Открыт приём заявок!

Третье ДЗ

Машинное обучение в продакшене

Суть этого ДЗ -- познакомиться с **airflow.**

Легенда:
1) Откуда-то берутся данные... Мы их используем для обучения МЛ модельки для задачи классификации.
2) Еженедельно, мы переобучаем модель на новых данных, ручками смотрим на метрики и если класс, то выкатываем ее на **прод.**
3) Ежедневно, текущая, выбранная нами модель, скорит данные и записывает предсказания куда-то
4) Эти предсказания используют -- все счастливы

В ДЗ предлагается на основе airflow реализовать описанную выше схему, к деталям:

0) Поднимите airflow локально, используя docker compose (можно использовать из примера https://github.com/made-ml-in-prod-2021/airflow-examples/)
1) (**5 баллов**) Реализуйте dag, который генерирует данные для обучения модели (генерируйте данные, можете использовать как генератор синтетики из первой дз, так и что-то из датасетов sklearn), вам важно проэмулировать ситуации постоянно поступающих данных
- записывайте данные в /data/raw/{{ ds }}/data.csv, /data/raw/{{ ds }}/target.csv

2) (**10 баллов**) Реализуйте dag, который обучает модель еженедельно, используя данные за текущий день. В вашем пайплайне должно быть как минимум 4 стадии, но дайте волю своей фантазии=)

- подготовить данные для обучения(например, считать из /data/raw/{{ ds }} и положить /data/processed/{{ ds }}/train_data.csv)
- расплитить их на train/val
- обучить модель на train (сохранить в /data/models/{{ ds }}
- провалидировать модель на val (сохранить метрики к модельке)

3) Реализуйте dag, который использует модель ежедневно (**5 баллов**)
- принимает на вход данные из пункта 1 (data.csv)
- считывает путь до модельки из airflow variables(идея в том, что когда нам нравится другая модель и мы хотим ее на прод
- делает предсказание и записывает их в /data/predictions/{{ds }}/predictions.csv

3а) Реализуйте **сенсоры** на то, что данные готовы для дагов тренировки и обучения (3 **доп балла)**

4) вы можете выбрать 2 пути для выполнения ДЗ.
- поставить все необходимые пакеты в образ с airflow и использовать bash operator, python operator (**0 баллов**)
- использовать DockerOperator, тогда выполнение каждой из тасок должно запускаться в собственном контейнере
-- 1 из дагов реализован с помощью DockerOperator (**5 баллов)**
-- все даги реализованы **только** с помощью DockerOperator (**10 баллов**) (пример

https://github.com/made-ml-in-prod-2021/airflow-examples/blob/main/dags/11_docker.py).
По технике, вы можете использовать такую же структуру как в примере, пакую в разные докеры скрипты, можете использовать общий докер с вашим пакетом, но с разными точками входа для разных тасок.
Прикольно, если вы покажете, что для разных тасок можно использовать разный набор зависимостей.

https://github.com/made-ml-in-prod-2021/airflow-examples/blob/main/dags/11_docker.py#L27 в этом месте пробрасывается путь с хостовой машины, используйте здесь путь типа /tmp или считывайте из переменных окружения.

5) Протестируйте ваши даги (5 **баллов**) https://airflow.apache.org/docs/apache-airflow/stable/best-practices.html
6) В docker compose так же настройте поднятие mlflow и запишите туда параметры обучения, метрики и артефакт(модель) (**5 доп баллов)**
7) вместо пути в airflow variables используйте апи Mlflow Model Registry (**5 доп баллов**)
Даг для инференса подхватывает последнюю продакшен модель.
8) Настройте alert в случае падения дага (**3 доп. балла**)
https://www.astronomer.io/guides/error-notifications-in-airflow
9) традиционно, самооценка (**1 балл**)

Чтобы получить баллы сделайте скриншоты списка всех дагов и каждого графа(в исполненном без ошибок виде) по отдельности и прикрепите их в описание пулл реквеста.

Весь код, относящийся к данному ДЗ должен лежать в папке airflow_ml_dags ,ветка homework3, label hw3

Дедлайны:
soft: 7 июня
hard: 14 июня (-40%)

Заполняем опрос по вероятно просмотренной паре: https://forms.gle/JgUim6xfjCf6T6GW7

Норм?

- Очень сложно
- Сложно
- Норм
- Легко
- Слишком легко

Воздержаться