших компонентов при гарантии полной сохранности всех данных, моделей и т.п.

Надежность

−Подсистема СХД корпоративного уровня обеспечивает максимальную доступность данных и отсутствие единой точки отказа.

−Поддержка всего программно-технического комплекса, как аппаратной, так и программной составляющей со стороны ООО «АИБ» или компаний партнёров, обеспечение совместимости новых компонентов с изначально заказанными, обеспечение жизненного цикла моделей равного или более длительного, чем таковой для аппаратного обеспечения (с помощью DevOps-ориентированной контейнеризованной инфраструктуры.

−Поддержка широко распространенных стандартов мониторинга, интеграции в инфраструктуру безопасности заказчика и ГОСТ-шифрование между СХД и источниками данных и сервером AI.

−При необходимости персонал заказчика может быть обучен всем процедурам, позволяющим самостоятельно реагировать на нештатные ситуации.

Препроцессинг входных данных

Скрипты препроцессинга, написанные на языке Python v3, обеспечивают обработку и первичный анализ входных данных, структурируют данные и подготавливают их для загрузки в модуль машинного обучения. Параметры скриптов препроцессинга данных настраиваются из интерфейса пользователя.

Модуль библиотек и скриптов для обработки и препроцессинга данных выполняет следующие основные функции:

- *анализ структуры, распределения и полноты данных;*
- *предварительная обработка, нормализация и стандартизация входных данных;*
- *преобразование, параметризация и подготовка данных для передачи в модели машинного обучения.*

Модуль содержит методы, которые обеспечивают обработку и анализ входных данных в соответствии с их типом:

***тексты** (препроцессинг, очистка, лемматизация, кодирование, представление многомерными векторами, определение ключевых слов;*

***табличные данные** (заполнение пропущенных значений и пропусков в табличных данных в формате CSV, используя метод частичного заполнения и удаления пропусков в виде удаления строк);*

***временные ряды** (тесты на нормальность и стационарность ряда, декомпозиция ряда, приведение ряда к стационарному);*

***числовые и категориальные данные** (заполнение пропущенных данных, проверка на нормальность, нормализация и стандартизация, выявление аномалий, проверка на сбалансированность и ребалансировка, выделение наиболее важных признаков, кодирование, генерация новых признаков);*

***графические данные** (приведение к единой цветовой палитре, обрезка, дополнение данных и генерация новых образов, бинарная сегментация).*

Типовые сценарии

Классификация изображений

Все сущности окружающего мира поделены на заранее известные классы. Платформа RAZUM AI распознает загружаемые изображения и определяет их принадлежность к тому или иному классу сущностей.

Кластеризация научных текстов «Scopus»

«Scopus» – всемирная база данных для отслеживания цитируемости статей, опубликованных в научных изданиях. База данных индексирует научные журналы, материалы конференций и серийные книжные издания, а также «профессиональные» журналы по техническим, медицинским и гуманитарным наукам.

Задача кластеризации относится к классу задач «без учителя» – Научный текст с заранее неизвестным содержимым самостоятельно разбирается на некоторое количество групп (кластеров), связанных между собой наборов ключевых слов.

Научный текст для анализа, разбитый на строки в исходном файле, автоматически разбивается на кластеры по принципу: Строки, в которых встречаются похожие по смыслу слова, объединяются в один кластер. В отчете перечисляются наборы таких ключевых слов, с указанием количество строк, в которых они встречаются.

Обучение нейронной сети для классификации текстов

В данном алгоритме создается модель нейронной сети, умеющая распознавать авторов произведений. Для этого используется набор обучающих файлов, в которых нейронная сеть видит правильные ответы и на основании этих ответов происходит ее обучение.

Количество прохождения алгоритма обучения нейронной сети равняется нескольким эпохам, по мере прохождения эпох доля верных ответов на проверочной модели возрастает.

Предобученная модель нейронной сети сохраняется и может использоваться повторно для распознавания авторов произведений, на которых было пройдено обучение.

Распознавание автора текста по предобученной модели

В качестве входной информации для алгоритма используется файл в текстовом формате, содержащий произведение автора, для распознавания которого была обучена нейронная сеть.

Результатом работы алгоритма является отображение автора анализируемого текста, с указанием точности результата в процентах.

Кластеризация патентов и выявление трендов по патентам

В данном алгоритме выполняется анализ входящей базы с патентами с последующим выявлением и группированием патентов в кластеры по определенным признакам, которые заранее не предопределены и выбираются Программой автоматически.

Формируются кластеры с информацией о патентах, объединенные набором ключевых слов. Для каждого кластера указывается количество входящих в него патентов. Информация о патентах визуализируется – Кластеры переносятся на двумерную плоскость по рассчитанным уникальным координатам каждого слова во входящей базе с патентами. Используется цветовая индикация для кластеров в соответствии с рассчитанной плотностью входящих в него слов.

По полученным кластерам выделяются тренды – тенденции изменения патентов (например, частоты цитируемости патентов в научных изданиях) по времени.

Таким образом выполняется кластеризация патентной информации, ее количественный анализ, анализ цитирования, обработка естественного языка, визуализация данных. Результаты кластеризации патентов могут быть использованы для определения стран, в которых ведутся научные исследования, для выявления технологической направленности научных исследований и т.д.

Прогнозирование временного ряда

Временной ряд – это собранный в разные моменты времени статистический материал о значении каких-либо параметров (в простейшем случае одного) исследуемого процесса.

В данном алгоритме Программа прогнозирует изменение параметров временного ряда на основании ранее полученных статистических данных. Предсказанные данные за прошедший период отображаются одновременно с реальными данными для визуализации точности прогнозирования.

Прогнозирование свойств композитных материалов

Программой выполняется прогноз свойств композита – целевого материала, изготовленного из известных исходных компонентов, имеющих различные физические и/ или химические свойства.

Отображается информация об исходных свойствах компонентов композита и о рассчитанных прочностных характеристиках композита.

Обучение модели прогнозирования временного ряда котла

В данном алгоритме создается модель прогнозирования значений *целевых* показателей водогрейных и паровых котлов по времени.

Предварительно подготавливается файл, содержащий статистические данные, характеризующие работу котла, зафиксированные в равных промежутках времени (с шагом ресемплирования). Эти данные разделяются на две части: обучающую и тестовую выборки. На обучающей выборке выполняется обучение будущей модели прогнозирования временного ряда. На тестовой выборке проверяется точность работы полученной модели – сравниваются фактические и прогнозные значения выбранных целевых показателей.

Прогнозирование временного ряда котла в онлайн-режиме

В данном алгоритме анализируется непрерывный поток входных данных, получаемый в режиме реального времени с котла. Для тех же целевых показателей, которые были выбраны при обучении модели прогнозирования, назначаются верхний и нижний пределы допустимых значений. В алгоритме используется ранее обученная модель, способная обрабатывать поступающие данные, предсказывая результат. При фактическом и/или прогнозируемом выходе значений показателей за границы допустимых интервалов, пользователю отправляется сообщение в телеграм-канал. При этом прогнозируемые значения рассчитываются на шаг вперед, с прибавлением шага ресемплирования, позволяя пользователю своевременно среагировать и предпринять необходимые действия.

Обучение модели прогнозирования лесного пожара

В данном алгоритме создается модель, умеющая прогнозировать вероятность возникновения лесного пожара на определенной территории. На вход алгоритма подается статистический материал с показателями погодных условий исследуемой

территории, собранный за определенный промежуток времени. Также как и для модели котла, эти данные разделяются на обучающую и тестовую выборки. На обучающей выборке с помощью *алгоритма бинарной классификации* анализируется взаимосвязь между всеми показателями погодных условий и значением целевого признака (1 – факт пожара / 0 – его отсутствие). На тестовой выборке выполняется проверка обученной модели: рассчитываются значения целевого признака, затем ответы модели машинного обучения сравниваются и валидируются с фактическими событиями.

Прогнозирование возникновения лесного пожара в онлайн-режиме

В данном алгоритме анализируется непрерывный входной поток данных, содержащий значения погодных показателей на местности. Эти данные, за исключением целевого признака, проходят предварительную обработку, приводятся к виду, пригодному для анализа. Для целевого признака устанавливается граничное значение (от 0 до 1), выход за пределы которого будет означать наступление интересующего события.

Онлайн-данные анализируются с использованием обученной модели прогнозирования лесного пожара. При фактическом и/или прогнозируемом выходе целевого признака за граничное значение пользователю отправляется сообщение на телеграм-канал.

Обучение модели поиска аномалий с помощью алгоритма кластеризации

Алгоритм кластеризации **DBScan** группирует объекты по кластерам, используя параметры: минимальное количество ближайших соседей, расстояние до ближайших соседей и метрика расстояния. Наблюдения/объекты из входной выборки данных, которые не попадают ни в одну группу кластеров, считаются аномалиями или шумами.

При обучении модели используется оптимизация гиперпараметров с помощью метода **GridSearchCV** библиотеки **Sklearn**. Метрика для оптимизации – **Silhouette**, которая учитывает расстояния от объекта до остальных объектов внутри его кластера и до объектов в других кластерах, и принимает значения от -1 до 1. Максимальное значение метрики характеризует лучшую модель с лучшими гиперпараметрами.

В результате работы алгоритма:

1.Создается модель машинного обучения, с лучшими гиперпараметрами, и максимальной метрикой.

2.Отображается список кластеров, с указанием количества объектов в каждом кластере, а также количество объектов, не попавших ни в один из кластеров.

3.Каждому кластеру присваивается отдельная метка (с нумерацией от 0), а аномалиям – метка со значением -1.

4.Аномалиям присваивается флаг со значением 1. Эта информация понадобится в алгоритме бинарной классификации, где понадобится бинарный признак (0 или 1), обозначающий принадлежность наблюдения к кластеру.

5.Трехмерный график с объектами, подкрашенными в соответствии с меткой кластера.

Обучение модели поиска аномалий с помощью алгоритма бинарной классификации

Алгоритм бинарной классификации **XGBClassifier** создает модель машинного обучения, умеющую распознавать аномалии во входящем трафике данных. Входными данными для алгоритма является датасет с флагом аномалии (поле **outlier_flag**),

полученный в результате работы алгоритма кластеризации (после «обучения без учителя»).

Выполняются операции:

1.В качестве целевого признака выбирается поле **outlier_flag**.

2.Категориальные признаки кодируются методом **OHE** (также как в алгоритме кластеризации), далее над всеми признаками (кроме целевого) проводится стандартизация.

3.Исходный датасет делится на обучающую (80%) и тестовую (20%) выборки. Так как классификация относится к задачам «обучения с учителем», то на 80% датасета модель обучается соотносить наблюдение к аномалиям, а на оставшихся 20% – ответы обученной модели валидируются.

4.Обучение модели выполняется с использованием гиперпараметров:

-*n_estimators* – количество деревьев решений в ансамбле. Подбирается достаточное количество деревьев/базовых моделей, чтобы повысилось качество на обучающей выборке, а качество на тестовой не выходило на асимптоту;

-*max_depth* – максимальная количество деревьев. Для задачи с предположительно большим количеством шумов используется небольшое количество деревьев.

По итогам валидации рассчитывается значение метрики **F1**, характеризующей количество правильных ответов модели. Идеально, когда значение метрики **F1** приближается к 1 (что означает 100% правильность ответов модели).

Для каждого наблюдения предсказывается бинарное значение целевого признака: определяется относится наблюдение к аномалиям или нет. А результат работы обученной модели на тестовой выборке данных отображается в *матрице ошибок*, наглядно отображающей количество правильных ответов модели и количество ошибок (в разрезе прогнозируемых и фактических значений целевого признака).

Поиск корреляции между признаками в датасете

В анализируемом датасете выполняется поиск признаков, имеющих сильную корреляцию. Для этого рассчитывается *коэффициент корреляции* между признаками, принимающий значения от -1 до 1: Значение коэффициента ближе к 1 означает сильную прямую связь, коэффициент ближе к -1 – обратную связь, коэффициент ближе к 0 – отсутствие связи. Задается топ-K значений – количество максимальных значений корреляции, которые попадут в *тепловую диаграмму*.

Алгоритм сначала строит матрицу корреляций по всем признакам, представляющей собой квадратную таблицу, столбцы и строки которой – анализируемые признаки. На пересечении строк и столбцов выводится рассчитанный коэффициент корреляции. На главной диагонали находятся коэффициенты корреляции, равные 1. В этой таблице определяются топ-K максимальных значения корреляции, по которым отбираются признаки для тепловой диаграммы.

Строится тепловая диаграмма, столбцы и строки которой – признаки в датасете, которые имеют сильную положительную или отрицательную корреляцию. Пользователь анализирует какие коэффициенты корреляции значимы, и принимает решение, какие признаки из датасета можно исключить, чтобы улучшить качество анализа данных. Коррелируемые признаки могут быть идентичны (если коэффициент корреляции ближе к 1) и исключение одного признака никак не повлияет на анализ. Некоторые признаки лучше исключить, чтобы улучшить качество анализа.

Поиск аномалий с помощью метода косинусного сходства

В данном алгоритме по заданному набору признаков выполняется поиск похожих наблюдений в общем объеме данных.

Задается *входной вектор*, в котором последовательно перечисляются значения признаков для анализа. Метод рассчитывает *косинусные расстояния* между вектором и выбранными столбцами в наблюдениях. Косинусное расстояние принимает значения от 0 до 2, где значение 0 означает полное совпадение между наблюдением и заданным вектором.

Формируется таблица с наблюдениями, в которой строки располагаются по мере возрастания рассчитанного косинусного расстояния. Первые топ-k строк в таблице выделяются жирным шрифтом, и имеют наибольшую схожесть с вектором, который предположительно характеризует аномалию.

Обучение модели искусственного интеллекта классификации изображений (нейронная сеть)

Модель ИИ обучается категоризировать изображения на классы. При этом реализована как бинарная классификация, так и многоклассовая.
В данном сценарии пользователь загружает в Систему изображения, которые относятся к тому или иному классу. Эти изображения делятся на обучающую и валидационную выборки, которые в свою очередь делятся на папки – в каждую папку складываются изображения одного класса.

Изображения подаются на вход обучаемой нейронной сети, которая состоит из нескольких слоев: сверточный слой, слой подвыборки и полносвязный слой. Попадая в первый слой изображение преобразуется. Во всех слоях до полносвязного выполняется предобработка изображения, и выделение различных признаков, которые затем подаются на вход классификатору.

Строится *функция потерь*, которая рассчитывает ошибку между реальными и полученными ответами нейронной сети. С помощью *алгоритма градиентного спуска* находится минимальное значение функции потерь, в которой максимизируется вероятность принадлежности к истинному классу для каждого объекта из тренировочной выборки. Точка минимума определяет оптимальные веса нейронной сети, которые соответствуют наилучшей модели.

На валидационной выборке проверяется, насколько хорошо обучилась модель.

Используется для классификации новых изображений, загруженных пользователем с локального компьютера.

Обучение модели кластеризации с применением фреймворка Apache Spark

В рамках решения задач, требующих обработки «больших данных» (файлы и базы данных, содержащих миллионы и более строк), на платформе реализована возможность применения фреймворка Apache Spark, позволяющего оптимизировать и ускорить вычисления и построить модель кластеризации с применением DBSCAN. В результате работы алгоритма:

1.Создается модель машинного обучения, с лучшими гиперпараметрами, и максимальной метрикой;

2.Отображается список кластеров, с указанием количества объектов в каждом кластере, а также количество объектов, не попавших ни в один из кластеров;

3.Каждому кластеру присваивается отдельная метка (с нумерацией от 0), а аномалиям – метка со значением -1;

4.Аномалиям присваивается флаг со значением 1. Эта информация понадобится в алгоритме бинарной классификации, где понадобится бинарный признак (0 или 1), обозначающий принадлежность наблюдения к кластеру;

5.Трехмерный график с объектами, подкрашенными в соответствии с меткой кластера.

Определение кратчайшего расстояния между двумя точками на карте города

В данном сценарии модель ИИ анализирует графические данные карты города, загруженные в формате графа для определения кратчайшего и самого оптимального расстояния между вершинами данного графа.

Встроенный функции позволяют определить:

1. Общее количество вершин и ребер графа;

2. Рассчитать ближайшие от исходной точки графа вершины;

3. Вычислить кратчайшие пути в графе, задав точки отправления и прибытия и количество маршрутов;

В результате работы платформа создает визуализацию в виде карты, на которой отмечен оптимальный маршрут между заданными точками.

Логический анализ данных. Алгоритм применяется для поддержки принятия решений при классификации и распознавании, особенно для решения задач, в которых велики негативные последствия принятия неверных решений. Алгоритм принимает на входе датасет с наблюдениями (и их признаками), которые разделены на 'положительные' и 'отрицательные' и возвращает классификацию датасета.

В результате работы алгоритма:

1. Выполняется бинаризация датасета;
2. Находится опорное множество;
3. Для каждого уникального наблюдения обучающей выборки формируется правило;
4. Производится оптимизация паттернов;
5. Выполняется классификация. Происходит определение весов отобранных правил для строк тестовой выборки, и выполняется предсказание.

Генетический алгоритм. Применяет метод оптимизации, использующий принципы естественного отбора и генетического перехода в популяции организмов. Основная идея генетических алгоритмов заключается в создании популяции, которая представляет собой набор индивидуумов, каждый из которых представляет собой потенциальное решение задачи оптимизации. Для этого моделируется процесс эволюции в несколько шагов:

1. Инициализация. Начальная популяция решений создается случайным образом.

2. Оценка. Каждое решение в популяции оценивается на основе функции пригодности, которая определяет, насколько хорошо это решение решает задачу. Это значение варьируется от 0 до 1, где 1 - лучшее решение.

3. Селекция. Лучшие решения из популяции выбираются для создания следующего поколения.

4. Скрещивание. Два решения из популяции выбираются для создания нового решения. Гены родительских решений комбинируются, чтобы создать потомство, которое наследует хорошие качества от обоих родителей.

5. Мутация. Случайное незначительное изменение всех потомков из популяции с

целью разнообразить многообразие рассматриваемых индивидов.

Все шаги повторяются для каждого поколения (их число указывает пользователь).

Определение ключевых слов в кластерах

Данный алгоритм при кластеризации текста определяет ключевые слова и отображает их вместе с визуализацией кластеров.

Ключевые слова выводятся на рабочей области и дашборде.

Определение выбросов в датасете

Алгоритм позволяет определить выбросы в датасете и получить следующий результат:

1. Отобразить таблицу с выбросами;
2. Построить график boxplot;
3. Удалить выбросы методом трёх сигм;
4. Сохранить результат в отдельный датасет.

Сравнительная таблица обученных моделей

Сравнительная таблица обученных моделей позволяет численно оценить качество обучения нескольких моделей машинного обучения и выбрать лучшую из них на основе рассчитанных метрик после обучения алгоритмов ИИ.

Данный функционал позволяет:

- отобразить все доступные метрики;
- увидеть и отобразить наилучшую модель (по метрикам);
- построить график roc-auc кривой для каждой модели на одном графике (plotly).

Сегментация изображений

Данный алгоритм анализирует исходные изображения, выделяет объекты и их границы и затем применяет результаты обработки к целевым изображениям. Практическая ценность - медицинская диагностика, робототехника, автономные транспортные системы.

Стекинг

Данный алгоритм использует ансамбль разнородных моделей для последующей обработки табличных данных. Результатом работы алгоритма является метка объекта (в случае решения задач классификации) или число (в случае решения задач регрессии). Практическая ценность - использование в антиспам, антифрод и рекомендательных системах.

Визуальный конструктор моделей

Основой системы является универсальный конструктор AI с использованием блок-схем в нотации BPMN 2.0. Конструктор позволяет создавать, обучать модели искусственного интеллекта без необходимости прямого кодирования по принципу drag n drop. Предусмотрена возможность настройки элементов блок-схемы с указанием их параметров.

Визуализация результатов работы

Программа позволяет получить графические результаты выполнения алгоритмов искусственного интеллекта и машинного обучения, с помощью платформы пользовательских визуализаций Power BI с открытым кодом. Визуализация может осуществляться при помощи графиков, таблиц и диаграмм. Например, это может быть: линейный график временного ряда, график автокорреляции и частичной автокорреляции, декомпозиция временного ряда, свечной график; тепловая, круговая или пузырьковая диаграммы; матрица ошибок и проч. Создавать такие визуализации можно как на рабочих областях, где непосредственно ведется работа с блок схемами, или в качестве отдельных сущностей на дашбордах.

Интеграция с внешними системами

В составе платформы реализованы соответствующие адаптеры для обмена данными со смежными системами на базе технологий: DCOM, REST или SOAP. Программное обеспечение Платформы имеет возможность загружать данные из смежных систем как в режиме реального времени, так и предоставленные в виде файлов/ баз данных. Платформа имеет возможность интеграции с СХД.

Конвейер приложений

В платформе реализован конвейер, благодаря которому обученная модель может быть сразу упакована в приложение, без необходимости писать отдельный код. Такое приложение можно скачать и развернуть за пределами программы (например в Docker), интегрировать с внешними системами, настроить получение входных данных, использовать функцию object detection и выполнить прогнозы.

6. Функционал программных модулей

Подсистема хранения данных

Включает в себя основные базы данных, которые необходимы для работы с данными, в процессе машинного обучения и анализа. Составные части модуля упакованы в изолированные docker-контейнеры и объединены в единый Pod, управляемый посредством Kubernetes.

Модуль хранит «сырые данные», полученные из различных источников, которые прошли проверку качества данных:

- структурированные и параметризованные данные (временные ряды, числовые данные);
- неструктурированные данные (изображения, тексты, таблицы, аудио, видео);
- пользовательские блок-схемы, отчеты, модели.

В модуле предусмотрены следующие возможности:

- масштабирование;
- автоматическое добавление данных в соответствующие базы данных, в соответствии с их типом.

В качестве баз данных на платформе используются MongoDB и PostgreSQL.

Подсистема искусственного интеллекта

Включает в себя модули, обеспечивающие работу искусственного интеллекта:

1. **Модуль анализа данных**

Модуль строится на базе существующего программного обеспечения с открытым исходным кодом (с использованием библиотек: Scikit-learn, Pandas, NumPy, SciPy), а также разработанных для Платформы библиотек, необходимых для выполнения прикладных задач, обработки данных разных типов, структур и размеров.

2. **Модуль библиотек и алгоритмов машинного обучения**

Модуль библиотек и алгоритмов машинного обучения строится на базе программного обеспечения с открытым исходным кодом, с использованием библиотек машинного обучения: Scikit-learn, Keras, Spark ML. Предусмотрена возможность обновления библиотек.

Модуль позволяет использовать основные алгоритмы машинного обучения и их ансамбли.

Модуль выполняет следующие функции:

- решение регрессионных задач;
- построение трендовых моделей;
- решение задач классификации (бинарной и множественной).

3. Модуль библиотек нейронных сетей и глубокого обучения

Модуль библиотек нейронных сетей и глубокого обучения строится на базе программного обеспечения с открытым исходным кодом, с использованием библиотек: TensorFlow, PyTorch, Keras, SparkDL.

В модуле предусмотрено построение основных архитектур нейронных сетей со слоями: Dense, Conv2D, LSTM.

Предусмотрено построение автокодировщиков, которые применяют для предварительного обучения глубокой сети без учителя. Слои обучаются друг за другом, начиная с первых.

Предусмотрено построение многомерного векторного пространства и создание векторно-семантических моделей Word2vec.

Модуль библиотек нейронных сетей и глубокого обучения выполняет следующие функции:

- решение задач классификации (бинарная и множественная);
- возможность настройки гиперпараметров прямо из режима конструктора (активационные функции, функции потерь, оптимайзеры);
- визуализация архитектуры нейронной сети.

Подсистема конструктора искусственного интеллекта

Модуль drag n drop конструктор искусственного интеллекта и машинного обучения строится на базе Программного обеспечения, написанного под проект на языке программирования JavaScript.

Модуль drag n drop конструктор выполняет следующие функции:

- визуальное моделирование – создание блок-схем машинного обучения и нейронных сетей без прямого кодирования;
- тонкая настройка параметров элементов блок-схемы;
- ввод гиперпараметров алгоритмов;
- сохранение шаблонов блок-схем для последующего использования и доработки.

Подсистема графического интерфейса

Модуль **GUI** строится на базе написанного под проект Программного обеспечения.

Модуль **GUI** предназначен для взаимодействия пользователя с Платформой. Обеспечивает интуитивно-понятный процесс создания и обучения моделей искусственного интеллекта, их сохранения, создания отчетов.

Модуль **GUI** выполняет следующие функции:

– обращение к соответствующим модулям Платформы в режиме пользовательского интерфейса;

– использование функционала модулей в режиме пользовательского интерфейса;

– гибкая настройка рабочих форм (дашбордов, отчетов, графиков);

– переход от режима без кодирования к стандартному режиму с программным кодом.

Подсистема визуализации и формирования пользовательских отчетов

Модуль визуализации и формирования пользовательских отчетов строится на базе программного обеспечения с открытым исходным кодом, с использованием библиотек: Matplotlib, Seaborn, Plotly.

Модуль выполняет следующие функции:

– отображение ключевых показателей обученных моделей;

– формирование отчетов о результатах обучения модели (график обучения, метрики ошибок и качества).

Подсистема логирования и мониторинга

Содержит модуль логирования и модуль мониторинга аппаратной части, которые позволяют осуществлять запись и отслеживание действий пользователей, а также самой системы.