Задание 1: построение ML-системы обработки потоковых данных

Кафедра интеллектуальных информационных технологий,
курс «Технологическая практика (Практикум на ЭВМ)»

16 февраля 2025

1 Введение

В рамках данного курса необходимо разработать MLOps конвейер для автоматического развертывания ML-моделей в независимой продуктовой среде. Разработанный конвейер должен быть применим к потоковым данным, обеспечивать процессы CI/CD и использовать контейнеризацию (Docker).

Построение решения разделено на 2 этапа:

1. (Задание 1) Разработка MVP (Minimum Viable Product) - программы непрерывного построения моделей машинного обучения на потоковых данных без использования сторонних технологий. **Срок сдачи: 10.04, 23:59:59**

2. (Задание 2) Доработка и развёртывание разработанного MVP с использованием сторонних технологий (docker, CI/CD и др.). **Срок сдачи: 10.05, 23:59:59**

Задания будут выданы с пересечением сроков сдачи для демонстрации возможностей CI/CD при изменяющейся программной реализации ML-системы. К сроку сдачи программа по Заданию 1 должна быть полностью работоспособной, однако может включать не весь планируемый функционал. В ходе сдачи Задания 2 функционал должен быть доработан (с учетом правок после проверки Задания 1).

2 Постановка задачи

Требуется реализовать MVP (Minimum Viable Product) – программу на языке Python, которая будет эмулировать работу системы ML на потоковых табличных данных. Система должна обеспечивать выполнение следующих этапов:

1. **Сбор данных**. Функционал: сбор потоковых данных (сбор по батчам), работа с хранилищем сырых данных, расчет метапараметров;

2. **Анализ данных**. Функционал: оценка качества данных (Data Quality), автоматический EDA, контроль выполнения критериев качества, очистка данных.

3. **Подготовка данных**. Функционал: зависит от выбранной модели;

4. **Обучение/дообучение модели**. Функционал: построение моделей ML (LR, kNN или дерево решений).

5. **Валидация модели**. Функционал: оценка качества модели, отбор гиперпараметров, интерпретация модели, работа с хранилищем моделей.

6. **Обслуживание модели**. Функционал: выбор и упаковка (сериализация) продуктовой модели (или нескольких), оценка производительности, мониторинг, логирование.

Целью программы является построение, оценка и мониторинг ML-системы, адаптируемой при работе с новыми порциями данных. Система осуществляет построение модели на первичных данных и последовательно дообучает модель на новых данных, параллельно обновляя хранилище поступивших сырых данных и построенных моделей. Программа должна поддерживать возможность: (1) построения прогнозов на новых данных (с помощью лучшей модели), (2) дообучения на новых порциях данных (по запросу), (3) формирования отчета об изменении качества данных, модели, отобранных гиперпараметров и производительности модели.

При разработке программы запрещается использовать специализированное ПО, содержащее встроенные реализации этапов MLOps (MLFlow, AirFlow, MLRun, Metaflow и т.д.). Реализация должна проводиться на основе базовых средств и классических библиотек Python (включая Scikit-learn (и её расширения), statsmodels, Pandas, Numpy, Pickle и т.д.). При возникновении сомнений по использованию специализированных библиотек рекомендуется обратиться к преподавателю.

3 Формат сдачи

Реализованный проект оформляется в GitHub-репозитории (можно разрабатывать приватно и открыть доступ проверяющему по почте). Репозиторий должен содержать файлы README, requirements и набор директорий для каждого из этапов конвейера.

Задание возможно выполнять в одиночку или в команде (до 4-х человек). При выполнении задания в одиночку необходимо реализовать все компоненты системы самостоятельно (оценка выставляется по сумме баллов). При выполнении задания в команде каждый участник выбирает определенный набор этапов работы системы, на основе которых и будет выставляться оценка (один этап не может быть разделен между двумя и более участниками команды).

До 24.02 необходимо прислать в общий чат курса состав команды (в формате списка: ФИО, ссылка на telegram через @), набор данных (предварительно согласовав с преподавателем) и список разбиения команды по этапам (номер этапа – ФИО).

По реализованной системе готовится отчет (документация), отражающая используемые данные и полный пайплайн работы системы. Команда готовит общий отчет (документацию) и проводит презентацию системы на одном из семинаров.

4 Сценарии использования (интерфейс работы с программой)

Для управления разработанной программой используются консольные команды обращения к python скрипту. В частности, необходима обработка следующих команд:

- **Inference**: применение итоговой обученной модели к внешним данным.
  Пример: python run.py -mode “inference” -file “./path_to.file”
  Возвращаемое значение: путь до сохраненного DataFrame с дополнительной колонкой “predict” (отклик модели).

- **Update**: запуск дообучения модели на обновленных данных.
  Пример: python run.py -mode “update”
  Возвращаемое значение: bool (успешное или неуспешное прохождение всех этапов конвейера).

- **Summary**: подготовка отчета (в любом виде) об изменении (во времени) качества данных, метрик лучшей модели, отобранных гиперпараметров и производительности модели.
  Пример: python run.py -mode “summary”
  Возвращаемое значение: путь до файла с отчетом мониторинга.

5 Выбор набора данных

Для реализации системы возможно использовать учебные или собственные наборы данных. Учебным набором данных является набор Ethiopian Insurance Corporation, который рассматривался в осеннем семестре курса “Технологическая практика”. Возможно и использование собственных данных по научной и исследовательской деятельности. Для поиска наборов рекомендуется воспользоваться базами данных открытых датасетов с платформ Kaggle, OpenML и т.д.

Ваши собранные данные должны удовлетворять следующим критериям:

1. Данные хранятся в файле формата .csv, .xls или .xlsx (подойдут данные из excel-таблицы);

2. Набор должен содержать одну временную переменную (для обеспечения потоковости данных);

3. В наборе должно быть не менее 10000 строк и 10 признаков (среди которых не менее 2-х категориальных);

4. В наборе должны быть пропущенные значения (можно предварительно выколоть некоторые значения случайным образом).

6 Критерии оценки

1. **Сбор данных (3–10 баллов):**

(a) Обязательная часть:

i. Функционал сбора потоковых данных: разделение исходного набора на батчи и эмуляция потока (1 балл);

ii. Разработка хранилища сырых данных: файловая система (1 балл) или БД (2 балла);

iii. Расчет метапараметров: (1-2 балла).

(b) Дополнительные баллы:

i. Создание конфигурационного файла (.py или YAML/JSON/TOML/XML) с гиперпараметрами сбора (1 балл);

ii. Интеграция с несколькими источниками данных (2 балла);

iii. Система логирования (с использованием библиотек или вручную) и обработки ошибок при сборе данных (1-2 балла).

2. **Анализ данных (2–10 баллов):**

(a) Обязательная часть:

i. Оценка и хранение показателей качества данных (data quality) (1-2 балла);

ii. Базовая очистка данных на основе порогов допустимых значений качества (1 балл).

(b) Дополнительные баллы:

i. Автоматический EDA (1-2 балла);

ii. Добавление Feature Engineering (1-2 балла);

iii. Генерация отчетов о качестве данных (1 балла);

iv. Мониторинг и обработка ситуаций data drift (1-2 балл).

3. **Подготовка данных (входит в pipeline построения модели) (0–5 баллов):**

(a) Обязательная часть зависит от используемой модели ML:

i. Обработка пропусков (0-1 балл);

ii. Обработка категориальных переменных (0-1 балл);

iii. Обработка числовых переменных (0-1 балл).

(b) Дополнительные баллы:

i. Создание нескольких вариантов предобработки с дальнейшим перебором при поиске лучшей модели (1-2 балла).

4. **Обучение/дообучение модели (входит в pipeline построения модели) (1–5 баллов):**

(a) Обязательная часть: построение модели (LR, kNN или дерево решений) (1 балл).

(b) Дополнительные баллы:

i. Реализация дообучения предыдущей модели (без обучения модели с нуля) (1-2 балла);

ii. Разработка нескольких моделей с различной устойчивостью к входным данным (1-2 балла).

5. **Валидация модели (2–10 баллов):**

   (a) Обязательная часть:

      i. Оценка качества модели/моделей (hold-out/CV/TimeSeriesCV) (1-3 балла);

      ii. Разработка хранилища версий моделей и контроль качества (1-2 балла).

   (b) Дополнительные баллы:

      i. Интерпретация прогнозов (визуализация структуры дерева, оценка коэффициентов LR, демонстрация ближайших соседей, LIME, SHAP) (1-3 балла);

      ii. Мониторинг и обработка ситуаций model drift (1–2 балла).

6. **Обслуживание модели (1–6 баллов):**

   (a) Обязательная часть: выбор и упаковка (сериализация) финальной модели (или нескольких) (1-2 балла);

   (b) Дополнительные баллы:

      i. Мониторинг производительности (времени применения/памяти) (1-2 балла);

      ii. Обеспечение гибкого прогноза на основе данных (выбор модели при разреженных данных или аномальных значениях) (1-2 балла).

7. **Управление программой (3–11 баллов):**

   (a) Обязательная часть:

      i. Создание скрипта управления конвейером и обработка запросов (Inference, Update, Summary) (2 балла);

      ii. Написание документации к программной реализации (README и requirements) (1 балл).

   (b) Дополнительные баллы:

      i. Построение расширенного отчета (dashboard) о работе системы (1-2 балла);

      ii. Создание конфигурационного файла параметров всех компонентов системы (размер батча, допустимое число пропусков, тип валидации и т.д.) (1-2 балла);

      iii. Создание Meta Learning модели: оценка влияющих гиперпараметров и динамика метрик качества моделей и данных (2 балла).

      iv. Построение архитектуры на основе паттернов проектирования (например, MVC (model/view/controller)) ПО (1-2 балла).

Максимальное количество баллов: 57