

## MODEL EXPERIMENTATION:

### Screenshot of mlflow ui- all experiments

The screenshot displays the MLflow UI interface for the experiment 'Lead\_scoring\_Baseline\_model\_01'. The interface includes a search bar, a list of experiments, and a table of runs. The table shows 12 matching runs with columns for Start Time, Duration, Run Name, User, Source, Version, Models, Metrics (AUC, Accuracy, F1, C), and Parameters (CPU Jobs). The runs are sorted by Start Time, showing a sequence of model training runs from 2 minutes ago to 18 seconds ago.

Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy	F1	C	CPU Jobs
2 minutes ago		Session Init...	root	ipykernel...	-	-	-	-	-	-	-1
7 seconds ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.738	0.762	-	-
15 seconds ago		Naive Bayes	root	ipykernel...	-	sklearn	0.738	0.679	0.728	-	-
15 seconds ago		Ridge Classif...	root	ipykernel...	-	sklearn	0	0.715	0.742	-	-
16 seconds ago		Linear Discr...	root	ipykernel...	-	sklearn	0.79	0.715	0.742	-	-
16 seconds ago		Logistic Reg...	root	ipykernel...	-	sklearn	0.792	0.717	0.741	1.0	-
16 seconds ago		Decision Tre...	root	ipykernel...	-	sklearn	0.817	0.736	0.758	-	-
17 seconds ago		Extra Trees C...	root	ipykernel...	-	sklearn	0.818	0.737	0.758	-	-
17 seconds ago		Random For...	root	ipykernel...	-	sklearn	0.819	0.737	0.76	-	-
18 seconds ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.738	0.762	-	-
18 seconds ago		Extreme Gra...	root	ipykernel...	-	-	-	-	-	-	-
2 minutes ago		Session Init...	root	ipykernel...	-	-	-	-	-	-	-1

### Screenshot of mlflow ui-- screenshot of one experiment with all the artifacts visible

The screenshot displays the MLflow UI interface for a specific experiment, 'Lead\_scoring\_Baseline\_model\_01'. The interface shows the experiment details, including the name, date, status, source, user, and parent run. The 'Artifacts' section is expanded, showing a list of artifacts including 'MLmodel', 'conda.yaml', 'model.pkl', 'python\_env.yaml', 'requirements.txt', and 'Holdout.html'. The 'MLflow Model' section is also visible, showing the model schema and the code snippets for making predictions using the logged model.

Lead\_scoring\_Baseline\_model\_01 > Light Gradient Boosting Machine

Date: 2022-11-15 13:13:11 Source: ipykernel\_launcher.py User: root

Status: UNFINISHED Lifecycle Stage: active Parent Run: 43f14f6fb4294b659bb8eeb62338668c

Artifacts

- MLmodel
- conda.yaml
- model.pkl
- python\_env.yaml
- requirements.txt
- Holdout.html

Full Path: /home/Assignment/02\_training\_pipeline/mlruns/1/9ba44ae9d5bc4d008172e3fc1f14550c/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:/9ba44ae9d5bc4d008172e3fc1f14550c/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, result_type='double')
```

## Screenshot of mlflow ui after dropping features—all experiments

The screenshot shows the MLflow UI interface. The top navigation bar includes the MLflow logo and a search bar. The main content area displays the 'Experiments' tab for 'Tuned\_model\_exp01'. A table lists 12 matching runs, showing columns for Start Time, Duration, Run Name, User, Source, Version, Models, Metrics (AUC, Accuracy, F1), and Parameters (C, CPU Jobs). The table is sorted by Start Time, with the most recent run at the top. The bottom status bar shows the system clock and weather information.

Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy	F1	C	CPU Jobs
6 minutes ago	Session Init...	root	ipykernel...	-	-	-	-	-	-	-	-1
50 seconds ago	Light Gradie...	root	ipykernel...	-	-	sklearn	0.82	0.737	0.763	-	-
4 minutes ago	Light Gradie...	root	ipykernel...	-	-	sklearn	0.821	0.739	0.762	-	-
5 minutes ago	Naive Bayes	root	ipykernel...	-	-	sklearn	0.734	0.67	0.723	-	-
5 minutes ago	Linear Disci...	root	ipykernel...	-	-	sklearn	0.773	0.7	0.728	-	-
5 minutes ago	Ridge Classi...	root	ipykernel...	-	-	sklearn	0	0.7	0.728	-	-
5 minutes ago	Logistic Reg...	root	ipykernel...	-	-	sklearn	0.794	0.71	0.74	1.0	-
5 minutes ago	Decision Tre...	root	ipykernel...	-	-	sklearn	0.817	0.737	0.758	-	-
5 minutes ago	Extra Trees C...	root	ipykernel...	-	-	sklearn	0.817	0.737	0.758	-	-
5 minutes ago	Random For...	root	ipykernel...	-	-	sklearn	0.818	0.737	0.759	-	-
5 minutes ago	Extreme Gra...	root	ipykernel...	-	-	sklearn	0.821	0.738	0.762	-	-
5 minutes ago	Light Gradie...	root	ipykernel...	-	-	sklearn	0.821	0.739	0.762	-	-

## Screenshot of mlflow ui after dropping features- one experiment with all artifacts : (Model, AUC, Confusion Matrix, Feature Importance, Holdout.html)

The screenshot shows the MLflow UI interface for a specific experiment. The top navigation bar includes the MLflow logo and a search bar. The main content area displays the 'Experiments' tab for 'Tuned\_model\_exp01'. The experiment details section shows the 'Light Gradient Boosting Machine' model, its source, user, and lifecycle stage. The 'Artifacts' section lists various files, including 'MLmodel', 'conda.yaml', 'model.pkl', 'python\_env.yaml', 'requirements.txt', 'AUC.png', 'Confusion Matrix.png', 'Feature Importance.png', and 'Holdout.html'. The 'Model schema' section shows the input and output schema for the model. The 'Make Predictions' section provides a code snippet for loading the model and making predictions.

Full Path: /home/Assignment/02\_training\_pipeline/mlruns/2/025dd9959bde41d38eec2b9f17b29196/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
------	------

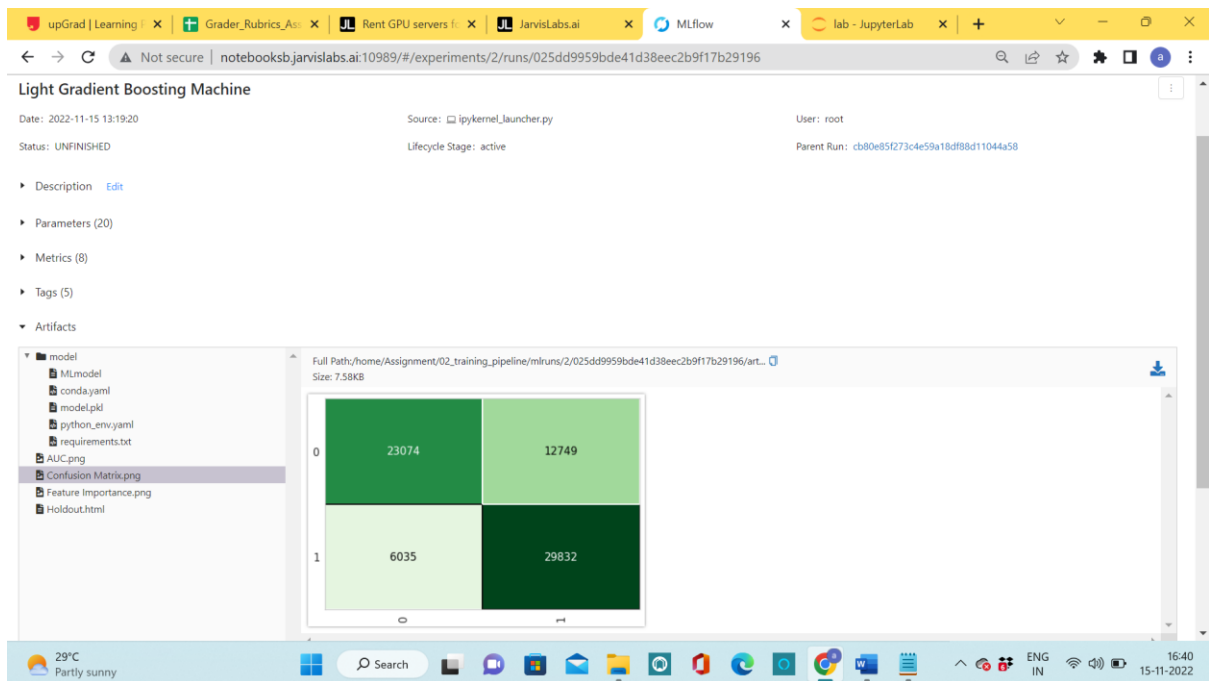
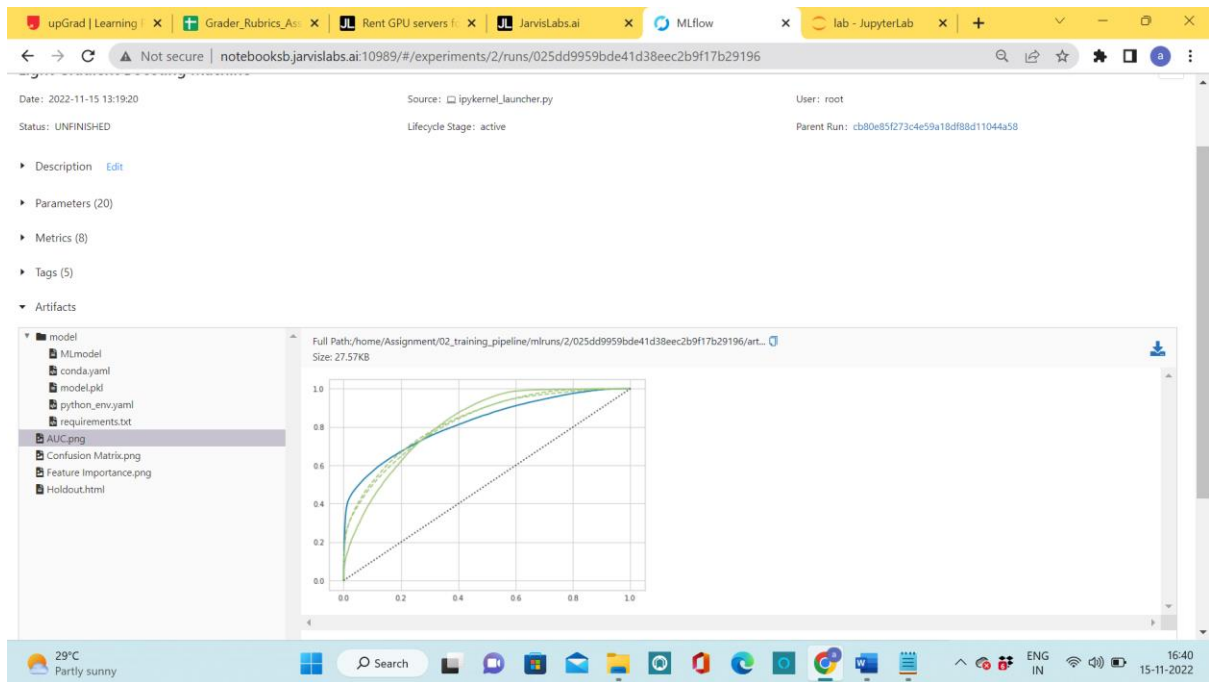
No schema. See MLflow docs for how to include input and

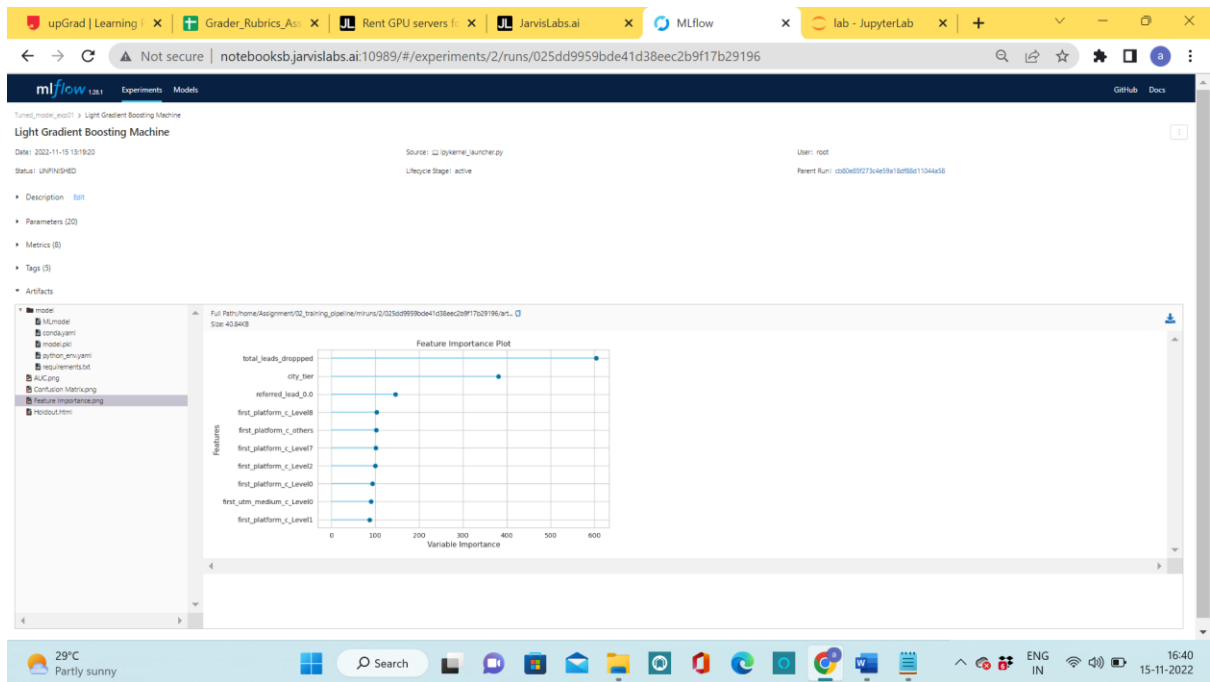
Make Predictions

Predict on a Spark DataFrame:

```
import mlflow
logged_model = 'runs:/025dd9959bde41d38eec2b9f17b29196/model'
```

# Load model as a Spark UDF. Override result\_type if the model does not return double values.





upGrad | Learning | Grader\_Rubrics\_As | Rent GPU servers f | JarvisLabs.ai | MLflow | lab - JupyterLab

Not secure | notebooks.jarvislabs.ai:10989/#/experiments/2/runs/025dd9959bde41d38eec2b9f17b29196

mlflow 1.26.1 Experiments Models

Tuned\_model\_exp01 > Light Gradient Boosting Machine

Date: 2022-11-15 13:19:20 Source: ipykernel\_launcher.py User: root  
Status: UNFINISHED Lifecycle Stage: active Parent Run: cb80e85f273c4e59a18df88d11044a58

Description (60)  
Parameters (20)  
Metrics (8)  
Tags (5)  
Artifacts

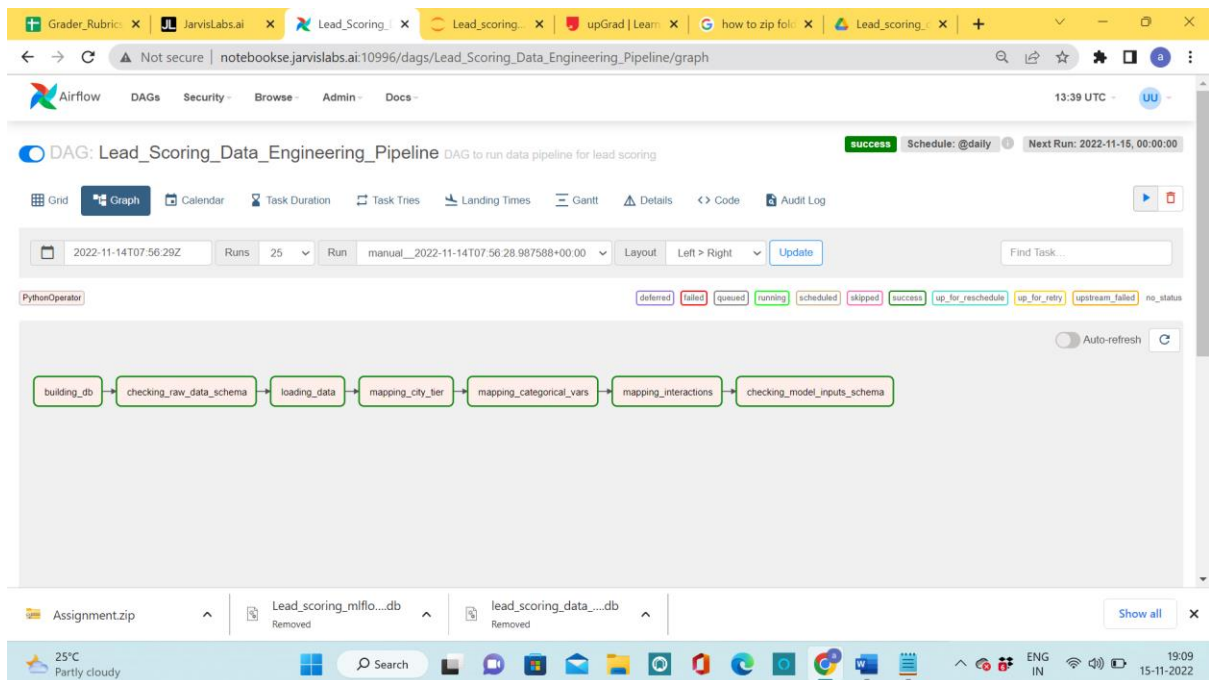
Full Path: /home/Assignment/02\_training\_pipeline/mlruns/2/025dd9959bde41d38eec2b9f17b29196/art...  
Size: 774 B

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	Light Gradient Boosting Machine	0.738	0.8199	0.8317	0.7006	0.7606	0.4759	0.4845

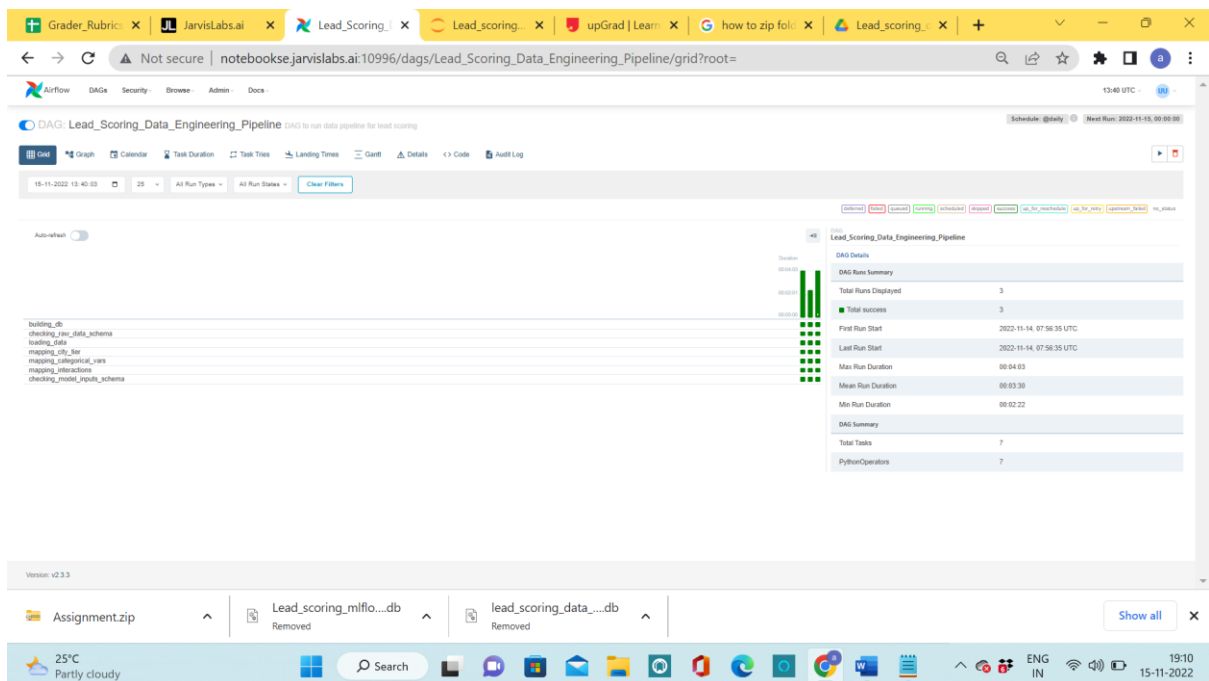
29°C Partly sunny 16:41 15-11-2022

# DATA PIPELINE

## Screenshot of successful execution Airflow DAG in graph



## Screenshot of Airflow UI grid



# TRAINING PIPELINE

screenshot of experiments with all the artifacts visible and also metric AUC

The screenshot shows the MLflow Experiments page for the 'Lead\_Scoring\_Training\_Pipeline' experiment. The interface includes a search bar, a list of experiments, and a table of runs. The table displays the following data:

Start Time	Duration	Run Name	User	Source	Version	Models	Metrics	Parameters
3 hours ago	5.8s	run_LightGB	root	ipykernel...	-	LightGBM/4	auc: 0.747	boosting_type: gbdt, class_weight: None, colsample_bytree: 1.0
1 day ago	4.3s	run_LightGB	root	airflow	-	LightGBM/3	auc: 0.747	boosting_type: gbdt, class_weight: None, colsample_bytree: 1.0
1 day ago	5.3s	run_LightGB	root	airflow	-	LightGBM/2	auc: 0.747	boosting_type: gbdt, class_weight: None, colsample_bytree: 1.0
1 day ago	4.8s	run_LightGB	root	ipykernel...	-	LightGBM/1	auc: 0.747	boosting_type: gbdt, class_weight: None, colsample_bytree: 1.0

The screenshot shows the MLflow Model page for the 'run\_LightGB' model. The page displays the model's metadata, including its source, user, and status. It also shows the model's schema and code snippets for predictions.

**Model schema**

Name	Type
MLmodel	MLmodel
conda.yaml	conda.yaml
model.pkl	model.pkl
python_env.yaml	python_env.yaml
requirements.txt	requirements.txt

**Make Predictions**

```
import mlflow
logged_model = 'runs:/33435c050dac491eb28bb209a216c22/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()
```

Grader\_Rubrics x JarvisLabs.ai x MLflow x lab (2) - Jupyter x upGrad | Learn x Lead\_scoring... x (46) WhatsApp x

Not secure | notebooks.jarvislabs.ai:10989/#/experiments/1/runs/33435c050dac491eb28bbb209a216c22

Lead\_Scoring\_Training\_Pipeline > run\_LightGB

### run\_LightGB

Date: 2022-11-15 19:24:02 Source: ipykernel\_launcher.py User: root  
Duration: 5.8s Status: FINISHED Lifecycle Stage: active

Description Edit

Parameters (20)

Metrics (1)

Name	Value
auc	0.747

Tags

Artifacts

models

- MLmodel
- conda.yaml
- model.pkl
- python\_env.yaml
- requirements.txt

Full Path: /home/Assignment/02\_training\_pipeline/mlruns/1/33435c050dac491eb28bbb209a216c22/artifacts/models

#### 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
logged_model = 'runs:/33435c050dac491eb28bbb209a216c22/model1s'
```

23°C Mostly clear 22:54 15-11-2022

screenshot of model registry with model name and stage as 'production'

Grader\_Rubrics x JarvisLabs.ai x MLflow x lab (2) - Jupyter x upGrad | Learn x Lead\_scoring... x (46) WhatsApp x

Not secure | notebooks.jarvislabs.ai:10989/#/models/LightGBM

mlflow 1.26.1 Experiments Models GitHub Docs

Registered Models > LightGBM

### LightGBM

Created Time: 2022-11-14 20:18:01 Last Modified: 2022-11-15 22:43:32

Description Edit

Tags

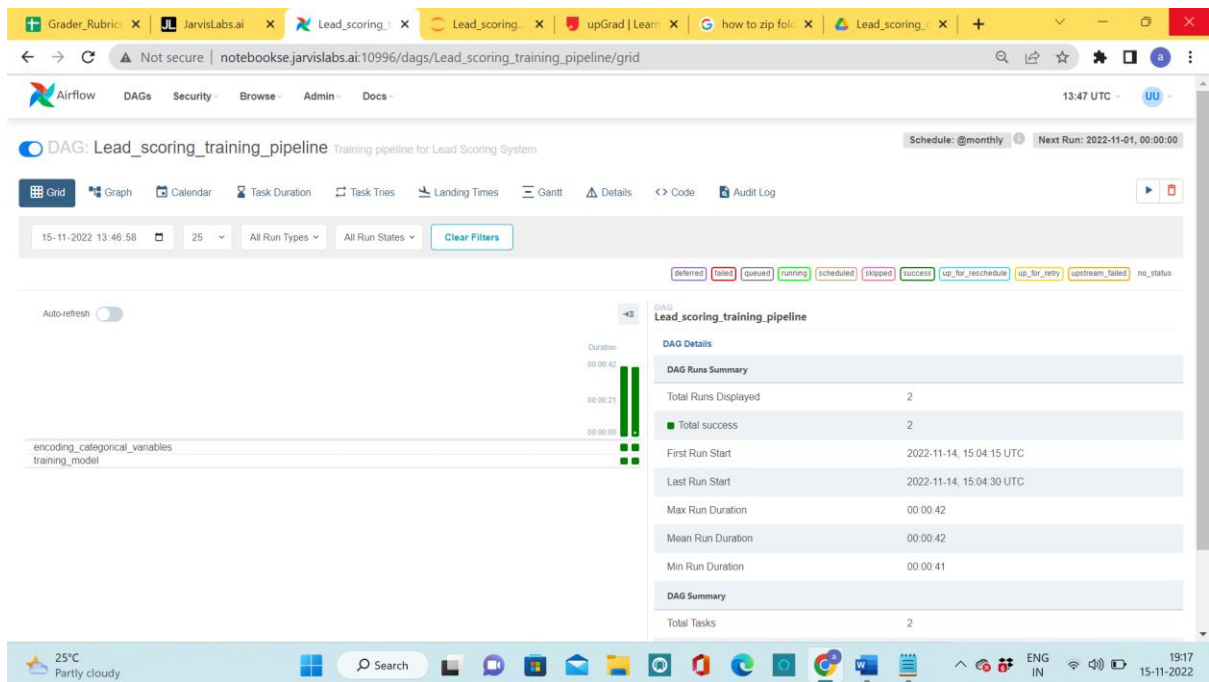
Versions All Active 1 Compare

Version	Registered at	Created by	Stage	Description
<input type="checkbox"/> Version 4	2022-11-15 19:24:07		Production	
<input type="checkbox"/> Version 3	2022-11-14 20:35:11		None	
<input type="checkbox"/> Version 2	2022-11-14 20:34:56		None	
<input type="checkbox"/> Version 1	2022-11-14 20:18:01		None	

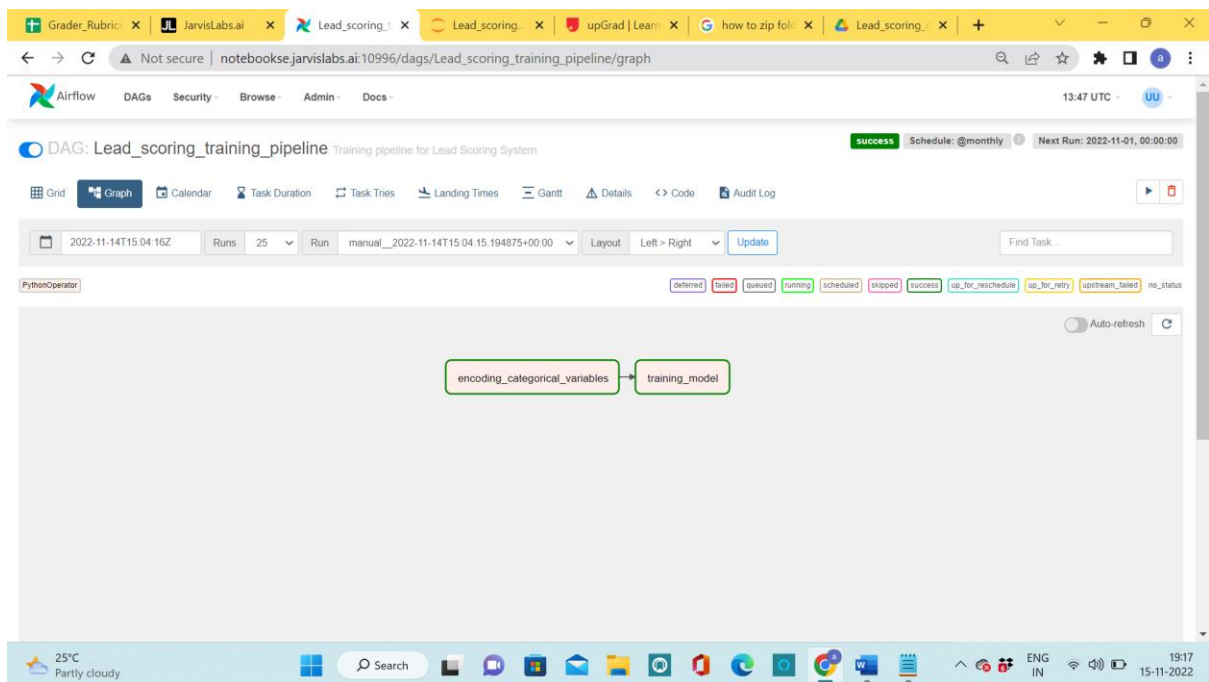
23°C Mostly clear 22:55 15-11-2022



## Screenshot of Airflow UI grid



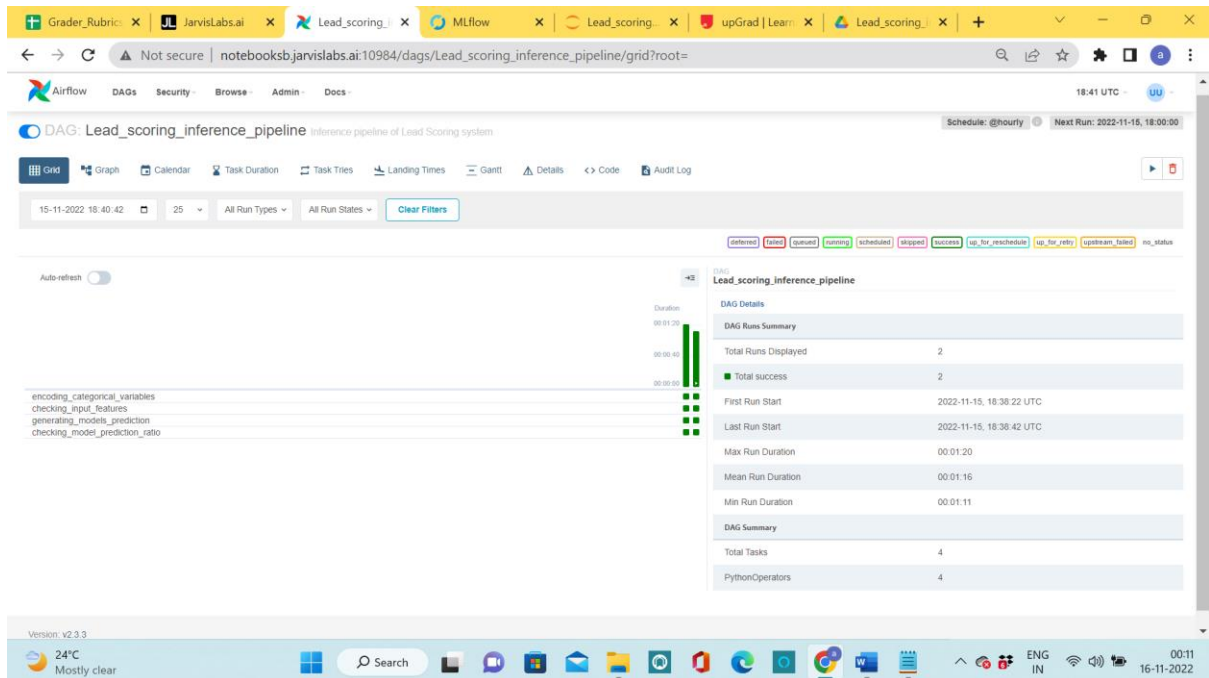
## Screenshot of successful execution Airflow DAG in graph



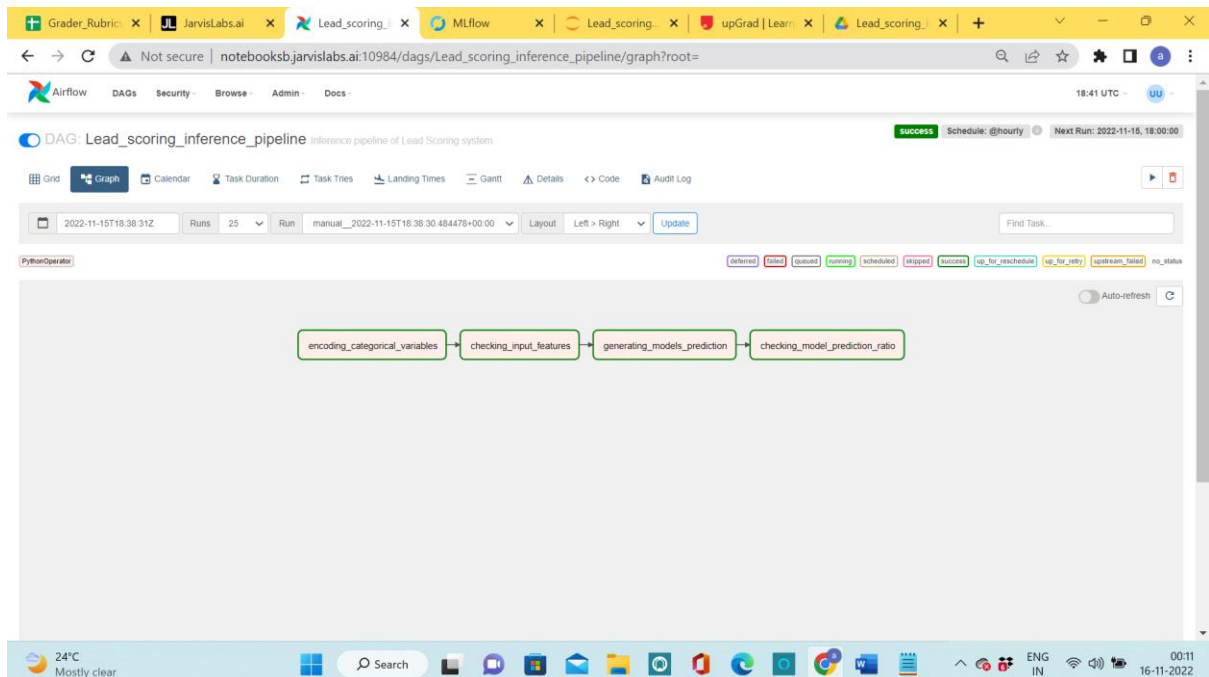


# INFERENCE PIPELINE

## Screenshot of Airflow UI grid



## Screenshot of successful execution Airflow DAG in graph



## Running data cleaning pipeline on leadscoring\_inference.csv

The screenshot shows the Apache Airflow web interface for the DAG 'Lead\_Scoring\_Data\_Engineering\_Pipeline'. The interface is in 'Graph' view, displaying a linear sequence of seven tasks: 'building\_db', 'checking\_raw\_data\_schema', 'loading\_data', 'mapping\_city\_tier', 'mapping\_categorical\_vars', 'mapping\_interactions', and 'checking\_model\_inputs\_schema'. The DAG is currently in a 'running' state, as indicated by the green status bar at the top right. The schedule is set to '@daily', and the next run is scheduled for 2022-11-15, 00:00:00. The interface includes various navigation tabs like Grid, Graph, Calendar, Task Duration, Task Tiles, Landing Times, Gantt, Details, Code, and Audit Log. The bottom of the screen shows a Windows taskbar with the date 16-11-2022 and time 00:40.

The screenshot shows the same Apache Airflow DAG in 'Grid' view. The interface displays a list of task instances and their execution status. A bar chart on the left shows the duration of the tasks. The right-hand side of the screen provides a detailed summary of the DAG's performance, including the number of runs, success/failure counts, and execution times.

DAG Summary	
Total Tasks	7
PythonOperators	7

DAG Details	
DAG Run Summary	
Total Runs Displayed	6
Total success	5
Total failed	1
First Run Start	2022-11-14, 07:56:35 UTC
Last Run Start	2022-11-15, 19:08:54 UTC
Max Run Duration	00:04:03
Mean Run Duration	00:02:43
Min Run Duration	00:01:33

## Running data inference pipeline on cleaned leadscoring\_inference.csv

The screenshot shows the Airflow web interface for the DAG 'Lead\_scoring\_inference\_pipeline'. The interface is in 'Graph' view, displaying a linear workflow with four tasks: 'encoding\_categorical\_variables', 'checking\_input\_features', 'generating\_models\_prediction', and 'checking\_model\_prediction\_ratio'. The DAG is currently in a 'running' state, as indicated by the green status bar at the top right. The schedule is set to '@hourly', and the next run is scheduled for 2022-11-15, 19:00:00. The interface includes various navigation tabs like Grid, Calendar, Task Duration, Task Times, Landing Times, Gantt, Details, Code, and Audit Log. The bottom of the screen shows a Windows taskbar with the date 16-11-2022 and time 00:42.

The screenshot shows the Airflow web interface for the DAG 'Lead\_scoring\_inference\_pipeline' in 'Grid' view. The interface displays a list of DAG runs with columns for 'Duration', 'Start Time', and 'End Time'. A bar chart on the right side of the grid shows the duration of the runs. The DAG is currently in a 'running' state, as indicated by the green status bar at the top right. The schedule is set to '@hourly', and the next run is scheduled for 2022-11-15, 19:00:00. The interface includes various navigation tabs like Grid, Calendar, Task Duration, Task Times, Landing Times, Gantt, Details, Code, and Audit Log. The bottom of the screen shows a Windows taskbar with the date 16-11-2022 and time 00:42.

Task	Duration
encoding_categorical_variables	00:01:20
checking_input_features	00:01:02
generating_models_prediction	00:00:27
checking_model_prediction_ratio	00:00:00

DAG Summary	
Total Runs Displayed	4
Total success	4
First Run Start	2022-11-15, 18:38:22 UTC
Last Run Start	2022-11-15, 19:11:34 UTC
Max Run Duration	00:01:20
Mean Run Duration	00:01:02
Min Run Duration	00:00:27

DAG Summary	
Total Tasks	4
PythonOperators	4