Задание 2: построение CI/CD workflow для ML-системы обработки потоковых данных

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

15 апреля 2025

1 Введение

Задание является прямым продолжением «Задания 1: построение ML-системы обработки потоковых данных». В рамках данного задания необходимо доработать разработанный MVP с использованием сторонних технологий и разработать CI/CD «рабочий процесс» (далее будем использовать англоязычный термин «workflow») для развертывания системы на платформе GitHub Actions.

Для разработки workflow CI/CD можно взять за основу один из трёх workflow GitHub проекта, представленного на семинарских занятиях:

1. **Базовый сценарий:** CI/CD процесс отвечает за единоразовый независимый запуск модели.
   Артефакт: текстовый файл логов.
   Итерации обучения: задаются в скрипте запуска системы.

2. **Итеративное независимое обучение:** CI/CD процесс отвечает за единоразовый запуск обучения модели с заданным числом итераций.
   Артефакт: текстовый файл логов.
   Итерации обучения: задаются в YAML.

3. **CRON-инкрементальное обучение:** CI/CD процесс отвечает за последовательный запуск дообучения модели на новом батче данных по расписанию.
   Артефакт: текстовый файл логов, файл серилизованной модели, файл сериализованного сборщика данных.
   Итерации обучения: контролируются числом запуска задачи.

Стоит отметить, что **возможна реализация собственного CI/CD workflow**, если описанные выше сценарии не подходят к разработанной ML-системе. В таком случае дополнительные баллы будут обсуждаться с проверяющими.

Альтернативное решение: также, данное задание можно выполнить с помощью чистого Docker на локальном устройстве (необходимо написать DOCKERFILE). В таком случае под артефактами понимаются файлы, помещаемые в примонтированную директорию (доступные для просмотра без обращения к докеру). Реализация пунктов, требующих использования GitHub Actions, при таком подходе невозможна

и оцениваться не будет (оцениваются только пункты, согласованные с возможностью локального Docker).

Срок сдачи: 10.05, 23:59:59

2 Формат сдачи

Реализованный проект оформляется в GitHub-репозитории. Репозиторий должен содержать файлы README, requirements и YAML файлы описания workflow.

При работе в команде выполнение обязательных/дополнительных пунктов распределяется между студентами самостоятельно так, чтобы каждый пункт выполнял ровно один студент (будет проверяться на основе коммитов в репозиторий).

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

3 Критерии оценки

1. Функциональность workflow (3-8 баллов):
   (a) Обязательная часть:
      i. Автоматическое выполнение при push/pull request (1 балл);
      ii. Корректная установка окружения (1 балл);
      iii. Успешное обучение модели (1 балл);
   (b) Дополнительная часть:
      i. Управление инкрементальным обучением изнутри YAML-файла (1 балл);
      ii. Настройка CI/CD с инкрементальным дообучением на основе предыдущих запусков по расписанию (3 балла);
      iii. Добавление тестов ML-системы с помощью средств CI/CD (1 балл);
      iv. Иное расширение workflow (интеграция с базами данных, запуск паралелльных jobs, и т.д.);
2. Управлением артефактами (1-6 баллов):
   (a) Обязательная часть: Сохранение логов обучения моделей (1 балл);
   (b) Дополнительная часть:
      i. Сохранение сериализованной модели, допускающей запуск извне (1 балл);
      ii. Сохранение модели сборщика данных, допускающего загрузку актуальных данных извне (1 балл);
      iii. Сохранение dashboard истории обучения или запуск онлайн-сервиса на платформе GitHub Actions (3 балла);
3. Документирование проекта. Обязательная часть (2 балла):

(a) Добавление раздела в README с инструкциями по развертыванию (1 балл);

(b) Текстовое описание и аргументация выбранного подхода (1 балл);

Максимальное количество баллов: 16