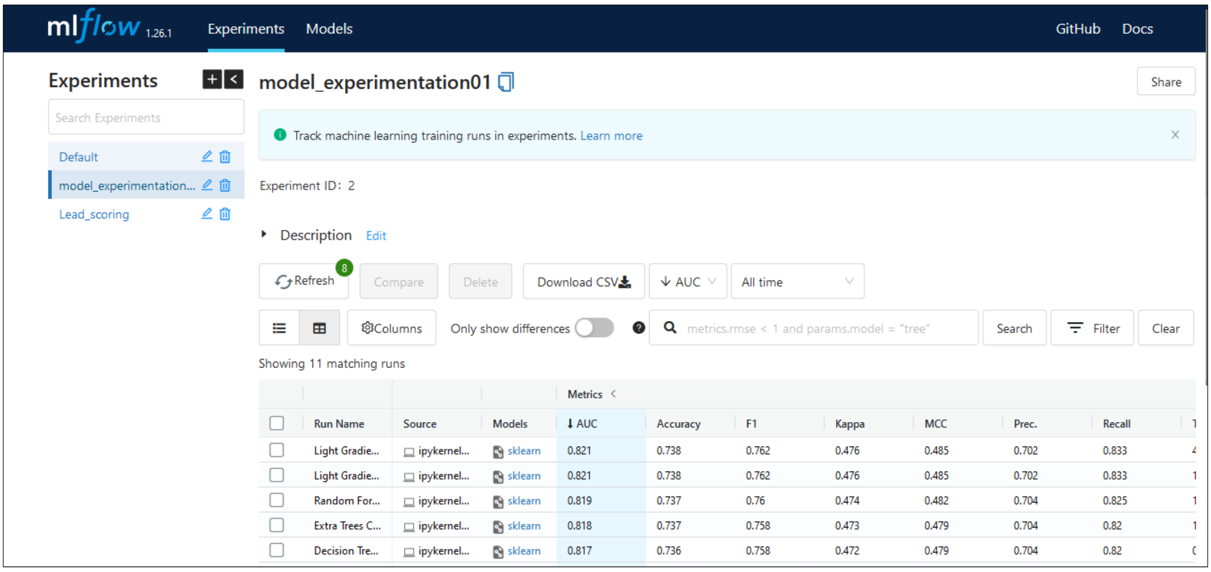
MODEL EXPERIMENTATION

1. SCREENSHOT OF MLFLOW UI BEFORE DROPPING FEATURES



model_experimentation01

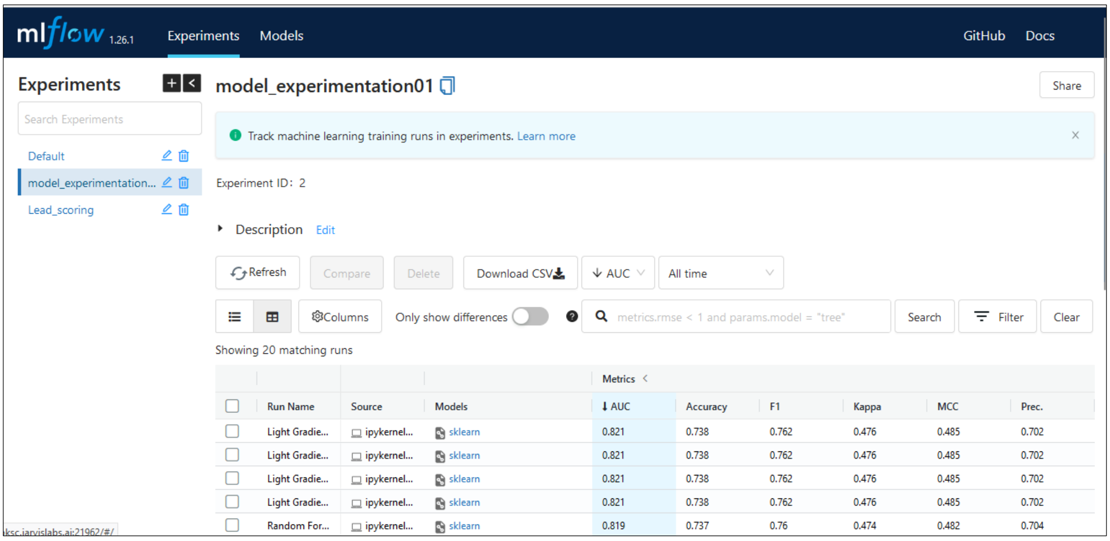
Track machine learning training runs in experiments. Learn more

Experiment ID: 2

Showing 11 matching runs

				Metrics							
□	Run Name	Source	Models	AUC	Accuracy	F1	Kappa	MCC	Prec.	Recall	T
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702	0.833	4
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702	0.833	1
□	Random For...	ipykernel...	sklearn	0.819	0.737	0.76	0.474	0.482	0.704	0.825	1
□	Extra Trees C...	ipykernel...	sklearn	0.818	0.737	0.758	0.473	0.479	0.704	0.82	1
□	Decision Tre...	ipykernel...	sklearn	0.817	0.736	0.758	0.472	0.479	0.704	0.82	(

a. SCREENSHOT OF ALL THE EXPERIMENTS



model_experimentation01

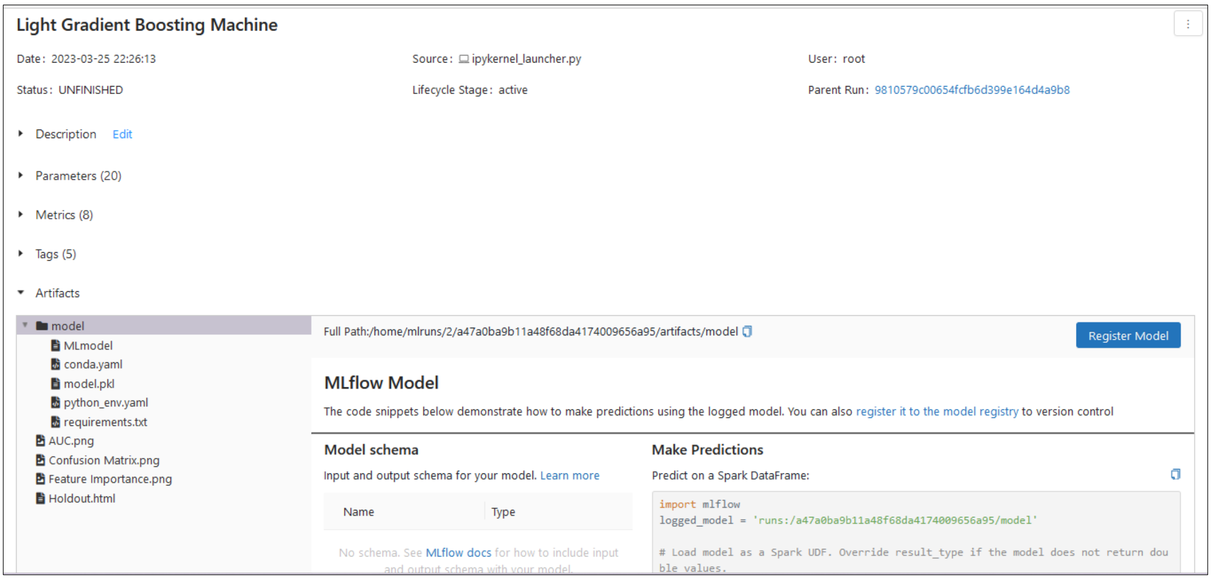
Track machine learning training runs in experiments. Learn more

Experiment ID: 2

Showing 20 matching runs

				Metrics					
□	Run Name	Source	Models	AUC	Accuracy	F1	Kappa	MCC	Prec.
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702
□	Light Gradie...	ipykernel...	sklearn	0.821	0.738	0.762	0.476	0.485	0.702
□	Random For...	ipykernel...	sklearn	0.819	0.737	0.76	0.474	0.482	0.704

b. SCREENSHOT OF ONE EXPERIMENTS WITH ALL THE ARTIFACTS VISIBLE



Light Gradient Boosting Machine

Date: 2023-03-25 22:26:13 | Source: ipykernel_launcher.py | User: root

Status: UNFINISHED | Lifecycle Stage: active | Parent Run: 9810579c00654fcfb6d399e164d4a9b8

Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

Model schema

Input and output schema for your model. Learn more

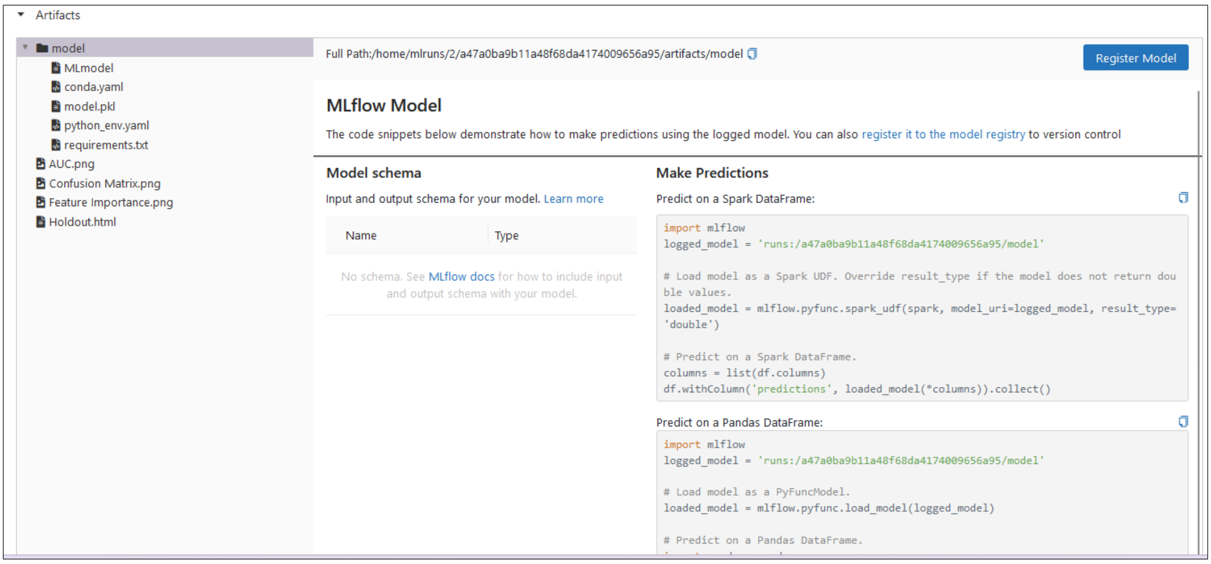
Name	Type

No schema. See MLflow docs for how to include input and output schema with your model.

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
logged_model = 'runs:/a47a0ba9b11a48f68da4174009656a95/model'

# Load model as a Spark UDF. Override result_type if the model does not return dou
ble values.
```



▾ Artifacts

model
MLmodel
conda.yaml
model.pkl
python_env.yaml
requirements.txt
AUC.png
Confusion Matrix.png
Feature Importance.png
Holdout.html

Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

Model schema

Input and output schema for your model. Learn more

Name | Type

No schema. See MLflow docs for how to include input and output schema with your model.

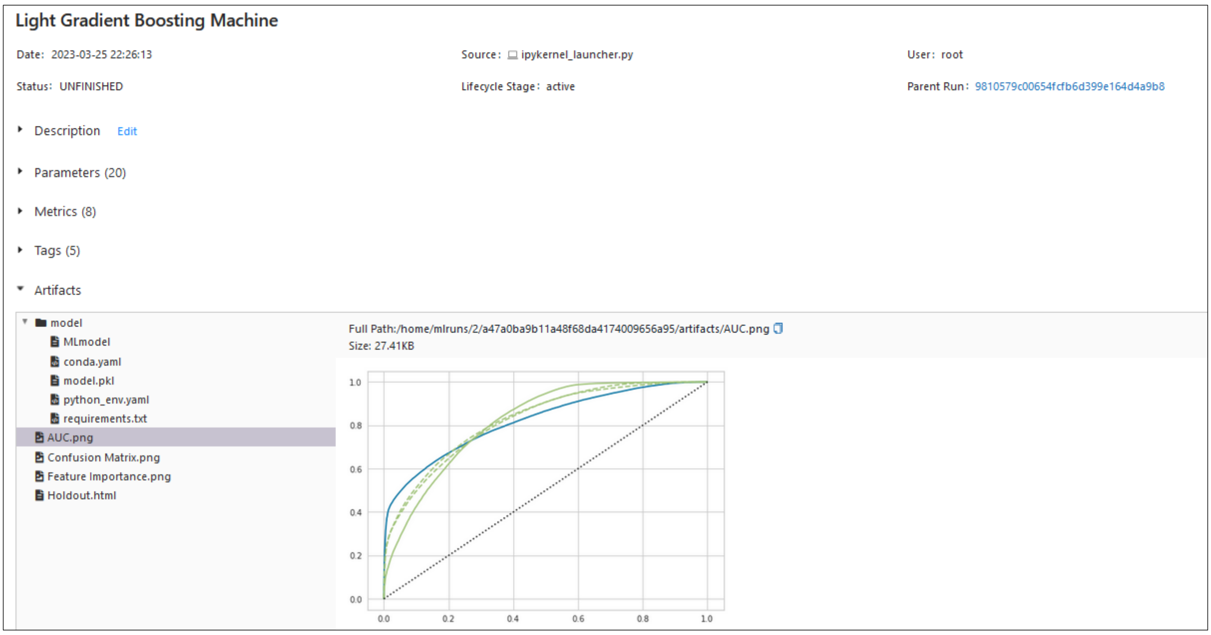
Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
logged_model = 'runs:/a47a0ba9b11a48f68da4174009656a95/model'

# Load model as a Spark UDF. Override result_type if the model does not return dou
ble values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type=
'double')

# Predict on a Spark DataFrame.
columns = list(df.columns)
df.withColumn('predictions', loaded_model(*columns)).collect()
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/a47a0ba9b11a48f68da4174009656a95/model'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
```



Light Gradient Boosting Machine

Date: 2023-03-25 22:26:13 | Source: ipykernel_launcher.py | User: root

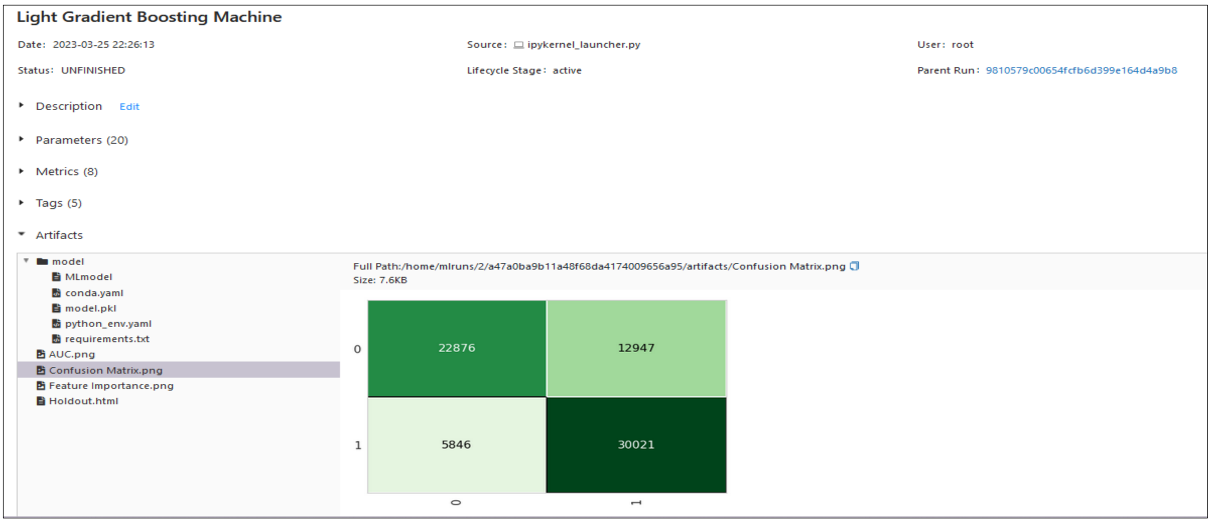
Status: UNFINISHED | Lifecycle Stage: active | Parent Run: 9810579c00654fcfb6d399e164d4a9b8

- Description Edit
- Parameters (20)
- Metrics (8)
- Tags (5)
- Artifacts

```
▾ model
    MLmodel
    conda.yaml
    model.pkl
    python_env.yaml
    requirements.txt
  AUC.png
  Confusion Matrix.png
  Feature Importance.png
  Holdout.html
```

Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/AUC.png

Size: 27.41KB



Light Gradient Boosting Machine

Date: 2023-03-25 22:26:13 | Source: ipykernel_launcher.py | User: root

Status: UNFINISHED | Lifecycle Stage: active | Parent Run: 9810579c00654fcfb6d399e164d4a9b8

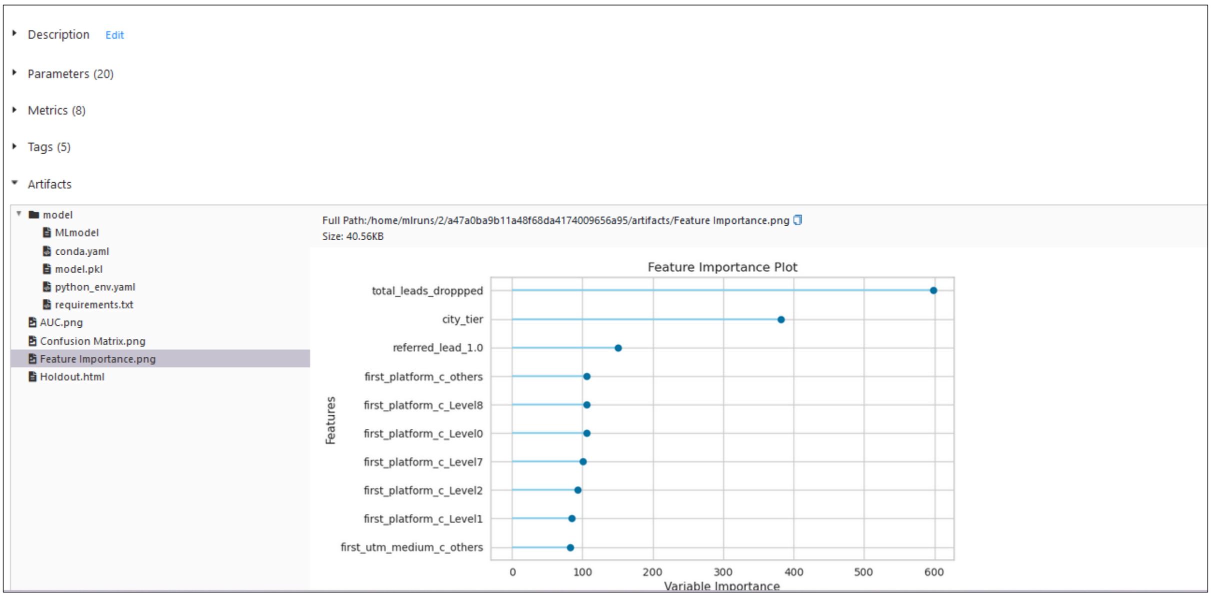
- Description Edit
- Parameters (20)
- Metrics (8)
- Tags (5)
- Artifacts

```
▾ model
    MLmodel
    conda.yaml
    model.pkl
    python_env.yaml
    requirements.txt
  AUC.png
  Confusion Matrix.png
  Feature Importance.png
  Holdout.html
```

Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/Confusion Matrix.png

Size: 7.6KB

	0	1
0	22876	12947
1	5846	30021

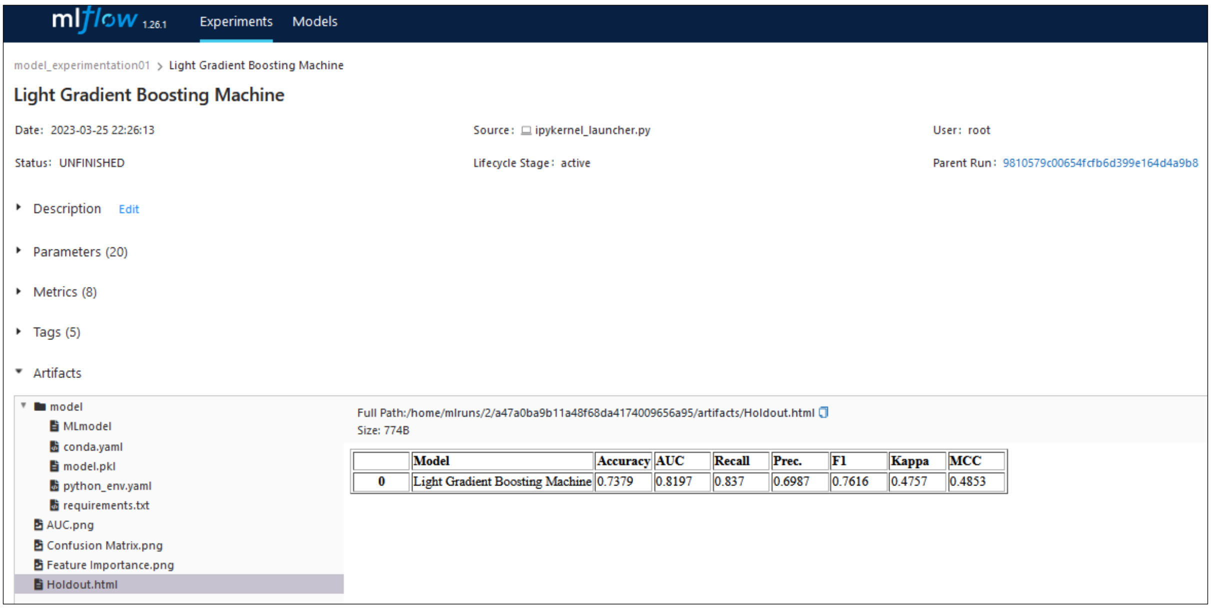


- Description Edit
- Parameters (20)
- Metrics (8)
- Tags (5)
- Artifacts

```
▾ model
    MLmodel
    conda.yaml
    model.pkl
    python_env.yaml
    requirements.txt
  AUC.png
  Confusion Matrix.png
  Feature Importance.png
  Holdout.html
```

Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/Feature Importance.png

Size: 40.56KB



Light Gradient Boosting Machine

Date: 2023-03-25 22:26:13

Source: ipykernel_launcher.py

User: root

Status: UNFINISHED

Lifecycle Stage: active

Parent Run: 9810579c00654fcfb6d399e164d4a9b8

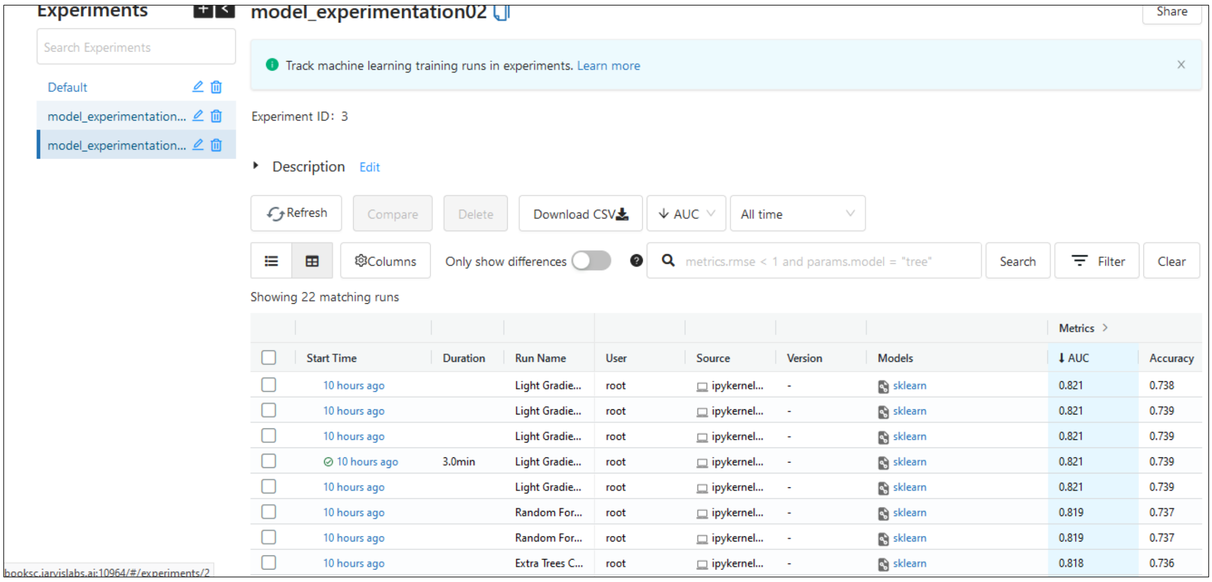
Full Path:/home/mlruns/2/a47a0ba9b11a48f68da4174009656a95/artifacts/Holdout.html

Size: 774B

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.7379	0.8197	0.837	0.6987	0.7616	0.4757	0.4853

2. SCREENSHOT OF MLFLOW UI AFTER DROPPING FEATURES

a. SCREENSHOT OF ALL THE EXPERIMENTS



model_experimentation02

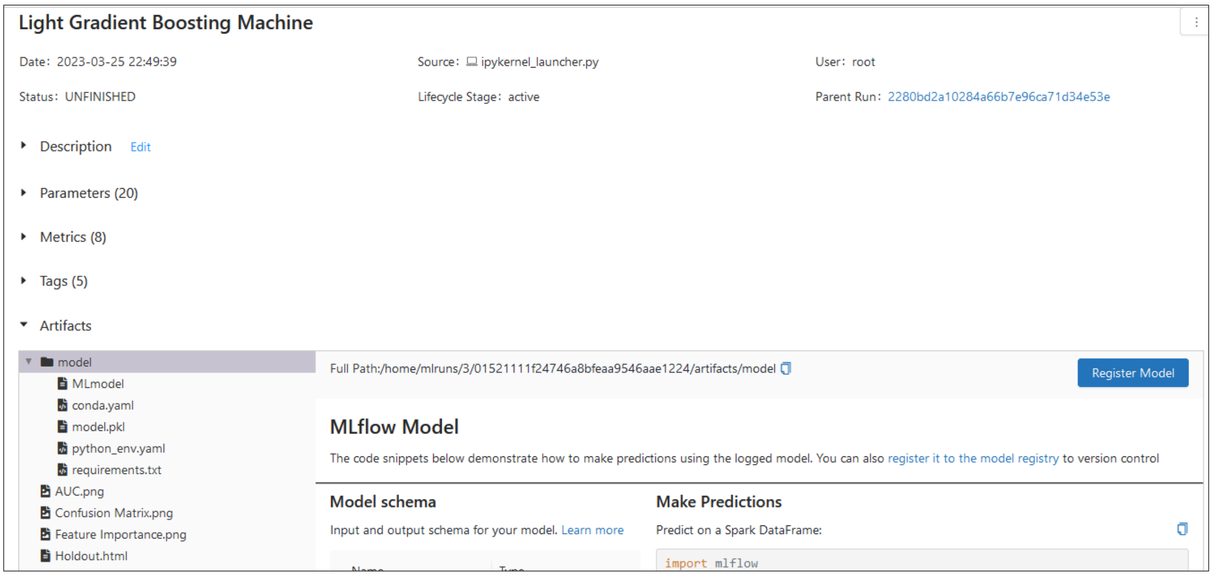
Track machine learning training runs in experiments. Learn more

Experiment ID: 3

Showing 22 matching runs

								Metrics >	
☐	Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy
☐	10 hours ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.738
☐	10 hours ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739
☐	10 hours ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739
☐	✓ 10 hours ago	3.0min	Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739
☐	10 hours ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739
☐	10 hours ago		Random For...	root	ipykernel...	-	sklearn	0.819	0.737
☐	10 hours ago		Random For...	root	ipykernel...	-	sklearn	0.819	0.737
☐	10 hours ago		Extra Trees C...	root	ipykernel...	-	sklearn	0.818	0.736

b. SCREENSHOT OF ONE EXPERIMENTS WITH ALL THE ARTIFACTS VISIBLE



Light Gradient Boosting Machine

Date: 2023-03-25 22:49:39

Source: ipykernel_launcher.py

User: root

Status: UNFINISHED

Lifecycle Stage: active

Parent Run: 2280bd2a10284a66b7e96ca71d34e53e

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt
- AUC.png
- Confusion Matrix.png
- Feature Importance.png
- Holdout.html

Full Path:/home/mlruns/3/01521111f24746a8bfeaa9546aae1224/artifacts/model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

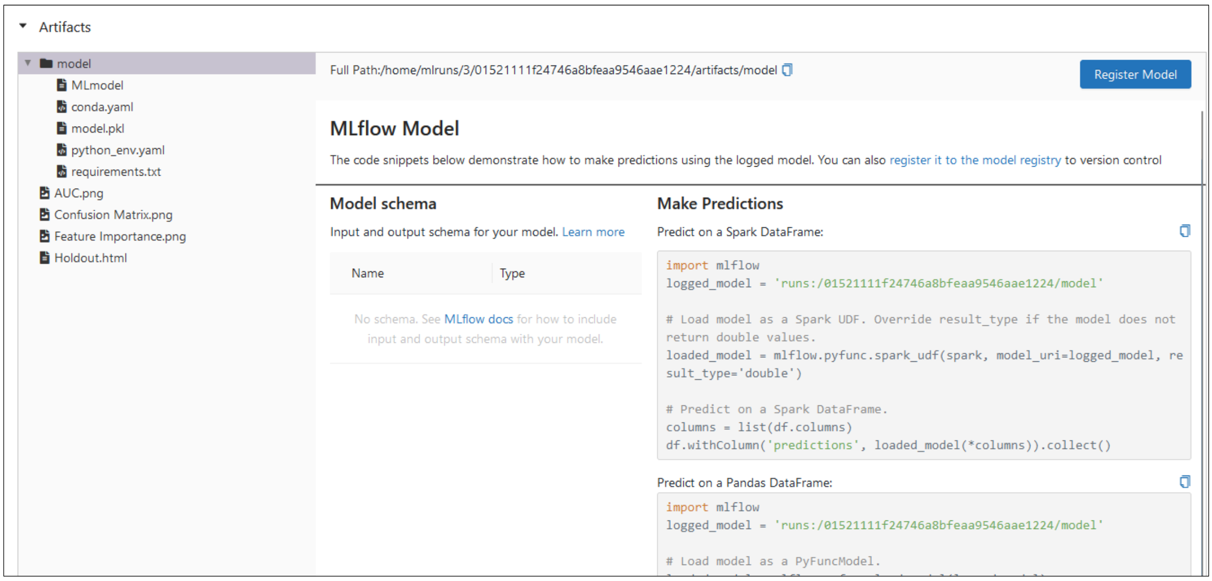
Model schema

Input and output schema for your model. Learn more

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
```



▾ Artifacts

- model
  - MLmodel
  - conda.yaml
  - model.pkl
  - python_env.yaml
  - requirements.txt
- AUC.png
- Confusion Matrix.png
- Feature Importance.png
- Holdout.html

Full Path:/home/mlruns/3/01521111f24746a8bfeaa9546aae1224/artifacts/model

Register Model

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. You can also register it to the model registry to version control

Model schema

Input and output schema for your model. Learn more

Make Predictions

Predict on a Spark DataFrame:

Name	Type

No schema. See MLflow docs for how to include input and output schema with your model.

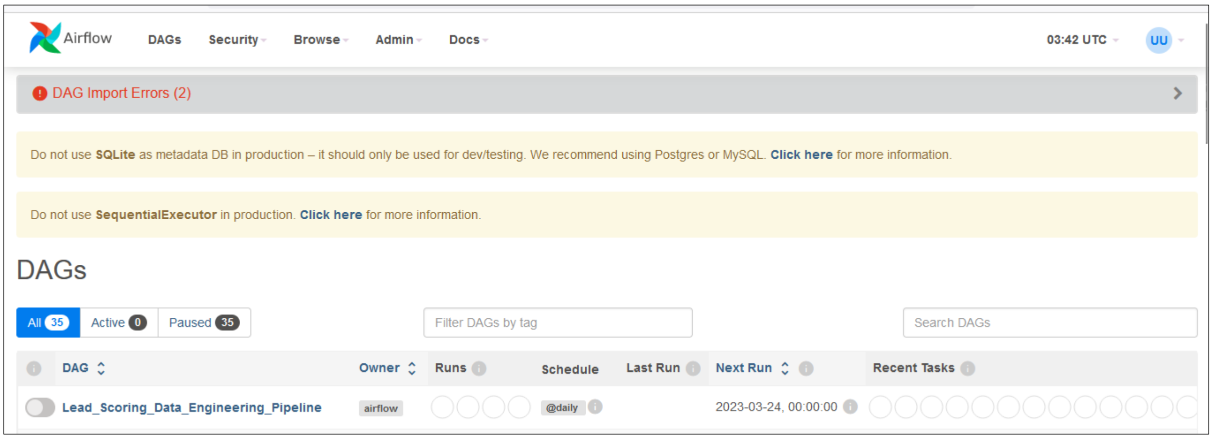
```
import mlflow
logged_model = 'runs:/01521111f24746a8bfeaa9546aae1224/model'

# Load model as a Spark UDF. Override result_type if the model does not
return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, re
sult_type='double')

# Predict on a Spark DataFrame.
columns = list(df.columns)
df.withColumn('predictions', loaded_model(*columns)).collect()
```

Predict on a Pandas DataFrame:

```
import mlflow
logged_model = 'runs:/01521111f24746a8bfeaa9546aae1224/model'

# Load model as a PyFuncModel.
```

DATA PIPELINE

1. CREATE AN AIRFLOW DAG PYTHON FILE FOR DATA PIPELINE



DAG Import Errors (2)

Do not use **SQLite** as metadata DB in production – it should only be used for dev/testing. We recommend using Postgres or MySQL. **Click here** for more information.

Do not use **SequentialExecutor** in production. **Click here** for more information.

DAGs

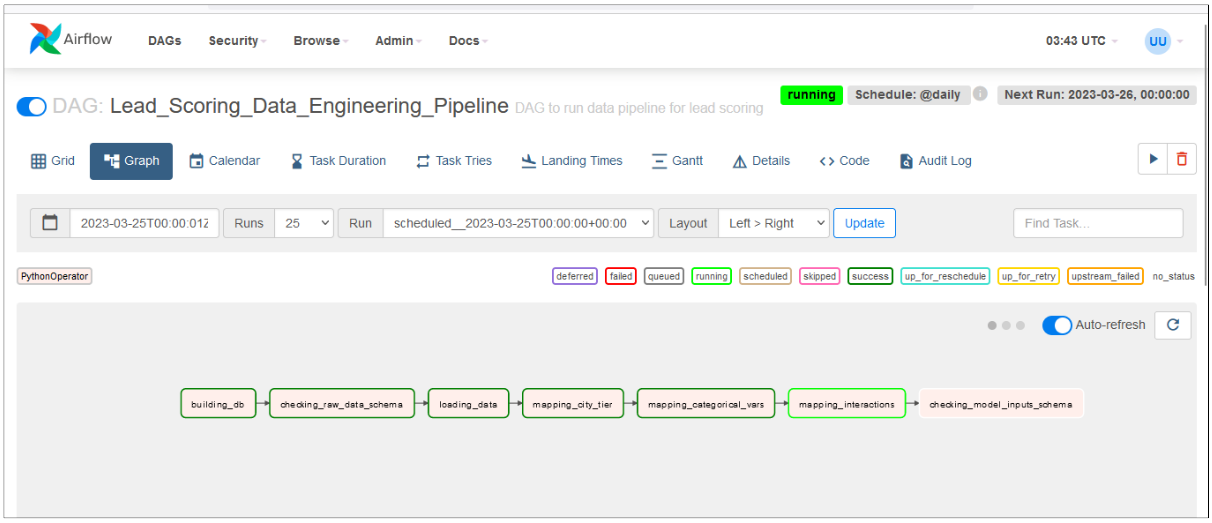
All 35 | Active 0 | Paused 35

Filter DAGs by tag

Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow		@daily		2023-03-24, 00:00:00	

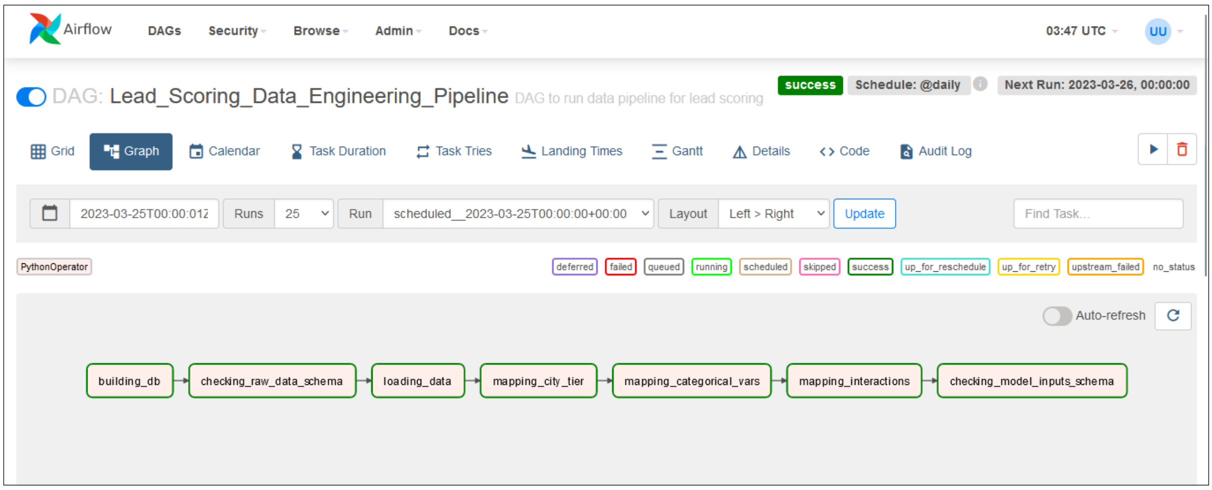
2. Screenshot of successful execution Airflow DAG in graph



DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

running | Schedule: @daily | Next Run: 2023-03-26, 00:00:00

2023-03-25T00:00:01Z | Runs 25 | Run scheduled__2023-03-25T00:00:00+00:00 | Layout Left > Right

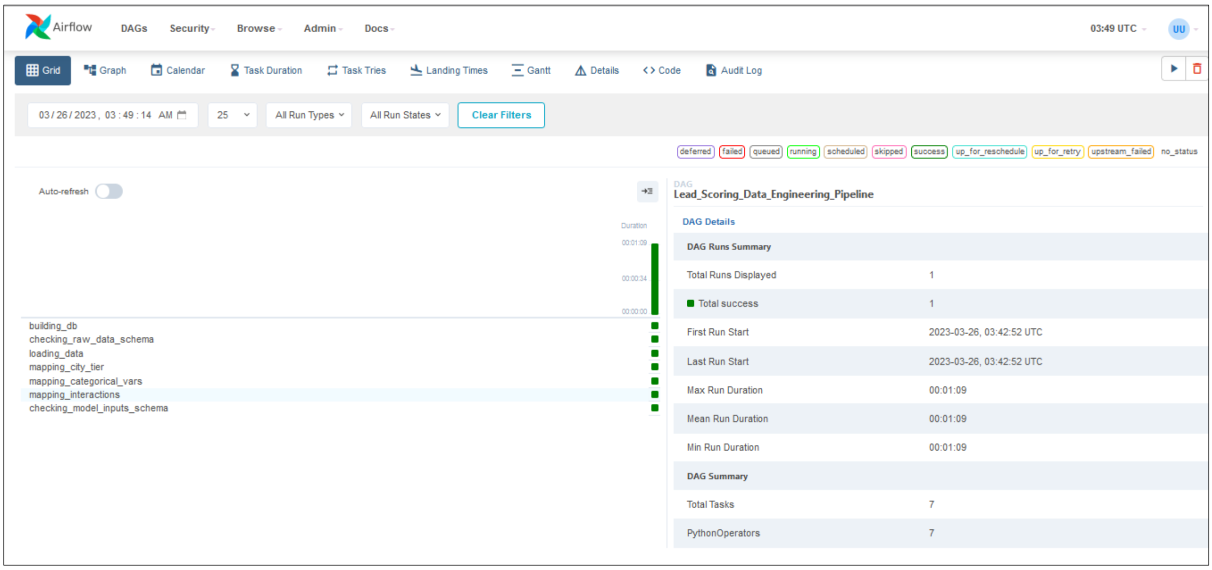


DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

success | Schedule: @daily | Next Run: 2023-03-26, 00:00:00

2023-03-25T00:00:01Z | Runs 25 | Run scheduled__2023-03-25T00:00:00+00:00 | Layout Left > Right

3. Screenshot of Airflow UI grid



building_db
checking_raw_data_schema
loading_data
mapping_city_tier
mapping_categorical_vars
mapping_interactions
checking_model_inputs_schema

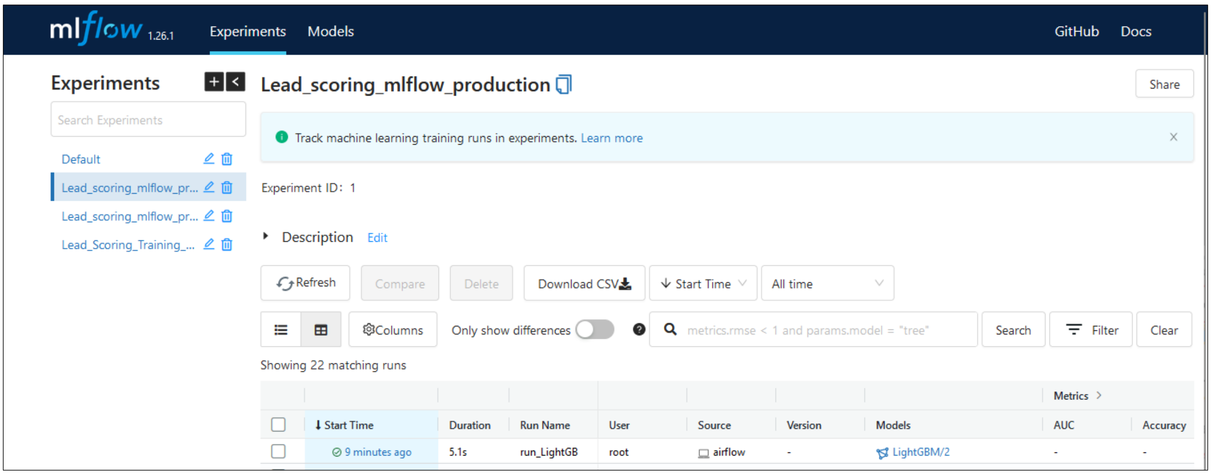
DAG
Lead_Scoring_Data_Engineering_Pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	1
🟩 Total success	1
First Run Start	2023-03-26, 03:42:52 UTC
Last Run Start	2023-03-26, 03:42:52 UTC
Max Run Duration	00:01:09
Mean Run Duration	00:01:09
Min Run Duration	00:01:09
DAG Summary	
Total Tasks	7
PythonOperators	7

TRAINING PIPELINE

1. Create functions for training pipeline



Lead_scoring_mlflow_production

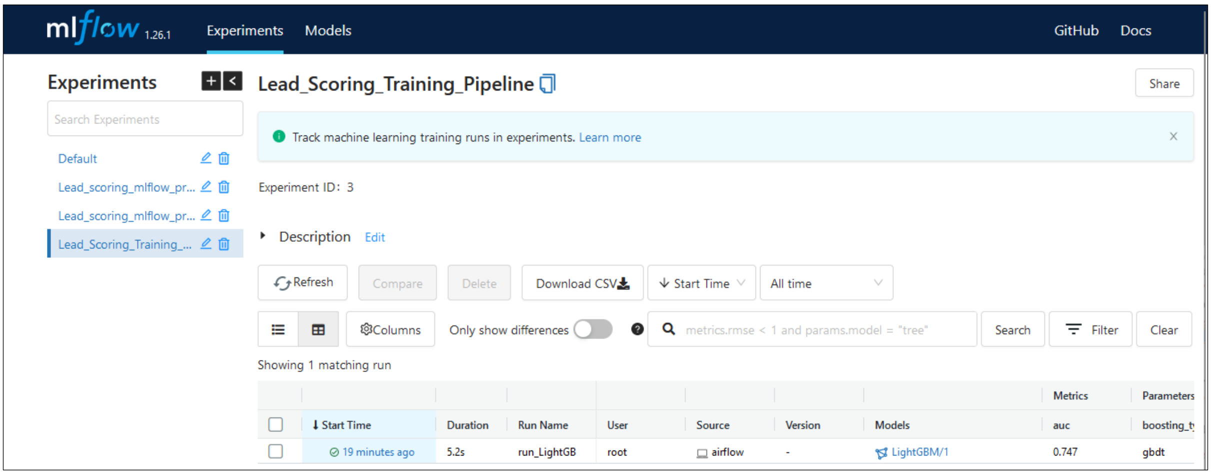
Track machine learning training runs in experiments. Learn more

Experiment ID: 1

Description Edit

Showing 22 matching runs

								Metrics >	
□	Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy
□	9 minutes ago	5.1s	run_LightGB	root	airflow	-	LightGBM/2	-	-



Lead_Scoring_Training_Pipeline

Track machine learning training runs in experiments. Learn more

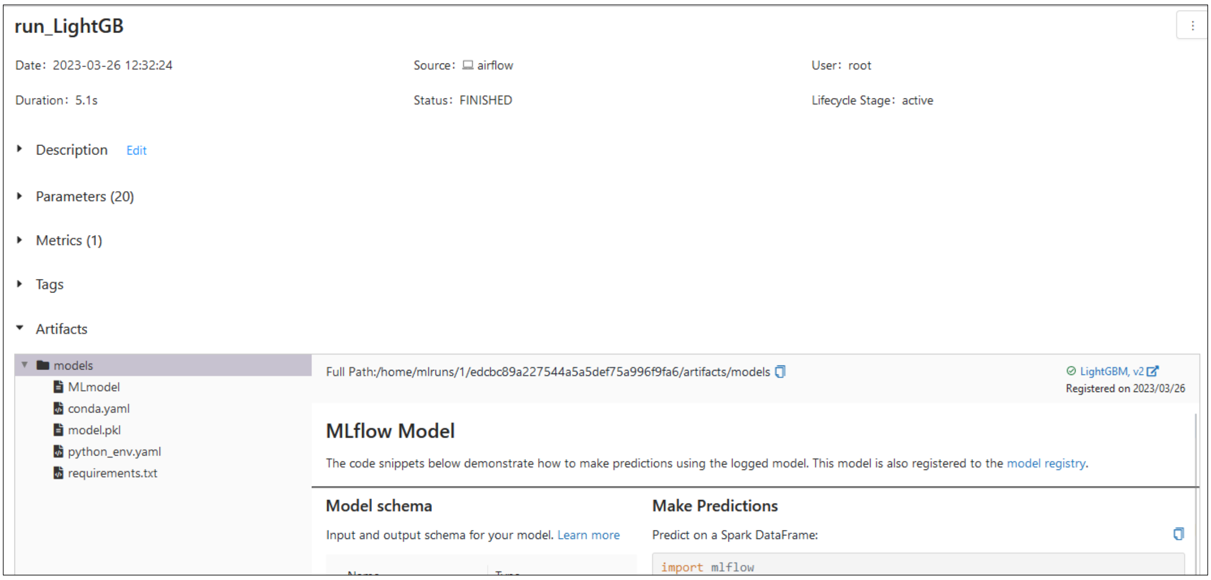
Experiment ID: 3

Description Edit

Showing 1 matching run

								Metrics	Parameters
□	Start Time	Duration	Run Name	User	Source	Version	Models	auc	boosting_t
□	19 minutes ago	5.2s	run_LightGB	root	airflow	-	LightGBM/1	0.747	gbdt

2. Screenshot of experiments with all the artifacts visible



run_LightGB

Date: 2023-03-26 12:32:24
Source: airflow
User: root
Duration: 5.1s
Status: FINISHED
Lifecycle Stage: active

Full Path:/home/mlruns/1/edcbc89a227544a5a5def75a996f9fa6/artifacts/models

LightGBM, v2
Registered on 2023/03/26

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the model registry.

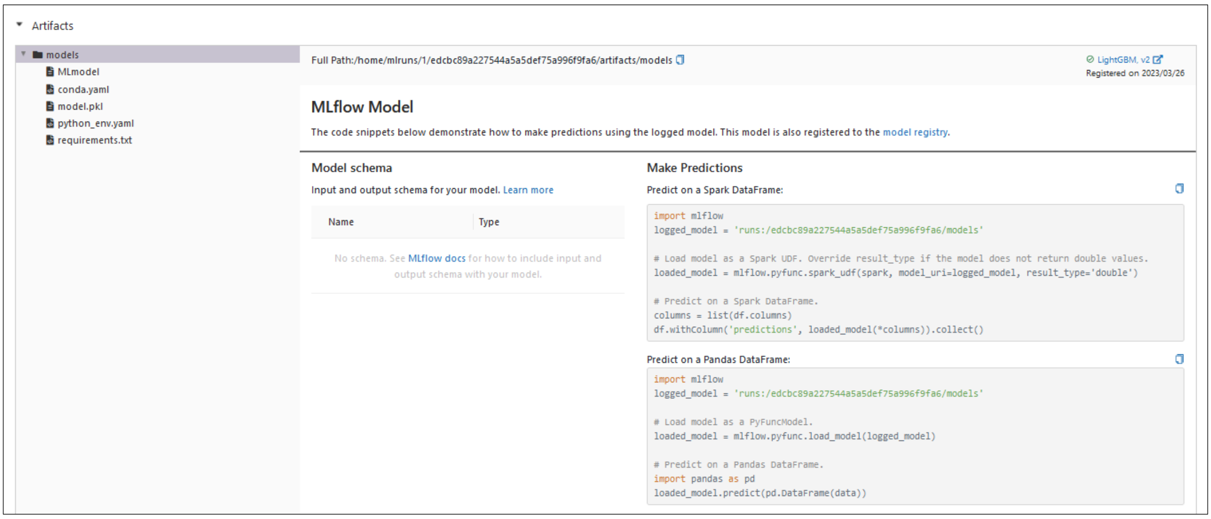
Model schema

Input and output schema for your model. Learn more

Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
```



Full Path:/home/mlruns/1/edcbc89a227544a5a5def75a996f9fa6/artifacts/models

LightGBM, v2
Registered on 2023/03/26

MLflow Model

The code snippets below demonstrate how to make predictions using the logged model. This model is also registered to the model registry.

Model schema

Input and output schema for your model. Learn more

Name	Type

No schema. See MLflow docs for how to include input and output schema with your model.

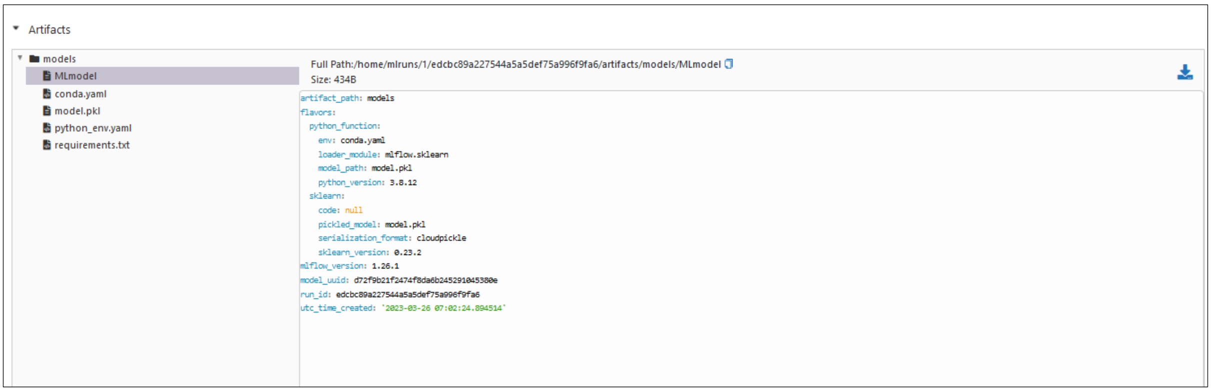
Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
logged_model = 'runs:/edcbc89a227544a5a5def75a996f9fa6/models'

# Load model as a Spark UDF. Override result_type if the model does not return double values.
loaded_model = mlflow.pyfunc.spark_udf(spark, model_uri=logged_model, result_type='double')

# Predict on a Spark DataFrame.
columns = list(df.columns)
df.withColumn('predictions', loaded_model(*columns)).collect()
```

Predict on a Pandas DataFrame:

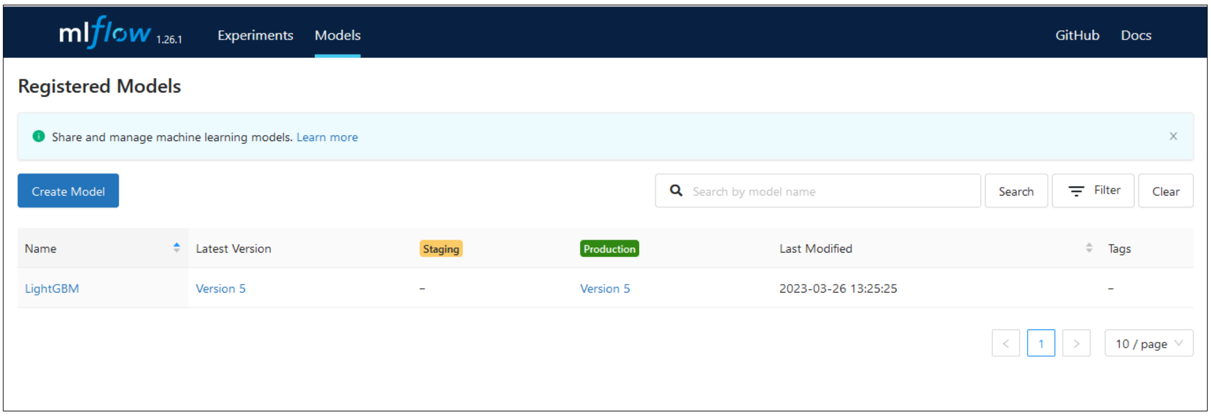
```
import mlflow
logged_model = 'runs:/edcbc89a227544a5a5def75a996f9fa6/models'

# Load model as a PyFuncModel.
loaded_model = mlflow.pyfunc.load_model(logged_model)

# Predict on a Pandas DataFrame.
import pandas as pd
loaded_model.predict(pd.DataFrame(data))
```



Full Path:/home/mlruns/1/edcbc89a227544a5a5def75a996f9fa6/artifacts/models/MLmodel
Size: 434B

```
artifact_path: models
flavors:
  python_function:
    env: conda.yaml
    loader_module: mlflow.sklearn
    model_path: model.pkl
    python_version: 3.8.12
  sklearn:
    code: null
    pickled_model: model.pkl
    serialization_format: cloudpickle
    sklearn_version: 0.23.2
mlflow_version: 1.26.1
model_uuid: d72f9b21f2474f8da6b245291045380e
run_id: edcbc89a227544a5a5def75a996f9fa6
utc_time_created: '2023-03-26 07:02:24.894514'
```

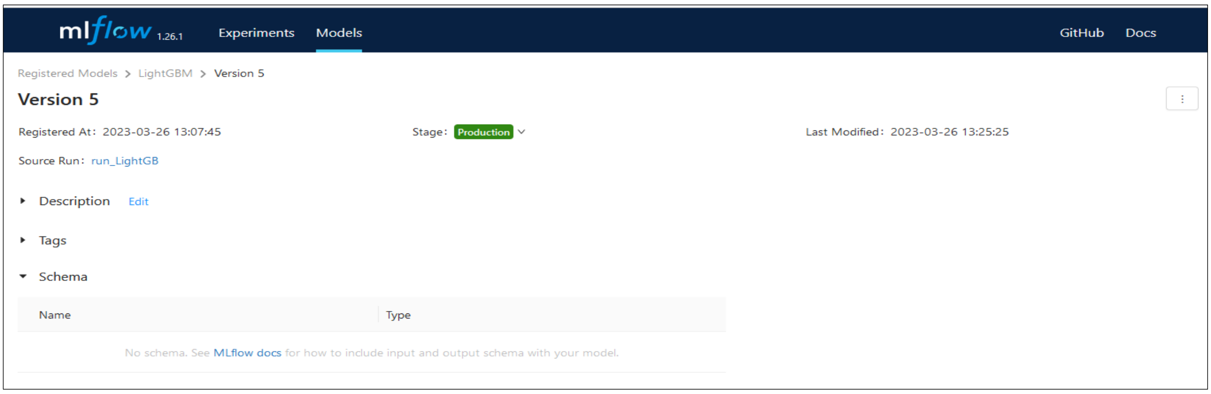
3. Screenshot of model registry with model name and stage as 'production'



Registered Models

Share and manage machine learning models. Learn more

Name	Latest Version	Staging	Production	Last Modified	Tags
LightGBM	Version 5	-	Version 5	2023-03-26 13:25:25	-



Version 5

Registered At: 2023-03-26 13:07:45

Stage: Production

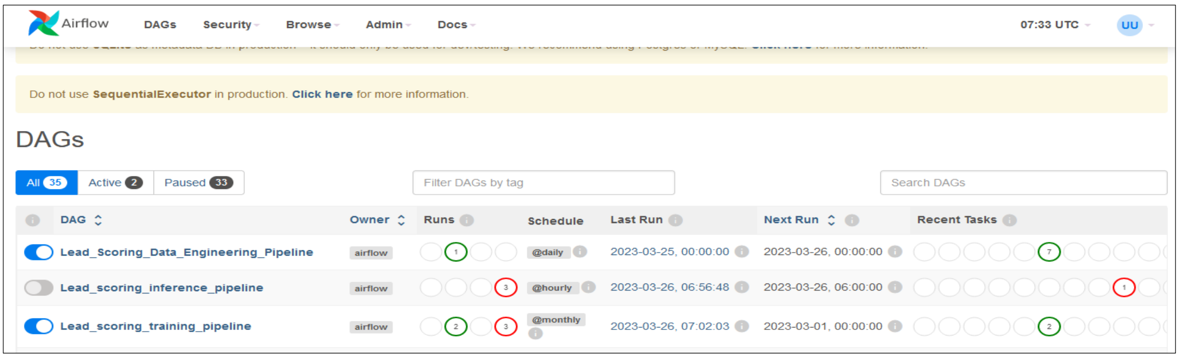
Last Modified: 2023-03-26 13:25:25

Source Run: run_LightGBM

Name	Type

No schema. See MLflow docs for how to include input and output schema with your model.

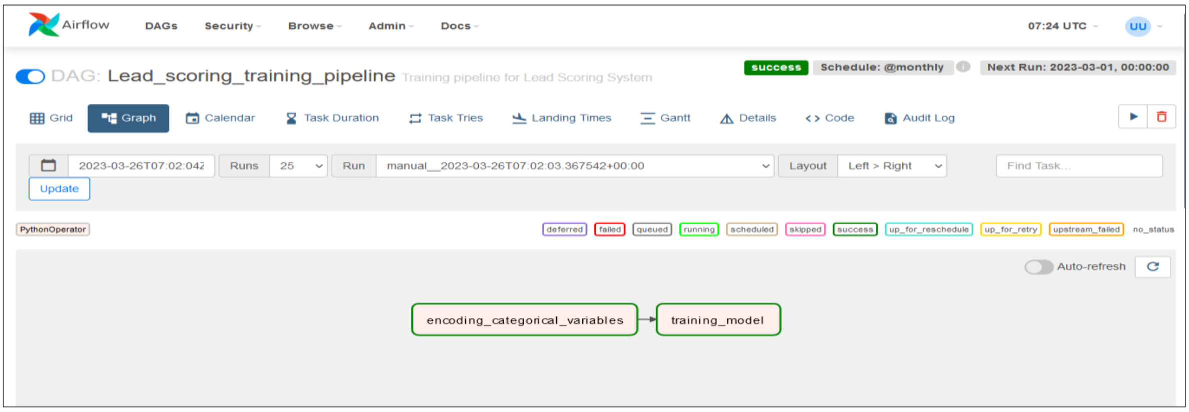
4. Screenshot of successful execution Airflow DAG in graph



Do not use SequentialExecutor in production. Click here for more information.

DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow	1	@daily	2023-03-25, 00:00:00	2023-03-26, 00:00:00	7
Lead_scoring_inference_pipeline	airflow	3	@hourly	2023-03-26, 06:56:48	2023-03-26, 06:00:00	1
Lead_scoring_training_pipeline	airflow	2 3	@monthly	2023-03-26, 07:02:03	2023-03-01, 00:00:00	2

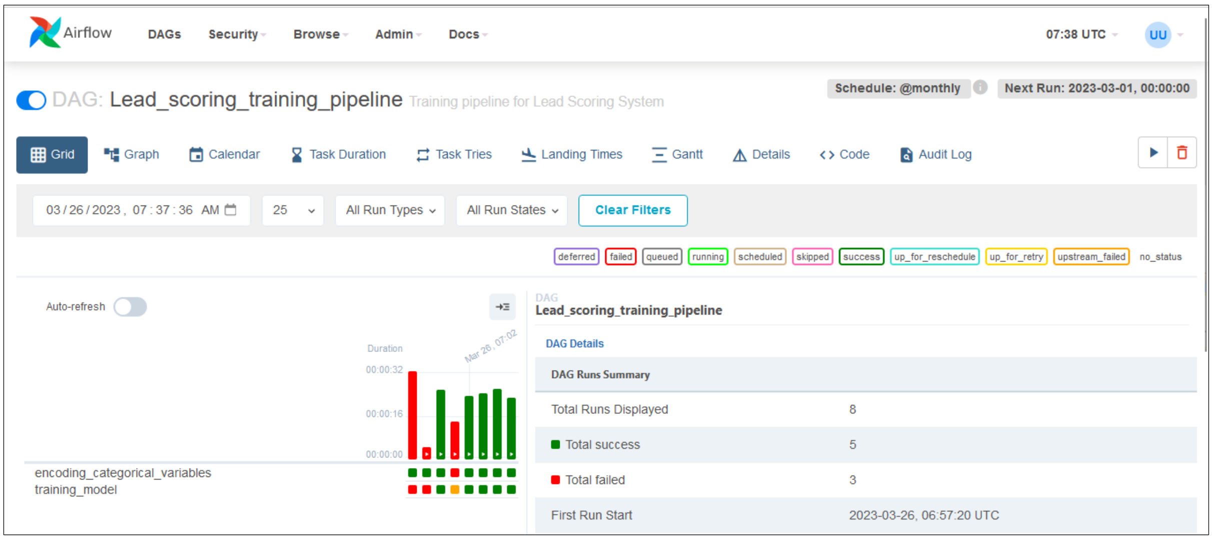


DAG: Lead_scoring_training_pipeline Training pipeline for Lead Scoring System

success Schedule: @monthly Next Run: 2023-03-01, 00:00:00

2023-03-26T07:02:04Z Runs 25 Run manual__2023-03-26T07:02:03.367542+00:00 Layout Left > Right

5. Screenshot of Airflow UI grid



DAG: Lead_scoring_training_pipeline Training pipeline for Lead Scoring System

Schedule: @monthly

Next Run: 2023-03-01, 00:00:00

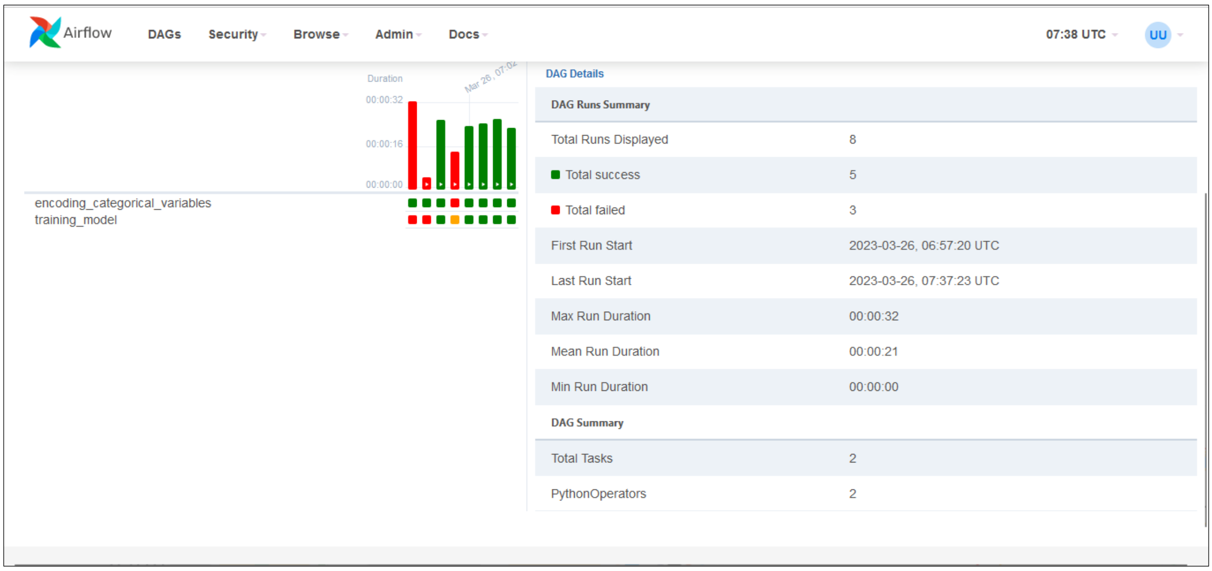
03 / 26 / 2023 , 07 : 37 : 36 AM

25

DAG
Lead_scoring_training_pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	8
🟩 Total success	5
🟥 Total failed	3
First Run Start	2023-03-26, 06:57:20 UTC

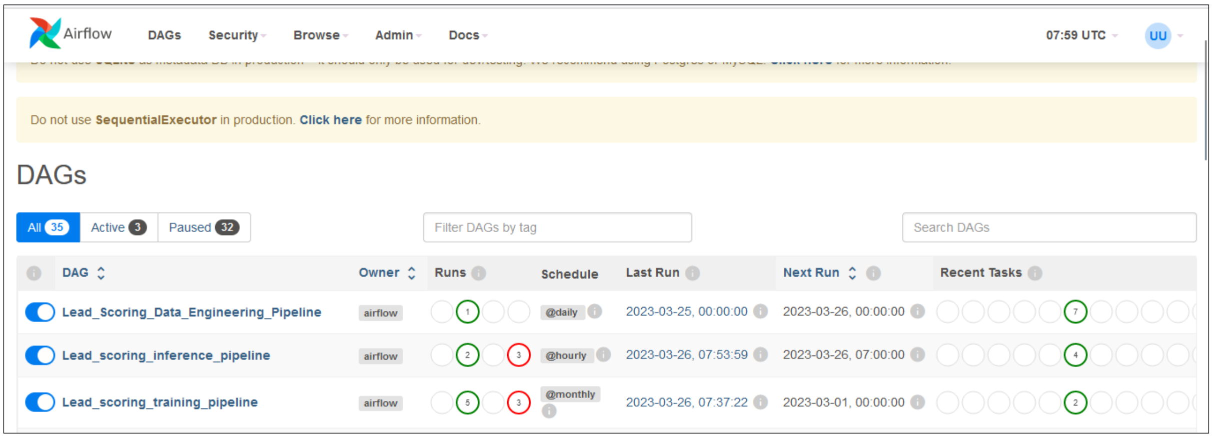


DAG Details

DAG Runs Summary	
Total Runs Displayed	8
🟩 Total success	5
🟥 Total failed	3
First Run Start	2023-03-26, 06:57:20 UTC
Last Run Start	2023-03-26, 07:37:23 UTC
Max Run Duration	00:00:32
Mean Run Duration	00:00:21
Min Run Duration	00:00:00
DAG Summary	
Total Tasks	2
PythonOperators	2

INFERENCE PIPELINE

1. Screenshot of Airflow UI DAGS



Do not use SequentialExecutor in production. Click here for more information.

DAGs

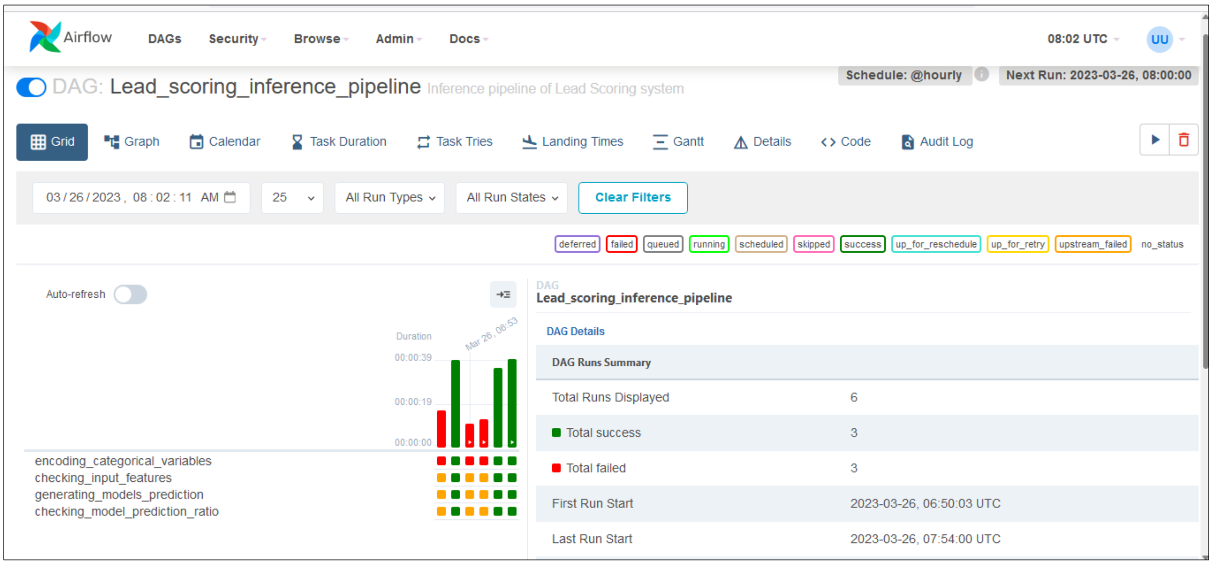
All 35 | Active 3 | Paused 32

Filter DAGs by tag

Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow	1	@daily	2023-03-25, 00:00:00	2023-03-26, 00:00:00	7
Lead_scoring_inference_pipeline	airflow	2 3	@hourly	2023-03-26, 07:53:59	2023-03-26, 07:00:00	4
Lead_scoring_training_pipeline	airflow	5 3	@monthly	2023-03-26, 07:37:22	2023-03-01, 00:00:00	2

2. Screenshot of Airflow UI grid



DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

Schedule: @hourly | Next Run: 2023-03-26, 08:00:00

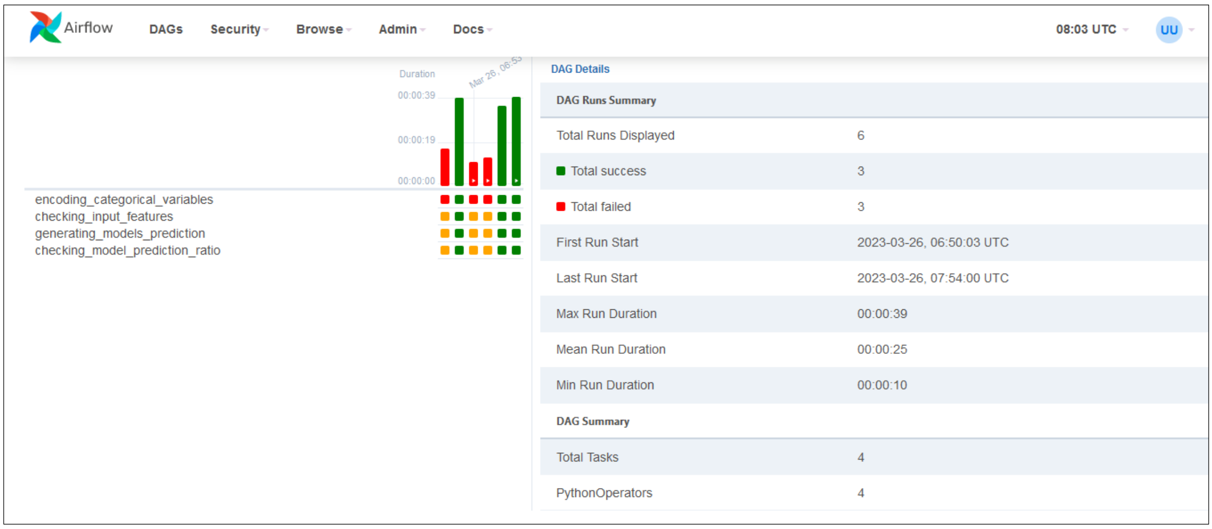
03 / 26 / 2023 , 08 : 02 : 11 AM | 25 | All Run Types | All Run States | Clear Filters

encoding_categorical_variables
checking_input_features
generating_models_prediction
checking_model_prediction_ratio

DAG
Lead_scoring_inference_pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	6
🟩 Total success	3
🟥 Total failed	3
First Run Start	2023-03-26, 06:50:03 UTC
Last Run Start	2023-03-26, 07:54:00 UTC

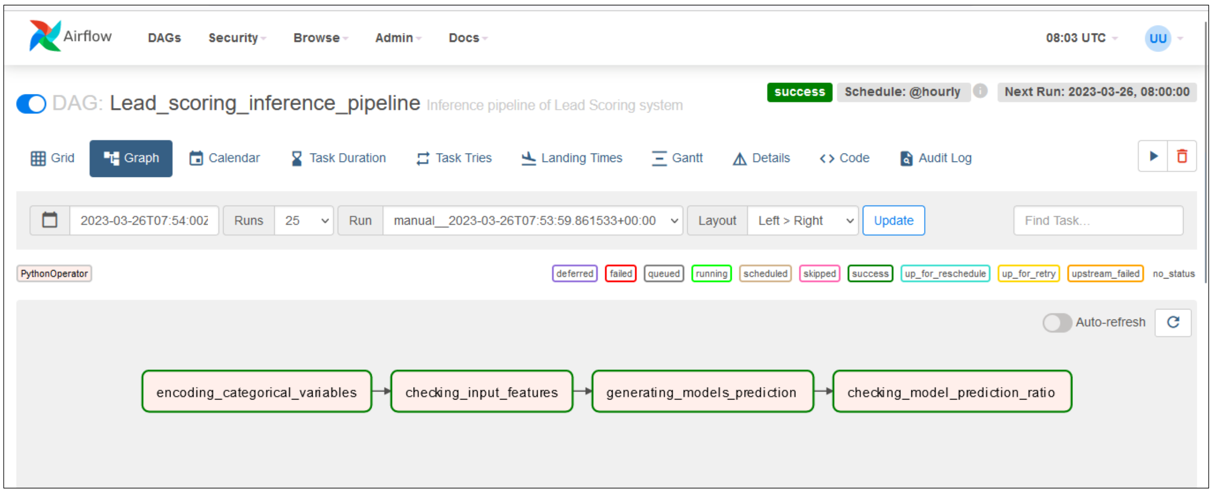


encoding_categorical_variables
checking_input_features
generating_models_prediction
checking_model_prediction_ratio

DAG Details

DAG Runs Summary	
Total Runs Displayed	6
🟩 Total success	3
🟥 Total failed	3
First Run Start	2023-03-26, 06:50:03 UTC
Last Run Start	2023-03-26, 07:54:00 UTC
Max Run Duration	00:00:39
Mean Run Duration	00:00:25
Min Run Duration	00:00:10
DAG Summary	
Total Tasks	4
PythonOperators	4

3. Screenshot of successful execution Airflow DAG in graph



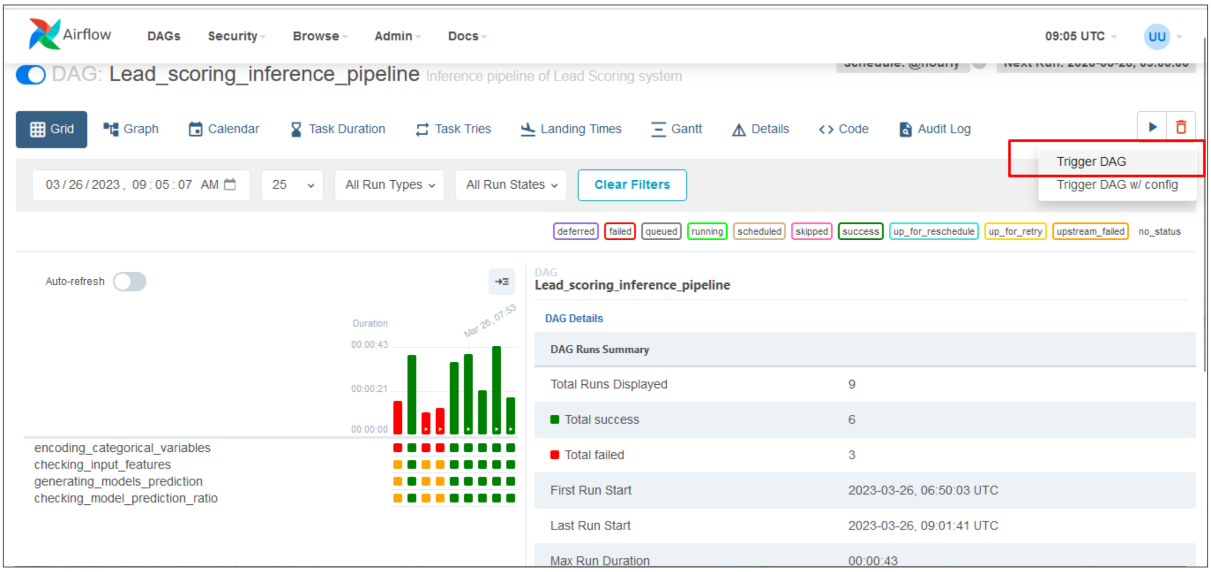
DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

success Schedule: @hourly Next Run: 2023-03-26, 08:00:00

2023-03-26T07:54:00Z Runs 25 Run manual__2023-03-26T07:53:59.861533+00:00 Layout Left > Right

Airflow UI for submission

1. Manually trigger the inference pipeline and see if we are getting the expected results



DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

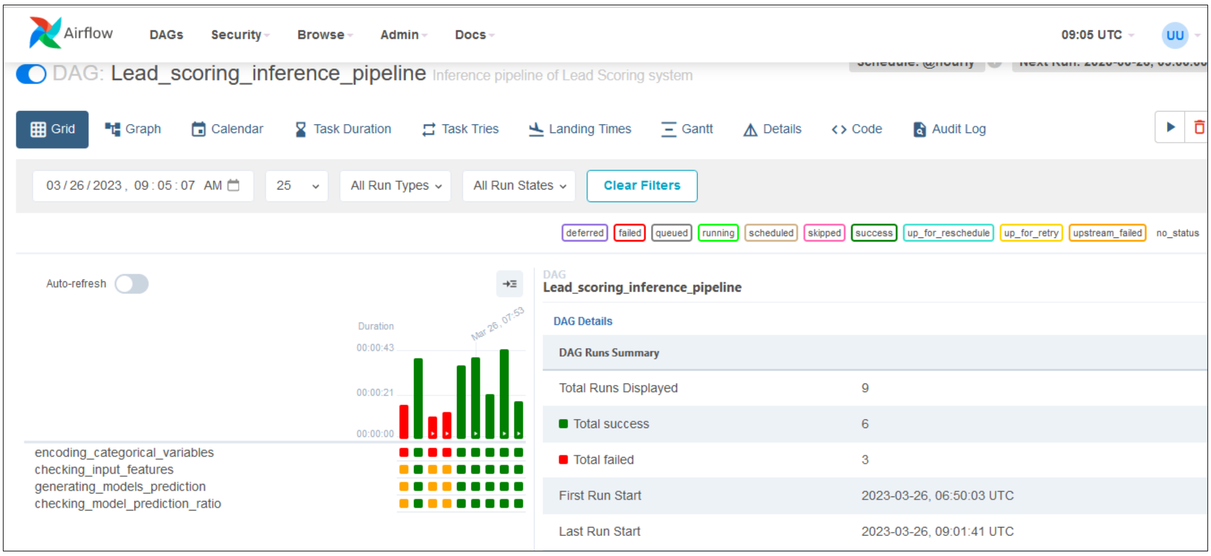
03 / 26 / 2023 , 09 : 05 : 07 AM | 25 | All Run Types | All Run States

encoding_categorical_variables
checking_input_features
generating_models_prediction
checking_model_prediction_ratio

DAG
Lead_scoring_inference_pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	9
🟩 Total success	6
🟥 Total failed	3
First Run Start	2023-03-26, 06:50:03 UTC
Last Run Start	2023-03-26, 09:01:41 UTC
Max Run Duration	00:00:43



DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

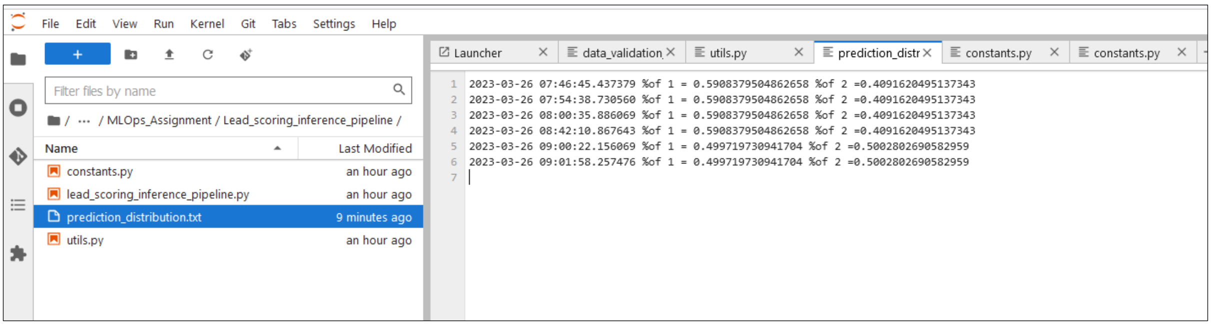
03 / 26 / 2023 , 09 : 05 : 07 AM | 25 | All Run Types | All Run States

encoding_categorical_variables
checking_input_features
generating_models_prediction
checking_model_prediction_ratio

DAG
Lead_scoring_inference_pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	9
🟩 Total success	6
🟥 Total failed	3
First Run Start	2023-03-26, 06:50:03 UTC
Last Run Start	2023-03-26, 09:01:41 UTC

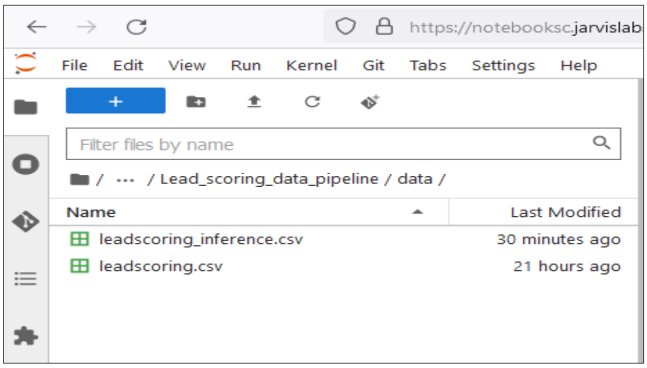


/ ··· / MLOps_Assignment / Lead_scoring_inference_pipeline /

Name	Last Modified
constants.py	an hour ago
lead_scoring_inference_pipeline.py	an hour ago
prediction_distribution.txt	9 minutes ago
utils.py	an hour ago

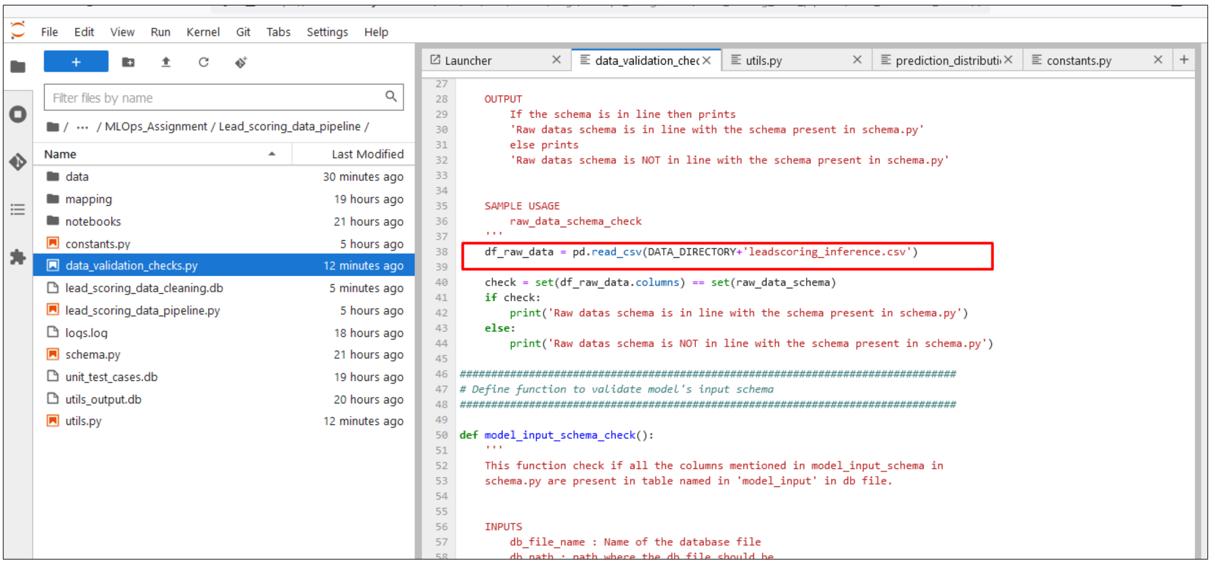
```
2023-03-26 07:46:45.437379 %of 1 = 0.5908379504862658 %of 2 =0.4091620495137343
2023-03-26 07:54:38.730560 %of 1 = 0.5908379504862658 %of 2 =0.4091620495137343
2023-03-26 08:00:35.886069 %of 1 = 0.5908379504862658 %of 2 =0.4091620495137343
2023-03-26 08:42:10.867643 %of 1 = 0.5908379504862658 %of 2 =0.4091620495137343
2023-03-26 09:00:22.156069 %of 1 = 0.499719730941704 %of 2 =0.5002802690582959
2023-03-26 09:01:58.257476 %of 1 = 0.499719730941704 %of 2 =0.5002802690582959
```

2. Copy the “leadscoring_inference.csv” file present in “Assignment/03_inference_pipeline/Data” to the “airflow/dags/Lead_scoring_data_pipeline/data” folder and modify the utils.py in the Lead_scoring_data_pipeline such that the data pipeline read “leadscoring_inference.csv” instead of “leadscoring.csv”



/ … / Lead_scoring_data_pipeline / data /

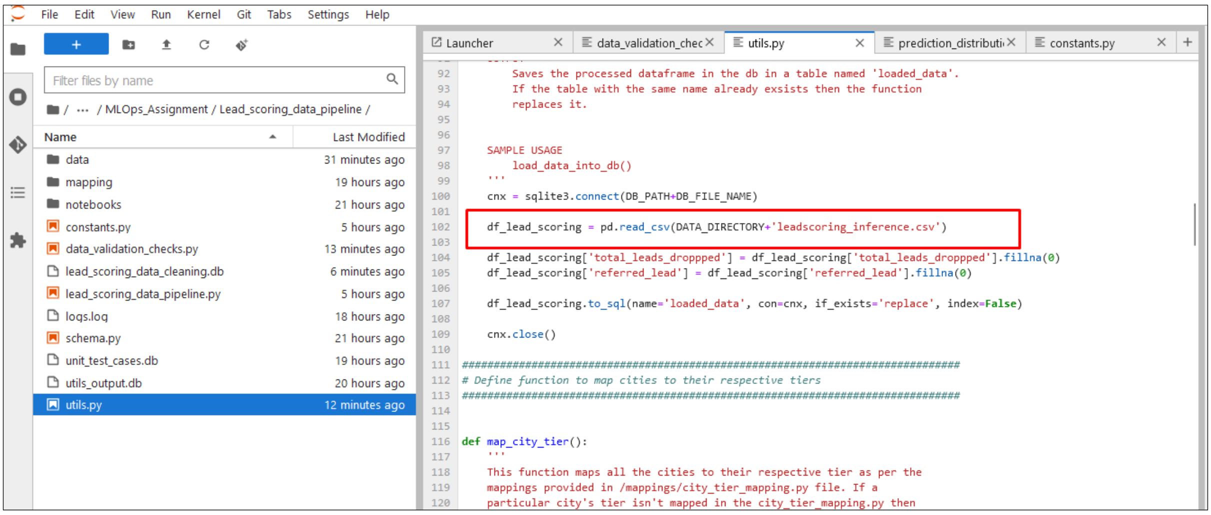
Name	Last Modified
leadscoring_inference.csv	30 minutes ago
leadscoring.csv	21 hours ago



/ … / MLOps_Assignment / Lead_scoring_data_pipeline /

Name	Last Modified
data	30 minutes ago
mapping	19 hours ago
notebooks	21 hours ago
constants.py	5 hours ago
data_validation_checks.py	12 minutes ago
lead_scoring_data_cleaning.db	5 minutes ago
lead_scoring_data_pipeline.py	5 hours ago
logs.log	18 hours ago
schema.py	21 hours ago
unit_test_cases.db	19 hours ago
utils_output.db	20 hours ago
utils.py	12 minutes ago

```

    OUTPUT
        If the schema is in line then prints
        'Raw datas schema is in line with the schema present in schema.py'
        else prints
        'Raw datas schema is NOT in line with the schema present in schema.py'


    SAMPLE USAGE
        raw_data_schema_check
    '''
    df_raw_data = pd.read_csv(DATA_DIRECTORY+'leadscoring_inference.csv')

    check = set(df_raw_data.columns) == set(raw_data_schema)
    if check:
        print('Raw datas schema is in line with the schema present in schema.py')
    else:
        print('Raw datas schema is NOT in line with the schema present in schema.py')

###############################################################################
# Define function to validate model's input schema
###############################################################################

def model_input_schema_check():
    '''
    This function check if all the columns mentioned in model_input_schema in
    schema.py are present in table named in 'model_input' in db file.


    INPUTS
        db_file_name : Name of the database file
        db_path : path where the db file should be
```

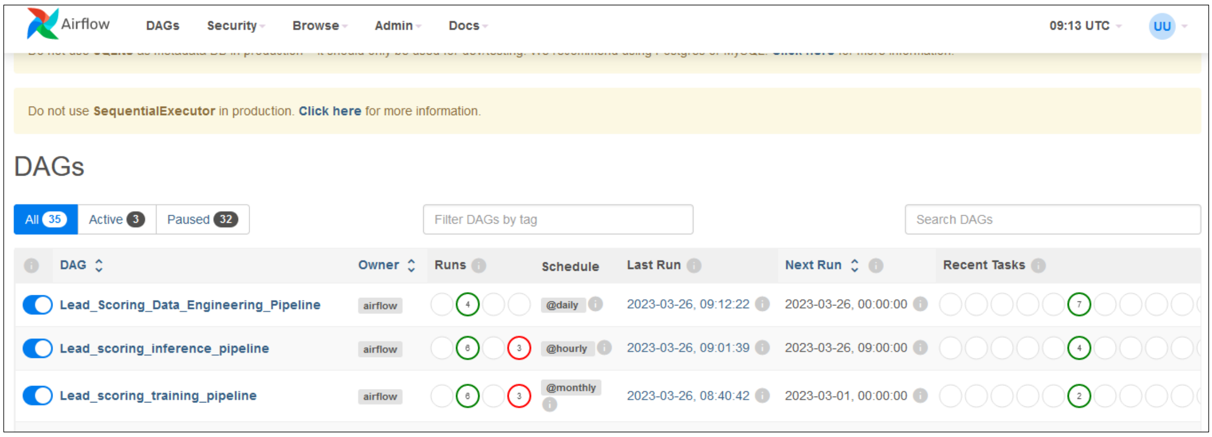


/ … / MLOps_Assignment / Lead_scoring_data_pipeline /

Name	Last Modified
data	31 minutes ago
mapping	19 hours ago
notebooks	21 hours ago
constants.py	5 hours ago
data_validation_checks.py	13 minutes ago
lead_scoring_data_cleaning.db	6 minutes ago
lead_scoring_data_pipeline.py	5 hours ago
logs.log	18 hours ago
schema.py	21 hours ago
unit_test_cases.db	19 hours ago
utils_output.db	20 hours ago
utils.py	12 minutes ago

```
        Saves the processed dataframe in the db in a table named 'loaded_data'.
        If the table with the same name already exsists then the function
        replaces it.


    SAMPLE USAGE
        load_data_into_db()
    '''
    cnx = sqlite3.connect(DB_PATH+DB_FILE_NAME)

    df_lead_scoring = pd.read_csv(DATA_DIRECTORY+'leadscoring_inference.csv')

    df_lead_scoring['total_leads_droppped'] = df_lead_scoring['total_leads_droppped'].fillna(0)
    df_lead_scoring['referred_lead'] = df_lead_scoring['referred_lead'].fillna(0)

    df_lead_scoring.to_sql(name='loaded_data', con=cnx, if_exists='replace', index=False)

    cnx.close()

###############################################################################
# Define function to map cities to their respective tiers
###############################################################################


def map_city_tier():
    '''
    This function maps all the cities to their respective tier as per the
    mappings provided in /mappings/city_tier_mapping.py file. If a
    particular city's tier isn't mapped in the city_tier_mapping.py then
```

3. Once this is done first trigger the Data pipeline manually and then trigger the inference pipeline once the data pipeline’s run is complete. Revise till you get the expected output. Follow the instructions mentioned in the Airflow module to run the Inference pipeline and take the screenshot of Airflow UI for submission. Trigger the training pipeline manually

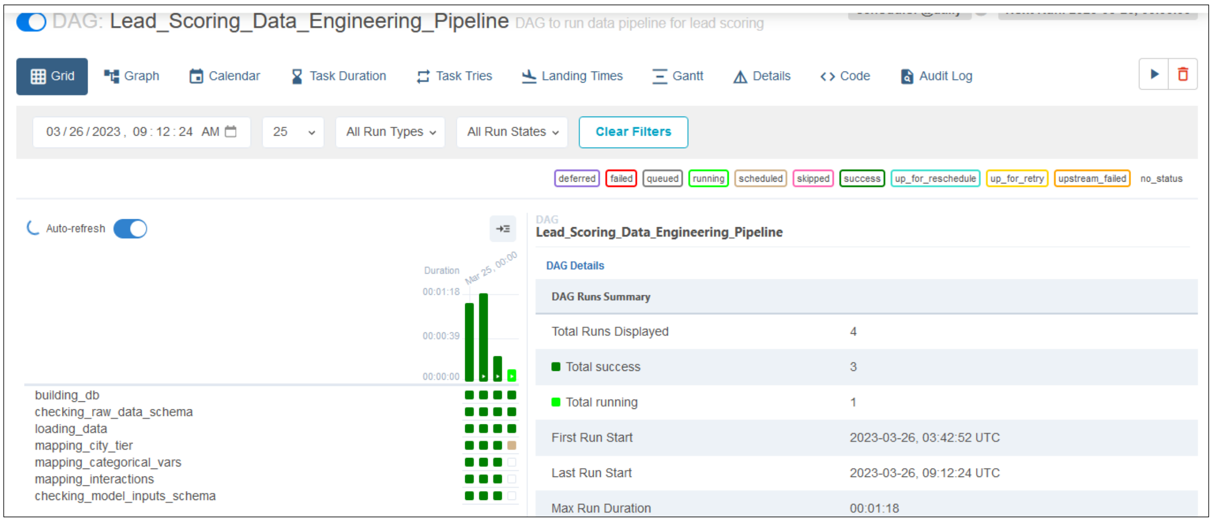


Do not use **SequentialExecutor** in production. **Click here** for more information.

DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow	4	@daily	2023-03-26, 09:12:22	2023-03-26, 00:00:00	7
Lead_scoring_inference_pipeline	airflow	6 3	@hourly	2023-03-26, 09:01:39	2023-03-26, 09:00:00	4
Lead_scoring_training_pipeline	airflow	6 3	@monthly	2023-03-26, 08:40:42	2023-03-01, 00:00:00	2

Lead_Scoring_Data_Engineering_Pipeline



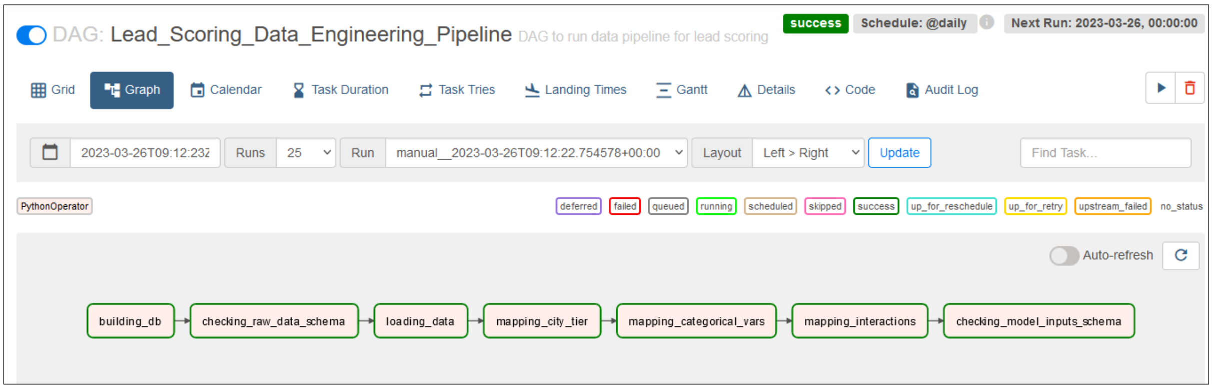
DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

building_db
checking_raw_data_schema
loading_data
mapping_city_tier
mapping_categorical_vars
mapping_interactions
checking_model_inputs_schema

DAG
Lead_Scoring_Data_Engineering_Pipeline

DAG Details

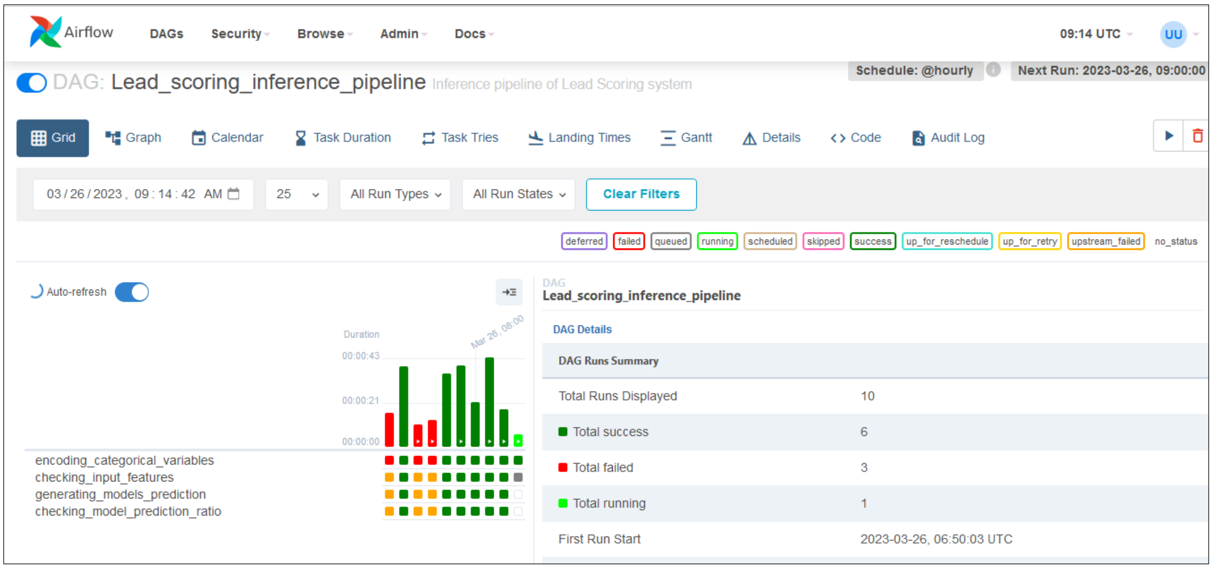
DAG Runs Summary	
Total Runs Displayed	4
🟩 Total success	3
🟩 Total running	1
First Run Start	2023-03-26, 03:42:52 UTC
Last Run Start	2023-03-26, 09:12:24 UTC
Max Run Duration	00:01:18



DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

success Schedule: @daily Next Run: 2023-03-26, 00:00:00

Lead_scoring_inference_pipeline



DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

Schedule: @hourly

Next Run: 2023-03-26, 09:00:00

03 / 26 / 2023 , 09 : 14 : 42 AM

25

All Run Types

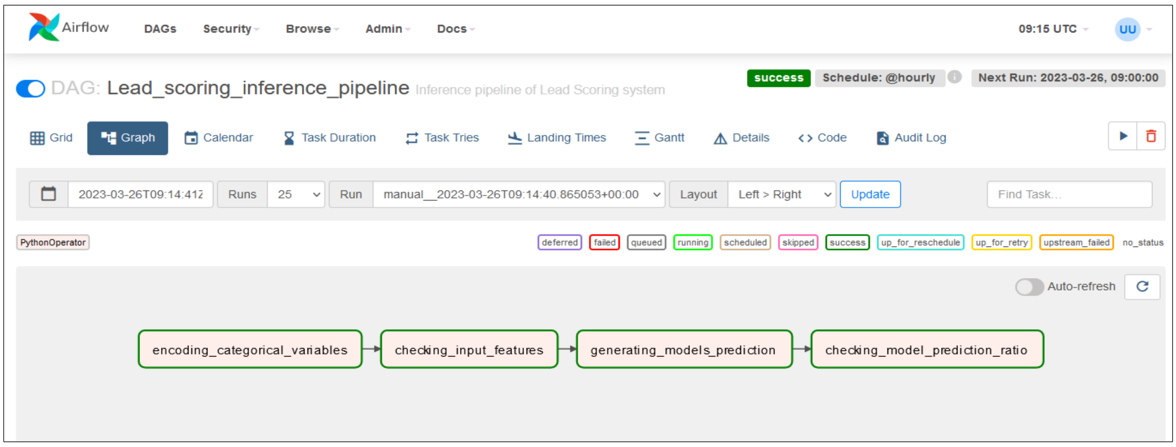
All Run States

encoding_categorical_variables
checking_input_features
generating_models_prediction
checking_model_prediction_ratio

DAG
Lead_scoring_inference_pipeline

DAG Details

DAG Runs Summary	
Total Runs Displayed	10
🟩 Total success	6
🟥 Total failed	3
🟩 Total running	1
First Run Start	2023-03-26, 06:50:03 UTC



DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system

success

Schedule: @hourly

Next Run: 2023-03-26, 09:00:00

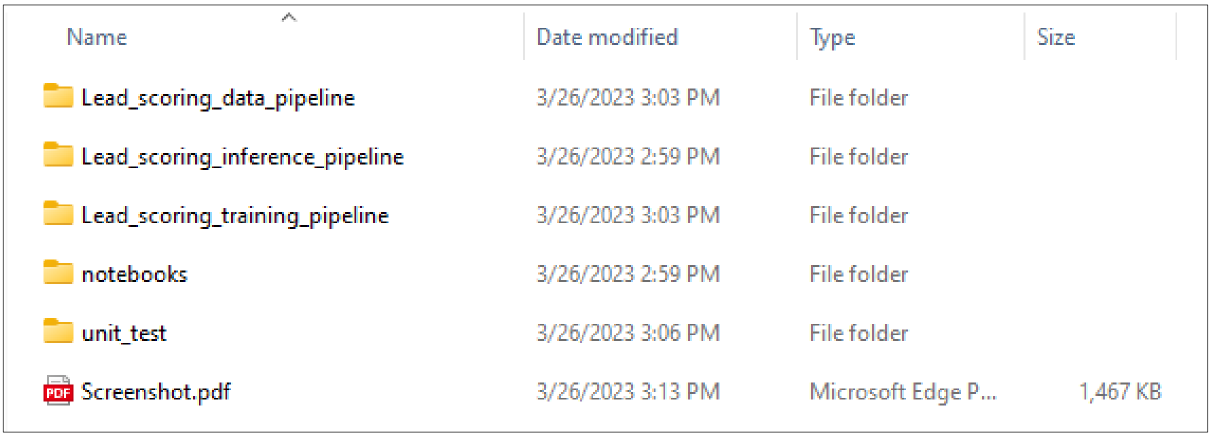
2023-03-26T09:14:41Z

Runs 25

Run manual__2023-03-26T09:14:40.865053+00:00

Layout Left > Right

Screenshot of Folder structure



Name	Date modified	Type	Size
Lead_scoring_data_pipeline	3/26/2023 3:03 PM	File folder	
Lead_scoring_inference_pipeline	3/26/2023 2:59 PM	File folder	
Lead_scoring_training_pipeline	3/26/2023 3:03 PM	File folder	
notebooks	3/26/2023 2:59 PM	File folder	
unit_test	3/26/2023 3:06 PM	File folder	
Screenshot.pdf	3/26/2023 3:13 PM	Microsoft Edge P...	1,467 KB