

АСТРА ИИ

**Платформа АСТРА ИИ
(ASTRA AI Platform)
Техническое описание**

Версия 1.0.1

Москва
2026

СОДЕРЖАНИЕ

1. Введение.....	3
2. Типы обрабатываемых данных с применением Платформы АСТРА ИИ	5
3. Архитектура.....	8
4. Модельный ряд.....	9
5. Основные свойства Платформы АСТРА ИИ.....	11
6. Функционал программных модулей	20

1. Введение

Программное обеспечение Платформа АСТРА ИИ (ASTRA AI Platform) - это платформа (далее – Платформа) представляет собой комплексную платформу для обработки структурированных и неструктурированных данных, разработки и обучения моделей искусственного интеллекта. Платформа ориентирована на решение задач в области создания баз знаний, предиктивной аналитики, автоматизации бизнес-процессов, мультиагентного ИИ и применения крупных языковых моделей (LLM).

Работа с Платформой не требует навыков программирования: специалист формирует блок-схему последовательных преобразований данных, запускает её и получает результат. Встроенные средства визуализации: графики, диаграммы, таблицы, изображения - позволяют наглядно отследить ход вычислений и представить итоги анализа.

Функциональные возможности Платформы охватывают полный цикл работы с данными:

- анализ данных - препроцессинг, предобработка и т. д.;
- машинное обучение - прогноз, классификация, кластеризация и т. д.;
- глубокое обучение - классификация, регрессия, распознавание объектов и т. д.

Интерес к технологиям искусственного интеллекта со стороны заказчиков из различных отраслей неуклонно растёт. При этом большинство из них стремятся минимизировать капитальные вложения, сократить сроки внедрения и обеспечить жизнеспособность решения на горизонте не менее 3–5 лет.

С учётом этих требований Платформа АСТРА ИИ разработана на основе следующих ключевых принципов:

- **Масштабируемость хранилища данных.** Современные задачи ИИ требуют сверхбольших объёмов данных, накапливаемых годами — как в целях статистического анализа, так и в соответствии с требованиями законодательства. Платформа обеспечивает горизонтальное масштабирование хранилища в соответствии с ростом объёма данных на протяжении всего жизненного цикла решения;

- **Производительность в реальном времени.** Обеспечивается оперативная нормализация, пред- и постобработка данных, а также загрузка ядра данных и моделей в систему ИИ без задержек.

- **Поддержка всех типов данных.** Платформа одинаково эффективно работает со структурированными данными (включая объектное хранение) и неструктурированными источниками (социальные сети, открытые датасеты и др.);

- **Аппаратная независимость.** Поддерживается широкий спектр высокопроизводительных специализированных процессорных платформ, что обеспечивает гибкость при изменении задач и эволюции аппаратных решений;

- **Эластичность ресурсов.** Вычислительная мощность динамически распределяется в соответствии с текущими приоритетами задач — между хранением, дедупликацией, предобработкой данных и непосредственно задачами ИИ;

- **Инструменты подготовки данных.** В состав Платформы входят инструменты для очистки, разметки и подготовки обучающих датасетов;

- **Сопровождение «под ключ».** Предоставляется полный спектр услуг: от сайзинга аппаратной платформы и выбора алгоритмов до сбора, разметки датасетов и обучения моделей;

- **Открытость и самостоятельность.** Заказчик может привлекать внешних консультантов или работать самостоятельно, опираясь на исчерпывающую документацию.

2. Типы обрабатываемых данных с применением Платформы АСТРА ИИ

Анализ изображения

Классификация изображений	Классификация дефектов продукции на производственной линии
	Распознавание дефектов на изображениях с использованием нейронных сетей. Автоматическая классификация дефектов без участия оператора. Точность классификатора более 90%
Сегментация изображений	Сегментация изображений для выделения объектов
	Анализ изображений с выделением границ объектов и применением результатов и к целевым изображениям. Практическая ценность: медицинская диагностика, робототехника, автономные системы
Поиск похожих изображений без разметки	Поиск эталонного изображения с помощью метрической классификации
	Поиск изображений, похожих на заданный эталон, с использованием метрической классификации без разметки. Точность классификации более 92%
Метрическая классификация изображений	Метрическая классификация на основе предобученных моделей для поиска похожих изображений без разметки. Точность классификации более 92%
	Поиск изображений, похожих на заданный эталон, с использованием метрической классификации без разметки

Анализ текста

Классификация текстов по тематике	Классификация текстов по тематике с использованием нейронных сетей. Автоматическая категоризация документов по тематике
Кластеризация текстов	Группировка текстовых документов по смысловой близости с автоматическим определением ключевых тем
Определение ключевых слов в текстах	Выделение ключевых слов в текстовых кластерах после кластеризации с визуализацией на дашборде
Лемматизация и векторизация текста	Препроцессинг текста: токенизация, лемматизация, векторизация для подачи в модель LLM

Анализ числовых данных/временных рядов

Предиктивная аналитика потребления ресурсов	Прогнозирование пиковой нагрузки на вычислительные ресурсы на основе исторических данных. Автоматическое масштабирование при достижении пороговых значений
	Формирование датасетов из внешних БД через ETL с загрузкой в PostgreSQL, MongoDB и автоматической валидацией датасета

Интеграция с внешними базами данных через ETL	Анализ временных рядов с автоматическим обнаружением аномалий и визуализацией декомпозиции
	Мониторинг состояния платформы через средства OpenTelemetry/Prometheus. Своевременное обнаружение отклонений и автоматическое оповещение
	Считывание данных из внешних БД в режиме реального времени и подача в модель для прогнозирования
Прогнозирование временных рядов	Прогнозирование значений временного ряда с визуализацией факт/прогноз. Используются модели ARIMA, SARIMA, Prophet. Оценка по метрикам RMSE, MAE, WMAPE
Обнаружение аномалий в табличных данных	Преобработка табличных данных: заполнение пропусков, нормализация, стандартизация, фильтрация и агрегация
	Обучение модели поиска аномалий (Isolation Forest, DBSCAN) на подготовленном датасете. Оценка по метрикам Silhouette, Calinski, Davies
	Визуализация результатов: 3D-график кластеров, таблица статистики по кластерам, сравнительная таблица метрик
Кластеризация числовых данных	Группировка числовых данных без учителя: K-Means, DBSCAN, OPTICS, агломеративная кластеризация. Оценка по метрикам Silhouette, Calinski, Davies

Мультиагентный ИИ

Автоматизация бизнес-процессов через мультиагентный ИИ	Создание автономных ИИ-агентов с поддержкой инструментов, памяти и RAG. Мультиагентная оркестрация для автоматизации бизнес-задач
RAG-система на корпоративных данных	Загрузка документов компании в векторную базу данных. LLM-агент отвечает на вопросы сотрудников, извлекая релевантные фрагменты через семантический поиск без передачи данных во внешние сервисы
Интеграция внешних инструментов через MCP	Подключение внешних источников данных и API к ИИ-агентам через стандартизированный протокол MCP в режиме клиента и сервера
Автономный агент для многоэтапных задач	Агент самостоятельно разбивает сложную задачу на шаги, обращается к базам данных, использует инструменты и принимает промежуточные решения для достижения конечного результата без участия пользователя

Анализ табличных данных

Работа с пропущенными значениями или пропусками в табличных данных	Заполнение пропусков в табличных данных с автоматической валидацией, агрегацией и фильтрацией. Формирование ETL-пайплайна
Поиск и удаление выбросов	Поиск и удаление выбросов методом 3-х сигм, визуализация boxplot. Сохранение результата в отдельный датасет для дальнейшей обработки

Эволюционный алгоритм

Подбор гиперпараметров (Grid Search)	Автоматизированный подбор гиперпараметров моделей машинного обучения методом поиска по сетке (Grid Search). Обеспечивает максимальное качество обученных моделей
Сравнительная таблица обученных моделей	Сравнительная таблица обученных моделей с выделением лучшей модели. Визуализация ROC-AUC для всех моделей на одном графике

3. Архитектура

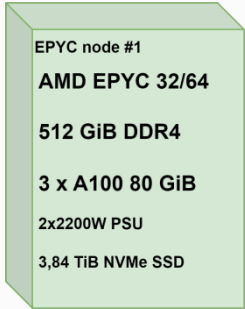
Платформа АСТРА ИИ построена на основе контейнерной архитектуры. Все компоненты системы реализованы в виде микросервисов, упакованных в Docker-контейнеры. Оркестрация контейнеров осуществляется средствами Kubernetes — платформы управления контейнерными рабочими нагрузками с открытым исходным кодом.

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



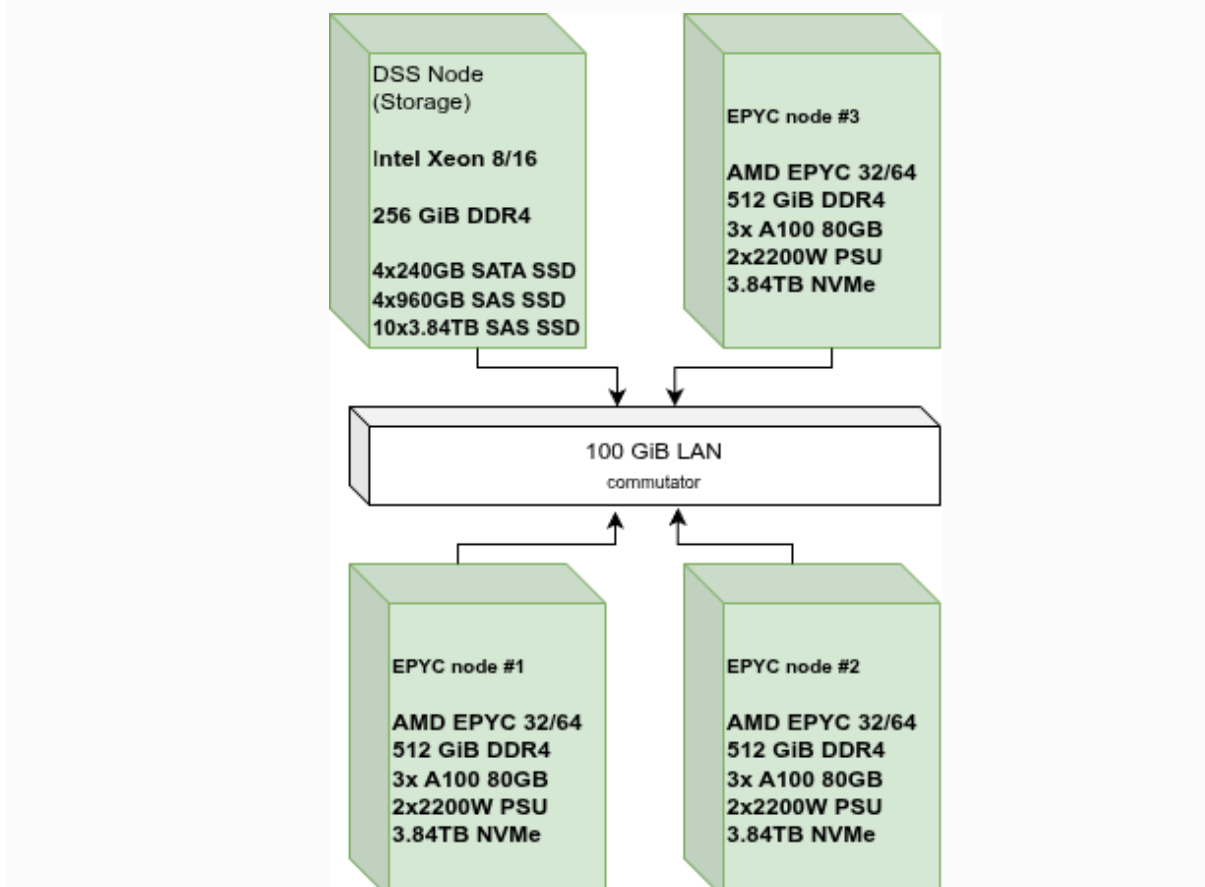
The diagram shows a 3D box representing a single node. The text inside the box lists the specifications: EPYC node #1, AMD EPYC 32/64, 512 GiB DDR4, 3 x A100 80 GiB, 2x2200W PSU, and 3.84 TiB NVMe SSD.

Конфигурация модели № 2:

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 SSD.

5. Основные свойства Платформы АСТРА ИИ

Ключевые преимущества Платформы АСТРА ИИ

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

В данном сценарии пользователь загружает в платформу набор размеченных изображений, разбитых по классам. Платформа АСТРА ИИ обучает нейронную сеть для классификации изображений по заданным классам, поддерживая как бинарную, так и многоклассовую классификацию. Изображения проходят через сверточные слои, слои подвыборки и полносвязные слои. Точность классификатора оценивается по метрикам accuracy, F1 и матрице ошибок.

Кластеризация текстовых документов

В данном сценарии пользователь загружает корпус текстовых документов и запускает пайплайн кластеризации.

Платформа выполняет препроцессинг: токенизацию, лемматизацию и векторизацию текстов.

Алгоритм группирует тексты по смысловой близости с помощью векторных представлений. Формируются кластеры с ключевыми темами и визуализацией на дашборде.

Обучение нейронной сети для классификации текстов

В данном сценарии создается модель нейронной сети, решающая задачу классификации текста по тематике. Для обучения используется набор размеченных документов, в которых нейронная сеть видит правильные классы и обучается их распознавать.

Обучение выполняется через несколько эпох. По мере обучения доля верных ответов на валидационной выборке возрастает.

Обученная модель сохраняется с метаданными и применяется для классификации новых документов. Оценка по метрикам accuracy, F1, precision, recall.

Применение обученной модели для классификации (инференс)

Обученная модель применяется для инференса на новом датасете. Входные данные проходят предобработку и подаются на вход модели. Результат — числовое значение предсказанного класса с указанием вероятности.

Результат интегрируется в пайплайн для дальнейшей обработки.

Прогнозирование временного ряда (оффлайн)

Временной ряд – это данные, собранные в разные моменты времени. В данном сценарии пользователь загружает исторические данные временного ряда, настраивает модель (ARIMA, SARIMA или Prophet) и получает прогноз.

Платформа выполняет декомпозицию ряда: выделяет трендовую и сезонную составляющие, выполняет тесты на стационарность. Прогноз визуализируется одновременно с фактическими данными для оценки точности прогноза.

Прогнозирование временного ряда

В данном сценарии пользователь загружает исторические данные временного ряда и запускает прогнозирование.

Платформа применяет модели ARIMA, SARIMA, Prophet. Предсказанные значения отображаются одновременно с фактическими для оценки точности по метрикам RMSE, MAE, WMAPE.

Прогнозирование временного ряда (онлайн)

В данном сценарии анализируется непрерывный поток входных данных, получаемых в режиме реального времени.

Для целевых показателей назначаются границы допустимых значений. При выходе за границы пользователю отправляется автоматическое уведомление.

Обучение модели прогнозирования временного ряда (оффлайн)

В данном сценарии пользователь загружает исторические данные временного ряда и запускает прогнозирование в режиме оффлайн. Создаётся модель прогнозирования значений *целевых показателей* временного ряда.

Данные разделяются на обучающую и валидационную выборки. Платформа выполняет декомпозицию ряда, тесты на стационарность, подбирает оптимальную модель (ARIMA, SARIMA, Prophet). Прогноз визуализируется одновременно с фактом, оценка по метрикам RMSE, MAE, WMAPE. Обученная модель сохраняется для повторного использования.

Прогнозирование временного ряда (онлайн)

В данном сценарии анализируется непрерывный поток входных данных, поступающих через ETL-интеграцию из внешнего источника в режиме реального времени. Для целевых показателей назначаются границы допустимых значений. Используется ранее обученная модель для предсказания значений на шаг вперед. При выходе значений за допустимые границы пользователю отправляется автоматическое уведомление.

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 **ONE** (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-К 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-К 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-к 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.

Мониторинг платформы через OpenTelemetry и Prometheus

В данном сценарии средства мониторинга платформы отслеживают состояние всех микросервисов в режиме реального времени.

Средства мониторинга позволяют определить:

1. Состояние каждого микросервиса: CPU, память, диск, сеть;
2. Историю запросов к всем модулям с временными метками и статусами;
3. Аномалии в работе платформы с автоматической фиксацией отклонений.

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 госс-аус 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.

RAG-система на основе корпоративных данных

В данном сценарии пользователь загружает корпоративные документы в векторную базу данных. LLM-агент, развёрнутый локально через модуль MCP, отвечает на вопросы пользователя, извлекая релевантные фрагменты через семантический поиск. Данные не передаются во внешние сервисы.

Контекст между сессиями сохраняется через настраиваемую память агента. Система поддерживает протокол MCP для стандартизированной интеграции дополнительных инструментов.

Построение мультиагентного рабочего процесса через DAG-конструктор

Пользователь через визуальный конструктор соединяет компоненты (LLM, промпты, инструменты, память) в направленный ациклический граф без написания кода. Каждый узел выполняет свою задачу и передаёт результат следующему. Готовый рабочий процесс экспортируется в JSON для повторного использования и интеграции в инфраструктуру.

Автономный агент с доступом к внешним инструментам через MCP

Агент подключается к внешним источникам данных и API через протокол MCP в режиме клиента. Агент самостоятельно принимает решения о последовательности действий, обращается к нужным инструментам и возвращает результат. Контекст между сессиями сохраняется через настраиваемую память агента.

Мультиагентная оркестрация для решения комплексных задач

Пользователь создаёт систему из нескольких специализированных агентов, каждый из которых отвечает за свою часть задачи. Главный агент делегирует подзадачи специализированным агентам, управляет их взаимодействием и собирает итоговый результат. Такой подход позволяет автоматизировать сложные многоэтапные бизнес-процессы, которые невозможно решить одним агентом.

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. 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.

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);
- 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. 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.

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 мультиагентного ИИ

Модуль мультиагентного ИИ представляет собой интеллектуальный роботизированный компонент платформы для разработки и интеграции автономных AI-агентов для автоматизации бизнес-процессов. Модуль выполняет следующие функции:

– визуальный DAG-конструктор на основе узлов для создания направленных ациклических графов рабочих процессов ИИ: пользователь соединяет компоненты (LLM, промпты, инструменты, память) для построения и визуализации рабочих процессов без глубоких навыков программирования по принципу drag & drop;

– мультиагентная оркестрация: создание сложных систем, в которых несколько специализированных агентов взаимодействуют для решения комплексных задач, включая управление взаимодействием между агентами и делегирование задач;

– RAG (Retrieval-Augmented Generation): встроенные компоненты для создания систем дополненной генерации — взаимодействие LLM с пользовательскими наборами данных, векторными базами данных и выполнение семантического поиска;

– поддержка локально развёрнутых LLM-моделей для работы агентов в защищённых контурах без передачи данных внешним сервисам;

– поддержка протокола MCP (Model Context Protocol) в режиме клиента и сервера для стандартизированной интеграции внешних инструментов и источников данных в агенты;

– разработка агентного ИИ: создание автономных агентов, способных использовать инструменты, обращаться к базам данных и принимать решения для выполнения многоэтапных действий;

– управление памятью агентов: встроенная настраиваемая память чата для сохранения контекста между сессиями, обеспечивающая непрерывность работы агентов в долгосрочных задачах;

– интеграция с векторными базами данных, поисковыми провайдерами и загрузчиками файлов; экспорт рабочих процессов в виде JSON-файлов для обмена и тиражирования ИИ-конвейеров;

– возможность создания кастомных компонентов на Python для расширения функциональности модуля, включая специализированную обработку данных и интеграцию с внешними API.

Подсистема логирования и мониторинга

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