

Министерство науки и высшего образования Российской Федерации

Московский физико-технический институт

Промышленное машинное обучение

Осень

2022

Лабораторная работа №1

КЛАССИЧЕСКИЙ ЖИЗНЕННЫЙ ЦИКЛ РАЗРАБОТКИ МОДЕЛЕЙ МАШИННОГО ОБУЧЕНИЯ

Цель работы:

Получить навыки разработки CI/CD pipeline для ML моделей с достижением метрик моделей и качества.

Ход работы:

1. Создать репозитории модели на GitHub, регулярно проводить commit + push в ветку разработки, важна история коммитов;
2. Провести подготовку данных для набора данных, согласно варианту задания;
3. Разработать ML модель с ЛЮБЫМ классическим алгоритмом классификации, кластеризации, регрессии и т. д.;
4. Конвертировать модель из *.ipynb в .py скрипты;
5. Покрыть код тестами, используя любой фреймворк/библиотеку;
6. Задействовать DVC;
7. Использовать Docker для создания docker image.
8. Наполнить дистрибутив конфигурационными файлами:
 - config.ini: гиперпараметры модели;
 - Dockerfile и docker-compose.yml: конфигурация создания контейнера и образа модели;

- requirements.txt: используемые зависимости (библиотеки) и их версии;
 - dev_sec_ops.yml: подписи docker образа, хэш последних 5 коммитов в репозитории модели, степень покрытия тестами;
 - scenario.json: сценарии тестирования запущенного контейнера модели.
9. Создать CI pipeline (Jenkins, Team City, Circle CI и др.) для сборки docker image и отправки его на DockerHub, сборка должна автоматически стартовать по pull request в основную ветку репозитория модели;
 10. Создать CD pipeline для запуска контейнера и проведения функционального тестирования по сценарию, запуск должен стартовать по требованию или расписанию или как вызов с последнего этапа CI pipeline;
 11. Результаты функционального тестирования и скрипты конфигурации CI/CD pipeline приложить к отчёту.

Результаты работы:

1. Отчёт о проделанной работе;
2. Ссылка на репозиторий GitHub;
3. Ссылка на docker image в DockerHub;
4. Актуальный дистрибутив модели в zip архиве.

Обязательно обернуть модель в контейнер (этап CI) и запустить тесты внутри контейнера (этап CD). **Дополнительно** – настроить веб сервер в отдельном контейнере (Apache/nginx + Flask/Django) для обработки запросов к модели в режиме реального времени. Выполнение дополнительного условия гарантирует четверть количества баллов за контрольный семинар, достаточного для оценки "отлично". Таким образом, выполнение дополнительного задания для каждой работы (их 4) даст автоматически оценку "отлично" за курс.

Варианты задания

Номер	Набор данных
1	https://www.kaggle.com/parulpandey/palmer-archipelago-antarctica-penguin-data
2	https://www.kaggle.com/c/bike-sharing-demand
3	https://archive.ics.uci.edu/ml/datasets/Wine+Quality
4	https://www.kaggle.com/c/boston-housing
5	https://archive.ics.uci.edu/ml/datasets/Ionosphere
6	https://www.kaggle.com/zalando-research/fashionmnist
7	https://www.kaggle.com/c/dogs-vs-cats
8	https://www.kaggle.com/uciml/breast-cancer-wisconsin-data
9	https://www.kaggle.com/c/twitter-sentiment-analysis2
10	https://www.kaggle.com/c/learn-ai-bbc
11	https://www.kaggle.com/uciml/sms-spam-collection-dataset
12	http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html
13	https://arxiv.org/abs/1609.08675 http://research.google.com/youtube8m/

14	https://jmcauley.ucsd.edu/data/amazon/
15	https://www.kaggle.com/ritesaluja/bank-note-authentication-uci-data
16	http://labelme.csail.mit.edu/Release3.0/index.php
17	https://archive.ics.uci.edu/ml/datasets/Connectionist+Bench+(Sonar,+Mines+vs.+Rocks)
18	https://www.kaggle.com/uciml/pima-indians-diabetes-database
19	https://www.kaggle.com/jmcaro/wheat-seedsuci
20	https://www.kaggle.com/tunguz/200000-jeopardy-questions
21	https://www.kaggle.com/rodolfomendes/abalone-dataset
22	https://www.kaggle.com/c/fake-news/overview
23	https://image-net.org/