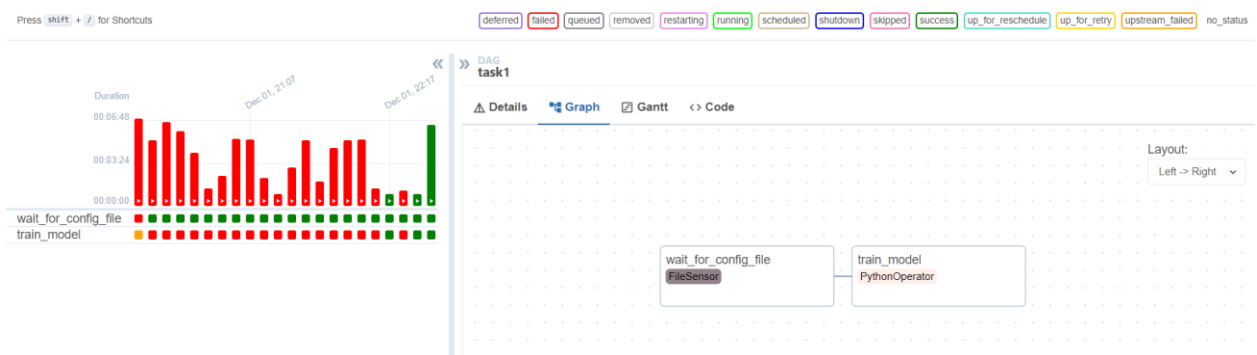


Лабораторная работа 3.

1. Пайплайн, который обучает любой классификатор из sklearn по заданному набору параметров.

Был реализован пайплайн вида `wait_for_config_file >> train_model_task`, граф которого представлен ниже.

Обучение модели проходило на данных из датасета mnist, как и во второй лабораторной работе. В следствии проблемы, с которой я тогда столкнулся (а именно большой размер данных), в этой лабораторной работе я, с вашего позволения, решил не загружать эти данные на github.



Граф пайплайна задания 1.

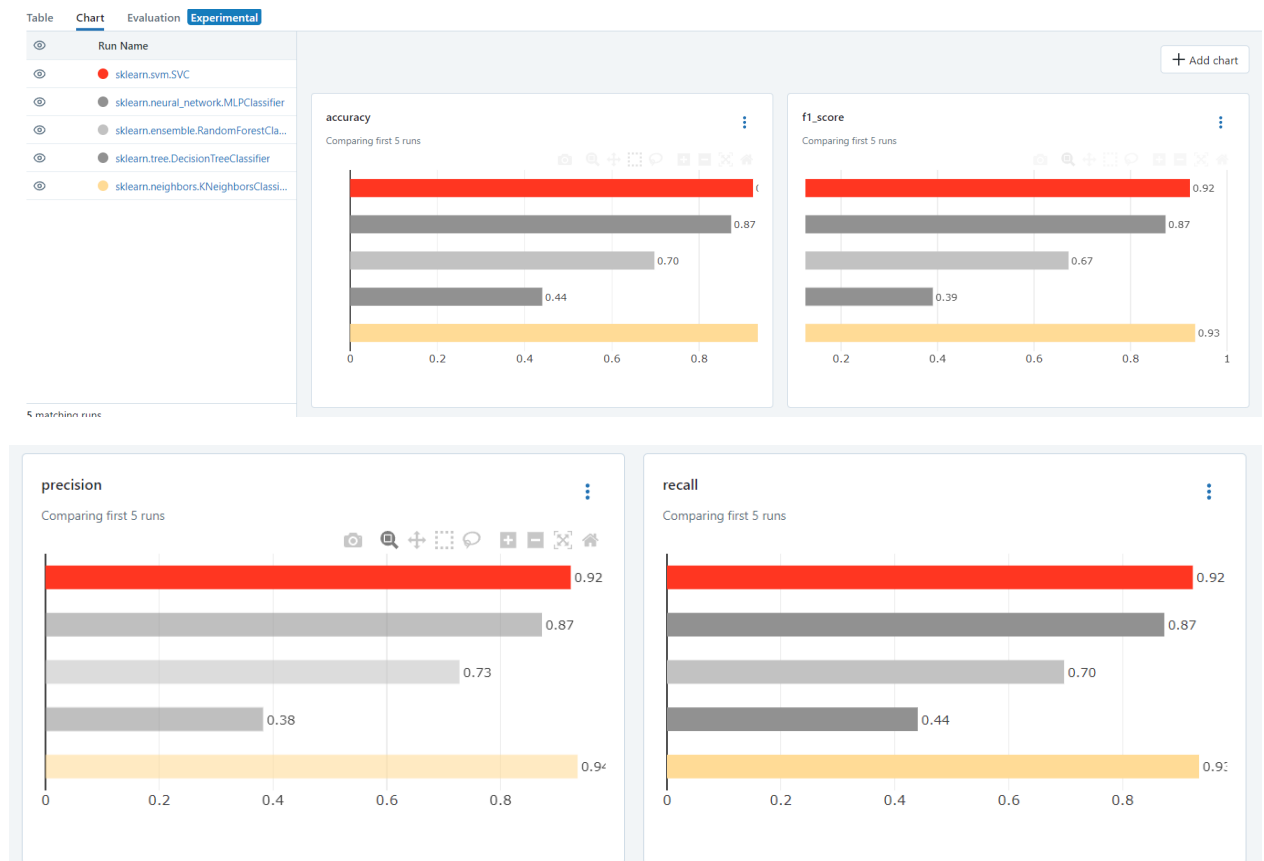
`FileSensor(wait_for_config_file)` сканирует папку и ожидает появления в ней json файла

`PythonOperator (train_model)` обучал классификаторы и отправлял значения метрик на mlflow.

Также был реализован сам json файл, содержащий информацию о классификаторах.

Основной сложностью для меня являлось правильное взаимодействие пайплайна с json файлом, почему-то в упор не видел ошибку, в следствии чего такое большое количество фэйлов в airflow.

На следующих рисунках представлены значения метрик из mlflow.



2. Пайплайн, который выбирает лучшую модель из обученных и производит её хостинг.

Был реализован пайплайн вида `validate_and_promote_models_task`

`PythonOperator (validate_and_promote_models _data)` сравнивает метрики моделей и выбирает из них лучшую на основе метрики `accuracy`.

Мне пришлось добавить проход по `experiment_name = "MNIST_Experiment"`, так как для теста я отдельно загружал набор данных и модель, которая при обучении дала `accuracy` 1 и поэтому, если бы я не добавил проход только по своему эксперименту, а реализовал бы проход по всем экспериментам, то эта модель всегда бы становилась лучшей.

На рисунке представлен лог пайплайна второго задания, который сообщает, какая модель является лучшей и какое значение её параметра

```
Promoted model sklearn.neighbors.KNeighborsClassifier version 3 to Production with metric value 0.9343809523809524
```


На следующем рисунке видно, что модель `KNeighborsClassifier` переведена в стадию `production`.

Registered Models

Filter registered models by name or tags

i

Q

Name 	Latest version	Staging	Production	Created by
sklearn.ensemble.RandomForestClassifier	Version 3	—	—	
sklearn.neighbors.KNeighborsClassifier	Version 3	—	Version 3	
sklearn.neural_network.MLPClassifier	Version 3	—	—	
sklearn.svm.SVC	Version 3	—	—	
sklearn.tree.DecisionTreeClassifier	Version 3	—	—	