Отчёт о разработке и развертывании сервиса предсказания цен на основе текстовых данных

Описание проекта

Проект направлен на создание сервиса, который предсказывает изменения цен на финансовые активы, такие как нефть (WTI, Brent) и фондовые индексы (S&P 500, Nasdaq-100). Основное внимание уделено обработке текстовой информации (новости, аналитика) и её интеграции с рыночными данными для построения точных прогнозов.

Архитектура сервиса

Сервис реализован с использованием:

- **Streamlit** как пользовательский интерфейс.
- **Виртуального частного сервера**, развернутого через Docker-контейнеры для обеспечения производительности и изоляции.
- API для загрузки рыночных данных и новостей.

Предобработка данных

1. Сбор данных

- Рыночные данные собираются через API биржевых и финансовых платформ.
- Новостная информация по ключевым активам агрегируется через тематические новостные API.

2. Обработка рыночных данных

- Расчёт **технических индикаторов**: средние скользящие (SMA), экспоненциальные скользящие (EMA), волатильность, объемы торгов.
- Формирование временных рядов с учётом различных временных лагов.

3. Обработка текстовых данных

- Сентимент-аналитика с использованием **FinBERT**:
    - Определение тональности новостей (позитивная, негативная, нейтральная).
    - Агрегация сентиментов на уровне дня.
- Построение производных метрик:
    - Скользящие средние сентиментов за неделю и месяц.
    - Разности и ускорения (акселерации) сентиментов.

4. Интеграция данных

- Все обработанные данные объединяются в единый датасет.
- Отбор наиболее значимых признаков для минимизации переобучения.

Основной функционал

1. Анализ данных

- Визуализация временных рядов:
    - Линейные и свечные графики.
    - Гистограммы распределения метрик.
    - Корреляционные матрицы.
- Расчёт описательной статистики (средние значения, медианы, квартили).

2. Модели и прогнозирование

- Предобученные модели **Ridge-регрессии** для прогнозирования цен активов.
- Возможность обучения моделей на пользовательских датасетах (при наличии необходимых столбцов).
- Построение прогнозов на заданный временной горизонт.

3. Загрузка пользовательских данных

- Пользователи могут загружать собственные датасеты в формате CSV.
- Интеграция данных в аналитическую платформу с последующей обработкой и прогнозированием.

Реализация серверной части

Серверная часть сервиса реализована на основе FastAPI, что обеспечивает высокую производительность и гибкость в разработке.

1. Основные модули API:

data_router.py: Отвечает за управление загрузкой и доступом к данным. Содержит следующие эндпоинты:

- /get_data_status: Возвращает интервалы данных, для которых в базе данных доступны данные о тикерах и новостях.
- /get_price_table: Получение табличных данных о ценах.
- /get_news_sentiment: Получение данных о тональности новостей.

model_router.py: Отвечает за получение информации об обученных моделях и позволяет обучать новые модели машинного обучения.

- /train: Запуск процесса обучения новой модели.
- /list_models: Получение списка доступных моделей и их статусов.

- /predict_price: Предсказание цен с использованием обученной модели.
- /list_inference_attributes: Получение списка доступных атрибутов, вычисленных в результате инференса.
- /get_inference_attribute_values: Получение значений атрибутов, вычисленных в ходе инференса.

2. Ключевые технологии и подходы:

- Модели данных: Для валидации входных и выходных данных используются Pydantic-модели, определенные в data_schemas.py и model_schemas.py.
- Логирование: Для отслеживания выполнения запросов, выявления ошибок и отладки используется стандартная библиотека logging.
- Асинхронность: Все основные операции API, такие как загрузка данных, обработка и предсказание, выполняются асинхронно с использованием async и await.
- Базы данных: Для хранения загруженных и обработанных данных используется база данных MariaDB.
- Внешние API: Для получения данных используются Yahoo Finance API и Webzio API.

Реализация клиентской части

Клиентская часть сервиса реализована на основе библиотеки Streamlit, обеспечивающей интерактивный и удобный пользовательский интерфейс. Клиентская часть содержит следующие элементы:

1. Стартовая страница:

При открытии приложения открывается модуль page1.py, который содержит описание проекта, описание данных и участников.

2. Навигация:

В левом сайдбаре реализовано меню выбора источника данных для последующего анализа: "Из базы данных” или “Загрузка CSV-файла”.

После выбора источника данных, вызывается функция навигации, которая перенаправляет пользователя на выбранную страницу.
Для данных, полученных из базы, используется функция navigation(flag), для загруженных CSV – функция navigation_csv(flag).
Загрузка данных:

При выборе “Из базы данных” флаг flag устанавливается в 1, и данные загружаются через вызов метода API /get_price_table.
При выборе “Загрузка CSV-файла” пользователю предоставляется файловый загрузчик. При успешной загрузке CSV-файла, вызывается функция database_csv.run(), которая обрабатывает файл и устанавливает флаг flag в 2. Если файл не содержит обязательные поля - выводится сообщение об ошибке.

3. Анализ данных:

После загрузки данных происходит запуск модуля EDA_page.py, который отвечает за отображение графиков и статистического анализа. Модуль предоставляет функциональность для выбора тикера, дат отображения данныхи,
модели машинного обучения, периода предсказания, технический индикаторов временного ряда.
Данная страница предоставляет средства для визуализации: графики временных рядов, свечной график, гистограммы распределения выбранных признаков корреляционные матрицы) и статистические данные.
В случае загрузки данных из csv - файла пользователю предоставляется возможность обучения модели по загруженному датасету и выбранному набору гиперпараметров

4. Анализ новостей:

Модуль news.py отвечает за визуализацию и анализ новостной информации. Позволяет пользователю исследовать тональность новостей.

Логика работы:
Главный модуль main() управляет загрузкой данных и выбором страницы. При запуске main(), Streamlit создает пользовательский интерфейс.
В зависимости от выбранного пользователем варианта загрузки устанавливается значение флага flag. Флаг flag определяет, какая функция навигации будет вызвана: navigation(flag) для данных из базы, или navigation_csv(flag) для загруженных CSV.
Функции навигации (navigation, navigation_csv) импортируют и запускают соответствующие модули страниц (EDA_page.py, news.py).

Пользовательский опыт:

Интерфейс построен интуитивно понятно, с использованием боковой панели для навигации и выбора опций загрузки.
Streamlit обеспечивает интерактивность, позволяя пользователю взаимодействовать с графиками и выбирать параметры анализа.
Предупреждения и информационные сообщения помогают пользователю правильно использовать сервис.
Клиентская часть обеспечивает удобство использования и гибкость для различных вариантов анализа данных.

Текущие результаты

- Построен и развернут сервис, позволяющий:
    - Проводить анализ временных рядов и текстовой информации, включая визуализацию графиков и расчёт статистик.
    - Вычислять и отображать технические индикаторы для различных активов.
    - Получать табличные данные о ценах и тональности новостей для детального анализа.
    - Генерировать прогнозы на основе предварительно обученных моделей.
    - Обучать новые модели на основе загруженнх данных и выбранных гиперпараметров.
    - Визуализировать результаты модели:
        - Сравнение фактических и предсказанных значений.

    - Остаточный анализ (распределение ошибок, Q-Q графики).

Особенности реализации

- Возможность работы с внешними пользовательскими данными, что делает сервис универсальным.
- Гибкость в выборе временных горизонтов для анализа и прогнозирования.
- Лёгкость в использовании за счёт интуитивно понятного интерфейса на основе Streamlit.
- Добавлена система логирования событий .