БУЛДАКОВ ДМИТРИЙ

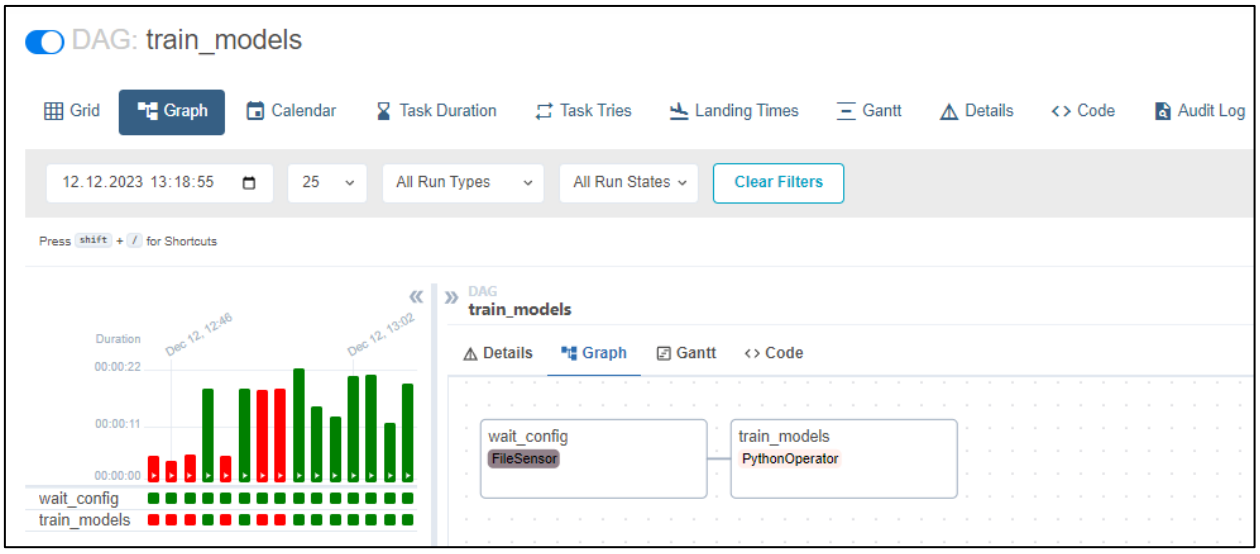
6233-010402D

ХОД РАБОТЫ

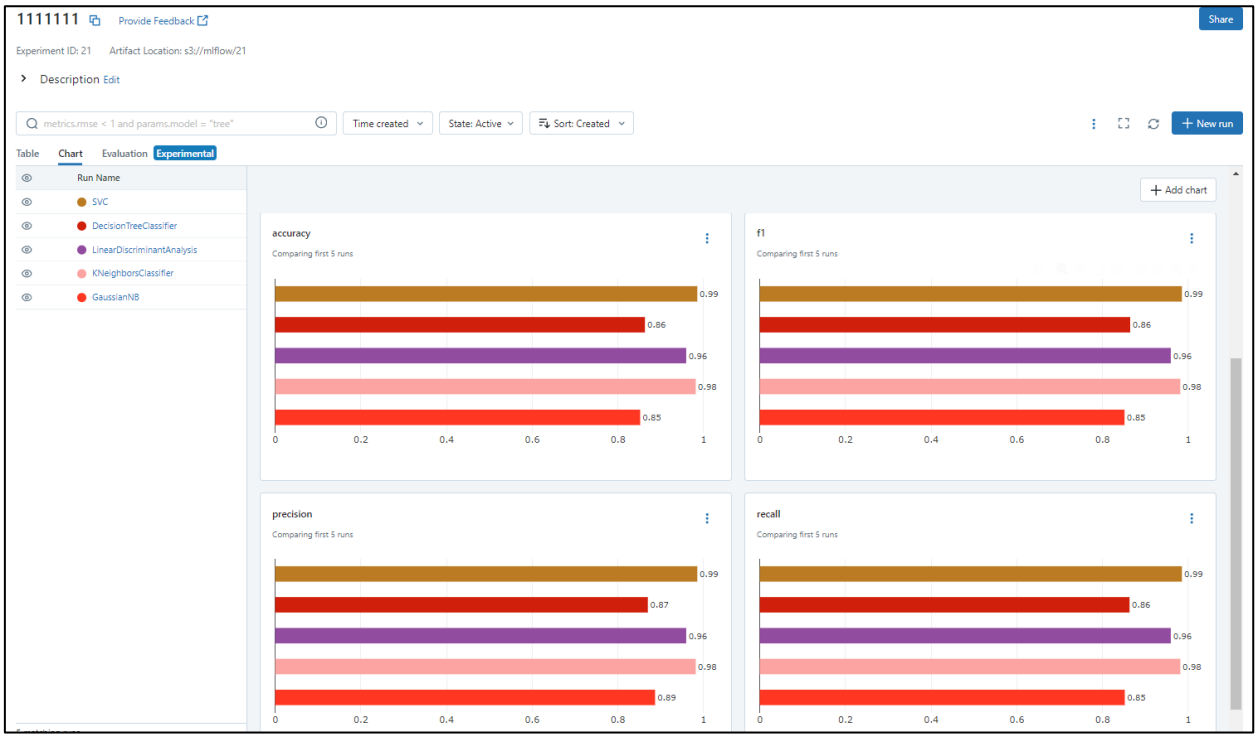
Пайплайн для обучения классификаторов

Код первого пайплайна можно посмотреть в файле train_models_DAG.py. Алгоритм работы следующий:

1. Проверяем наличие конфига с классификаторами в папке;
2. Разбираем конфиг, импортируем необходимые модули, указываем параметры для классификаторов;
3. Проводим обучение и тестируем классификатор
4. Логируем модели, метрики и параметры в mlflow



В mlflow отобразились наши модели, а также их метрики

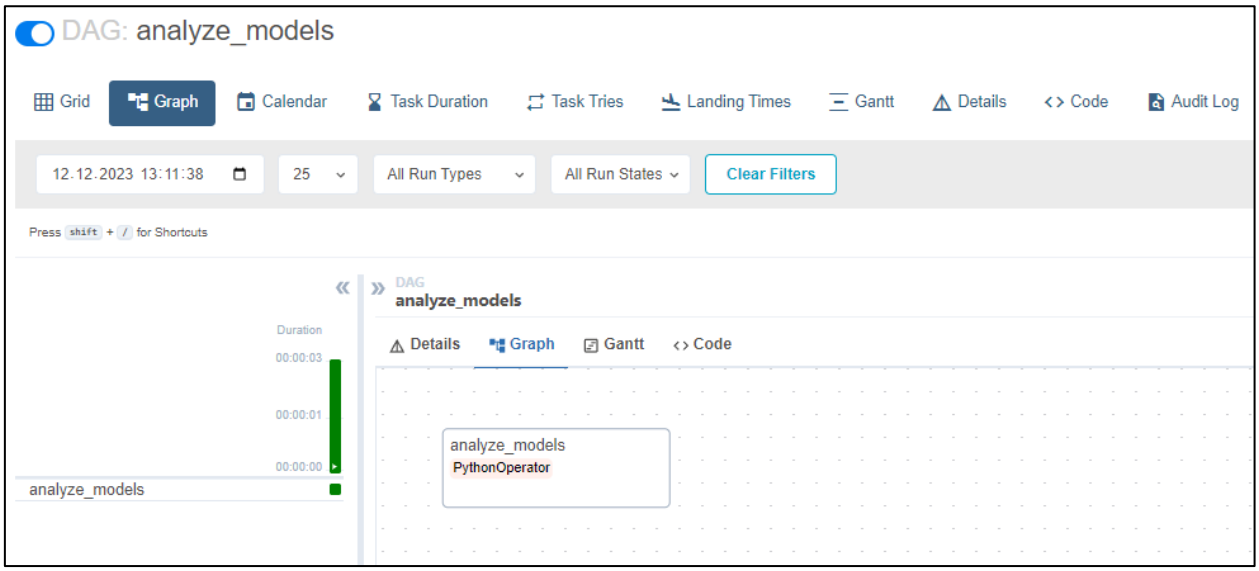

1111111
Provide Feedback
Share
Experiment ID: 21
Artifact Location: s3://mlflow/21
Description Edit
metrics.rmse < 1 and params.model = "tree"
Time created
State: Active
Sort: Created
New run
Table
Chart
Evaluation
Experimental
Run Name
SVC
DecisionTreeClassifier
LinearDiscriminantAnalysis
KNeighborsClassifier
GaussianNB
Add chart
accuracy
Comparing first 5 runs
0.99
0.86
0.96
0.98
0.85
f1
Comparing first 5 runs
0.99
0.86
0.96
0.98
0.85
precision
Comparing first 5 runs
0.99
0.87
0.96
0.98
0.89
recall
Comparing first 5 runs
0.99
0.86
0.96
0.98
0.85
0
0.2
0.4
0.6
0.8
1

Пайплайн для хостинга лучшей модели

В качестве основной метрики, по которой мы будем выбирать наилучшую модель была выбрана f1_score. Для второго пайплайна получился следующий код:

```python
import mlflow
import mlflow.sklearn
import pandas as pd
import numpy as np

from sklearn.metrics import f1_score
from mlflow import MlflowClient
from datetime import datetime
from airflow import DAG
from airflow.operators.python import PythonOperator

with DAG(
    'analyze_models',
    start_date=datetime(2023, 12, 12),
    schedule_interval=None,
    catchup=False
    ) as dag:

    def analyze_models():
        mlflow.set_tracking_uri('http://mlflow_server:5000')

        mlflow.set_experiment("1111111")

        X_val = np.asarray(pd.read_csv(f"/opt/airflow/data/X_val.csv"), dtype = np.float32)
        y_val = pd.read_csv(f"/opt/airflow/data/y_val.csv")

        list_models = {}
        mlflow.start_run(run_name = "Best model")
        models = pd.read_csv("/opt/airflow/data/res_models.csv", header = None)

        for model_info in models.iterrows():
            name_model = model_info[1][1]
            uri_model = model_info[1][2]
            list_models[name_model + " " + uri_model] = mlflow.sklearn.load_model(uri_model)

        results = {}
        for i, j in list_models.items():
            prediction = j.predict(X_val)
            results[i] = f1_score(y_val, prediction, average="weighted")

        best_model = max(results, key=results.get)

        version = MlflowClient().search_model_versions(f"name = '{best_model.split(' ')[0]}' and run_id = '{best_model.split(' ')[1].split('/')[1]}'")[0].version
        MlflowClient().transition_model_version_stage(name = best_model.split(' ')[0], version = version, stage = "Production")
        mlflow.end_run()

    analyze_models = PythonOperator(
        task_id = "analyze_models",
        python_callable=analyze_models,
        dag=dag
    )

    analyze_models
```

DAG-файл запущен успешно.



После отработки пайплайна в mlflow можно увидеть, что модель с наилучшим f1_score перешла в стадию Production.