Отчёт 3 лабораторной работы по дисциплине “Инженерия данных”

выполнил Доружинский Дмитрий из группы 6233-010402D.

1. Пайплайн для инференса данных

Пайплайн для обучения классификаторов

Построенный пайплайн будет выполнять следующие действия поочередно:

- Производить мониторинг целевой папки на предмет появления новых конфигурационных файлов классификаторов (в форматах .json или .yaml).
- Обучать классификатор в соответствии с полученными параметрами.
- Производить логгирование параметров модели в MLflow.
- Производить логгирование процесса обучения MLflow.
- Производить тестирование модели и сохранять его результаты в MLflow.
- Сохранять обученный классификатор в model registry MLflow.

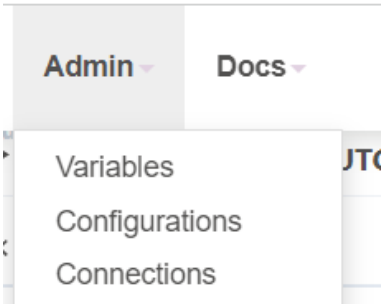
2. Пайплайн для хостинга лучшей модейли

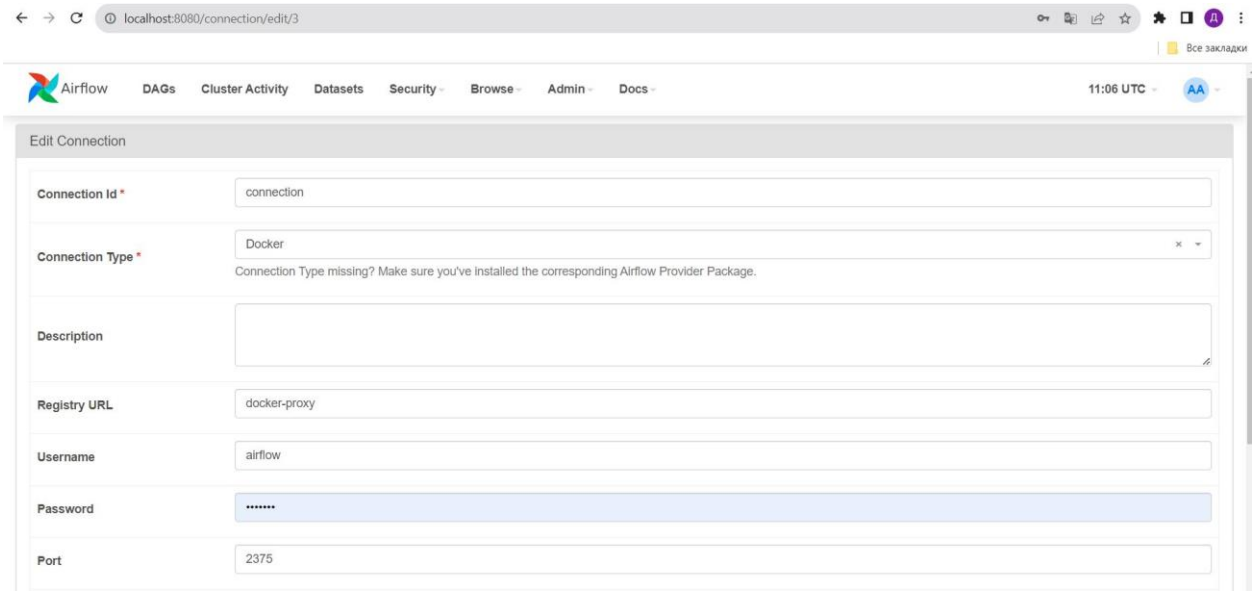
- Построенный пайплайн будет выполнять следующие действия поочередно:
- В соответствии с таймером производит валидацию новых моделей из model registry.
- Модель с лучшим показателем метрики переводится на stage: Production
- (Опционально) произвести хостинг лучшей модели

В начале необходимо установить подключение к airflow.

```
monitoring = FileSensor( # загрузка файла config.json
    task_id = 'download_json', # название этапа
    poke_interval = 10, #  время, в течение которого задание должно ждать между каждой попыткой
    filepath = '/opt/airflow/data/config.json', # путь по которому происходит поиск файла и название файла
    fs_conn_id = 'connection', # название подключения, необходимо ввести в airflow
    dag = dag, # использование DAG
)
```

В параметре fs_conn_id необходимо указать имя, которое после необходимо указать при создание connections в выпадающем списке Admin графического интерфейса Airflow. Параметр poke_interval задает интервал времени в секундах обновления просмотра новых файлов





Для работы был использован dataset из библиотеки sklearn load_breast_cancer. Содержит набор данных о раке молочной железы. Подходит для двоичной классификации.

В качестве классификаторов были выбраны следующие:

- MLPClassifier;
- KNeighborsClassifier;
- GaussianProcess;
- RandomForest;
- AbaBoost.

Для упрощения работы был подготовлен config.json файл, содержит список классификаторов и их гипперпараметрами выбранными вручную.

Рисунок 1 – подготовленные файл config.json

Мониторинг для поиска файла config.json реализуется в следующем DAG.

```
monitoring = FileSensor( # загрузка файла config.json
    task_id = 'download_json', # название этапа
    poke_interval = 10, #  время, в течение которого задание должно ждать между каждой попыткой
    filepath = '/opt/airflow/data/config.json', # путь по которому происходит поиск файла и название файла
    fs_conn_id = 'connection', # название подключения, необходимо ввести в airflow
    dag = dag, # использование DAG
)
```

Рисунок 2 – DAG для мониторинга файла config.json

За обучение моделей отвечает следующий DAG.

```
train_data = BashOperator(
    task_id = "train_data_lab_3", # название задания
    bash_command = "python /opt/airflow/data/train_data_lab_3.py", # запуск файла python из данной директории
    dag = dag # использование DAG
)
```

Рисунок 3 – DAG для обучения моделей

Был подготовлен файл train_data_lab_3.py в котором происходит загрузка набора данных, его разбиение, обучение моделей, передачу информации в MlFlow о метриках таких как:

- Accuracy;
- F1;
- Precision;
- Recall.

Рисунок 4 – результат работы 1 части лабораторной работы

Для второй части был подготовлен файл запускающий файл validation_lab_3.py. В нём происходит проверка работы моделей и выбор лучшего по f1 метрике. Также происходит пометка в MlFlow лучшей модели.

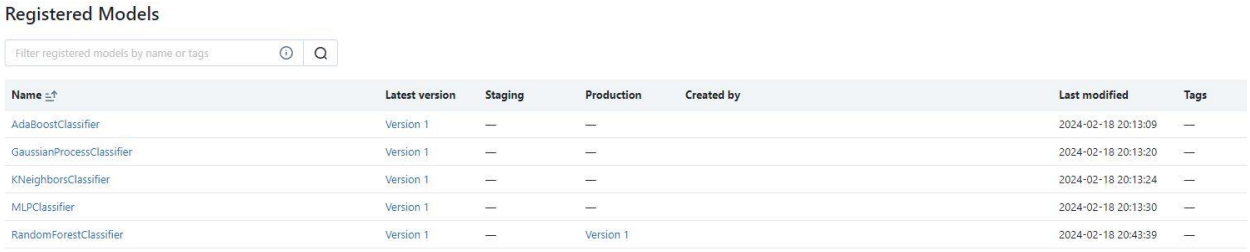


Рисунок 6 – результат работы 2-й части лабораторной работы.

ЗАКЛЮЧЕНИЕ

В результате выполнения лабораторной работы были получены навыки по работе с Docker, Apache Airflow, MLflow. В данной работе были построены два пайплайна – для обучения классификаторов и выбора лучшей модели.