

# AIRFLOW HOMEPAGE

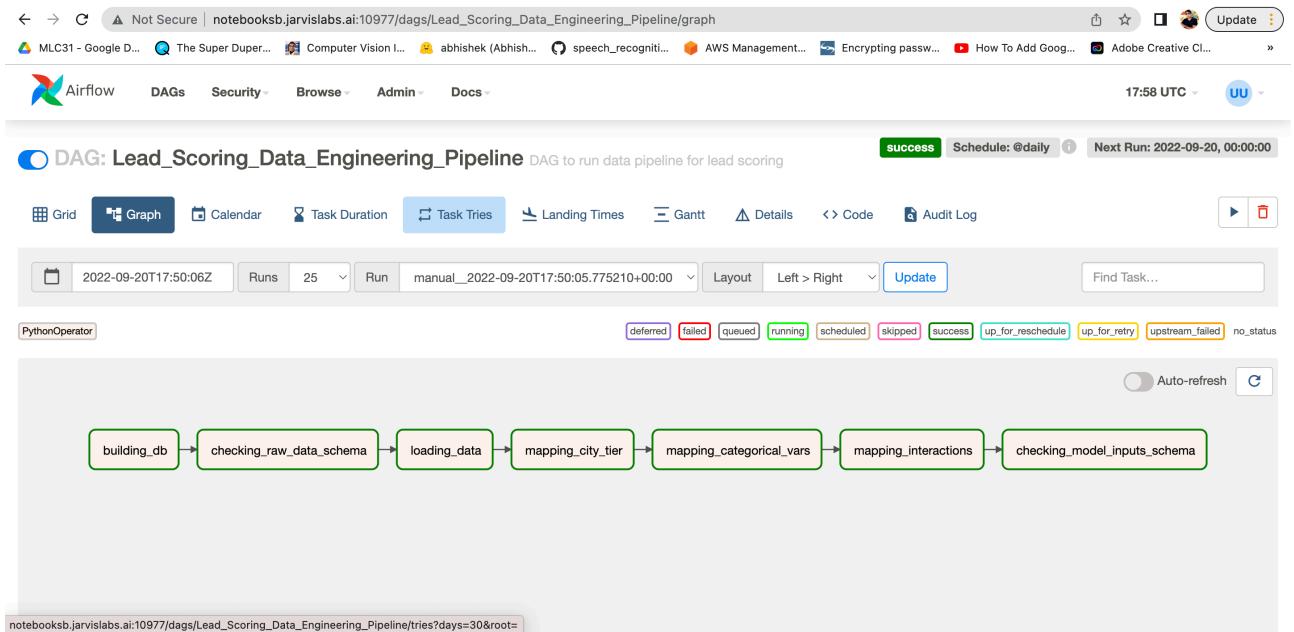
The screenshot shows the Airflow homepage with the title "AIRFLOW HOMEPAGE". At the top, there is a navigation bar with links like "Not Secure", "notebooksb.jarvislabs.ai:10969/home?status=active", and various user and system status indicators. Below the navigation bar, the main content area is titled "DAGs". It features a table listing three active DAGs:

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow	4 (green), 3 (red)	@daily	2022-09-20, 06:02:24	2022-09-20, 00:00:00	7 (green), 1 (red)
Lead_scoring_inference_pipeline	airflow	3 (green), 8 (red)	@hourly	2022-09-20, 06:05:41	2022-09-20, 06:00:00	4 (green), 1 (red)
Lead_scoring_training_pipeline	airflow	5 (green), 1 (red)	@monthly	2022-09-20, 06:04:14	2022-09-01, 00:00:00	2 (green), 1 (red)

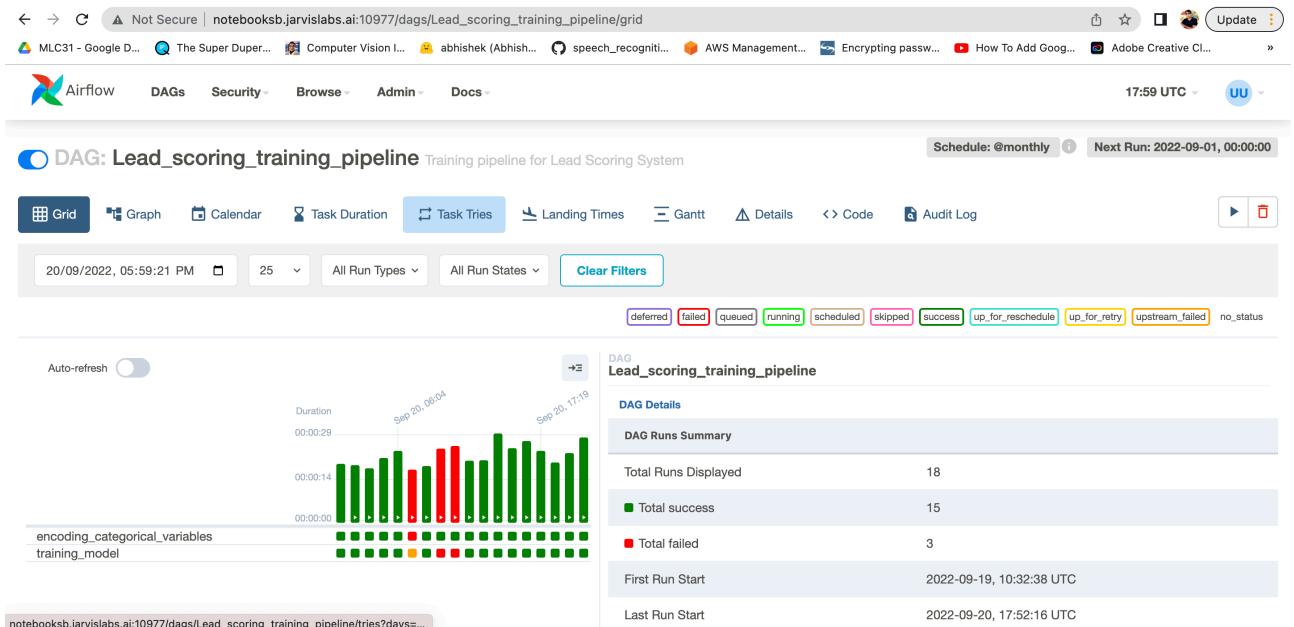
Below the table, there is a pagination control showing "Showing 1-3 of 3 DAGs". At the bottom of the page, a note says "Version: v2.3.3".

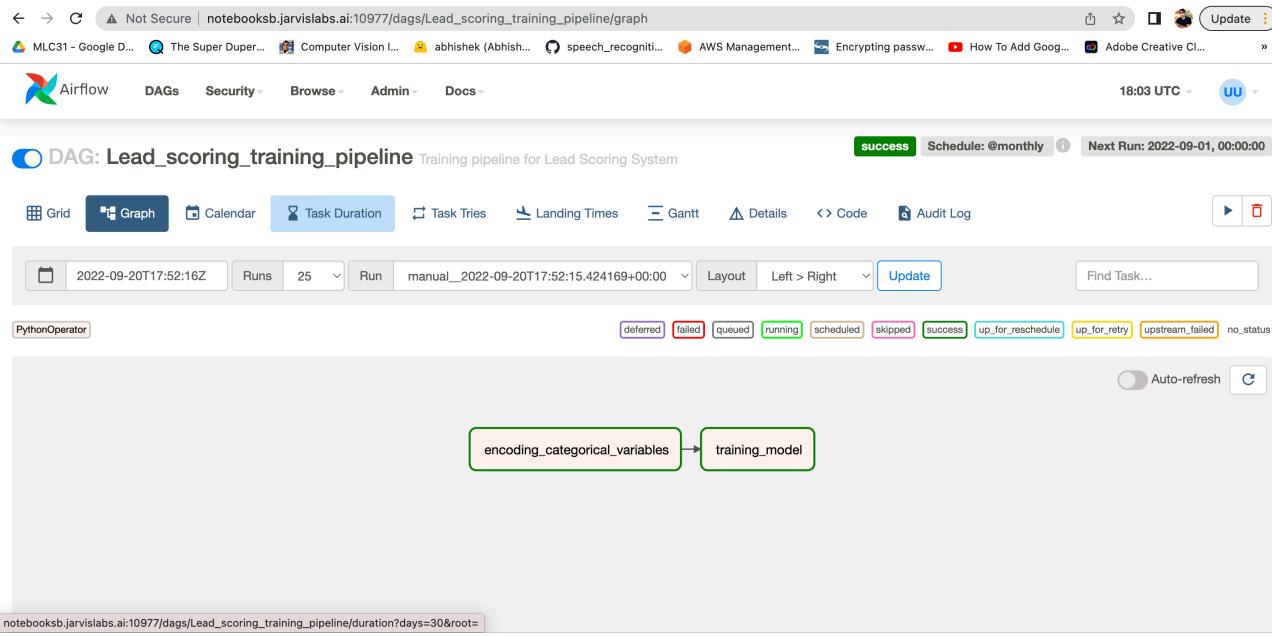
# DATA PIPELINE

The screenshot shows the Data Pipeline interface for the "Lead\_Scoring\_Data\_Engineering\_Pipeline". At the top, there is a navigation bar with links like "Airflow", "DAGs", "Security", "Browse", "Admin", "Docs", and a timestamp "17:57 UTC". Below the navigation bar, the main content area is titled "DAG: Lead\_Scoring\_Data\_Engineering\_Pipeline". It displays the DAG's details, including its schedule (@daily) and next run (2022-09-20, 00:00:00). The interface includes various tabs: Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, and Audit Log. The "Task Tries" tab is selected, showing a timeline from 2022-09-18 to 2022-09-20. The tasks are represented by colored bars: red for failed, green for success, and grey for other states. A legend at the bottom right shows the color coding for different states. On the left, a sidebar lists the tasks: "building\_db", "checking\_raw\_data\_schema", "loading\_data", "mapping\_city\_tier", and "mapping\_categorical\_vars". The bottom of the page shows a URL: "notebooksb.jarvislabs.ai:10977/dags/Lead\_Scoring\_Data\_Engineering\_Pipeline/tries?days=30".

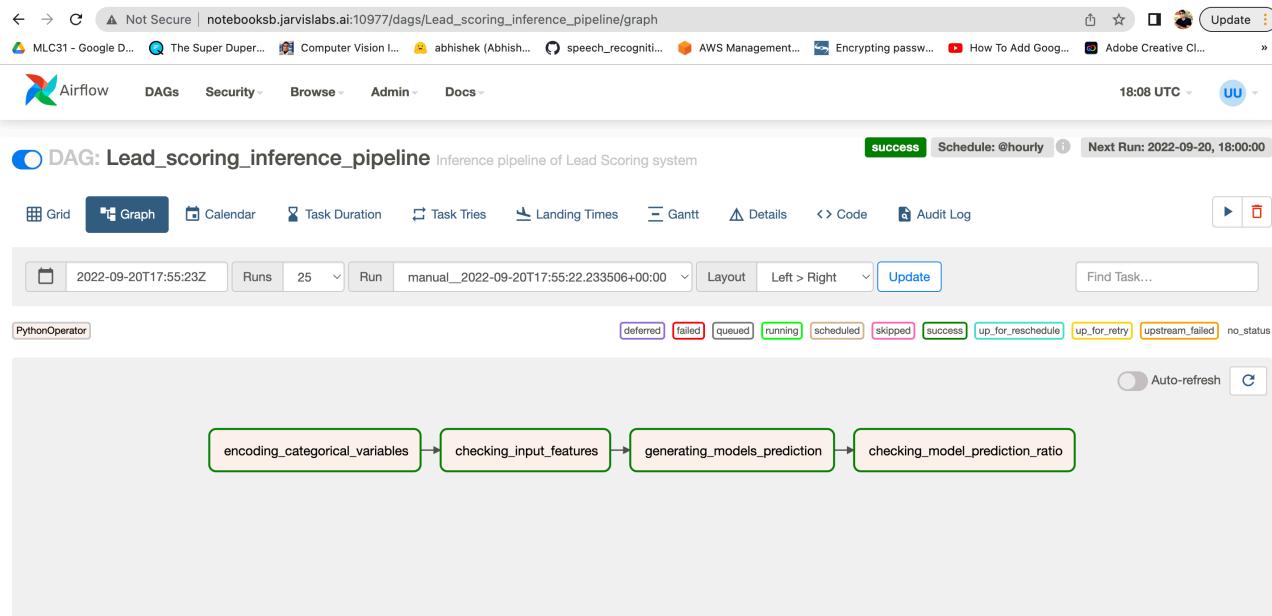
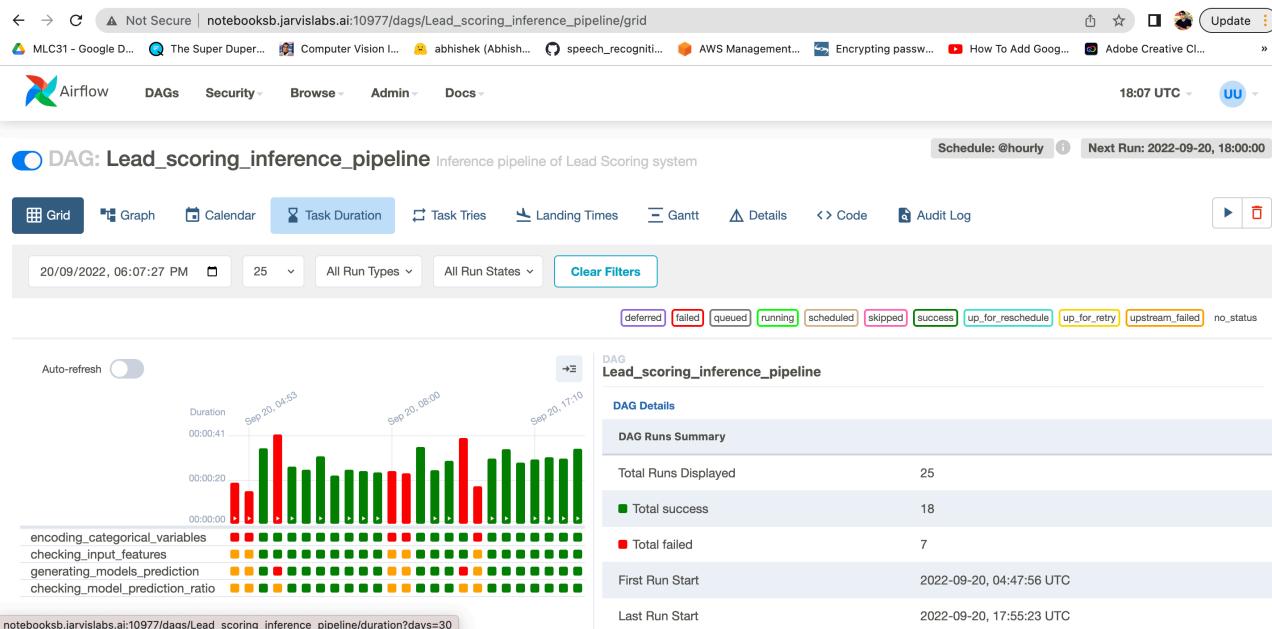


# Training Pipeline

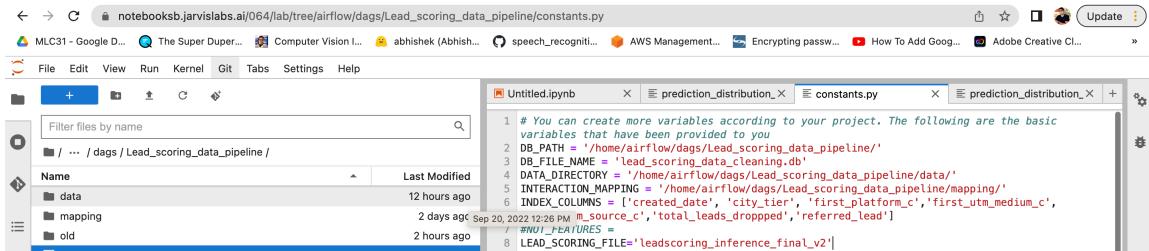




## Inference Pipeline

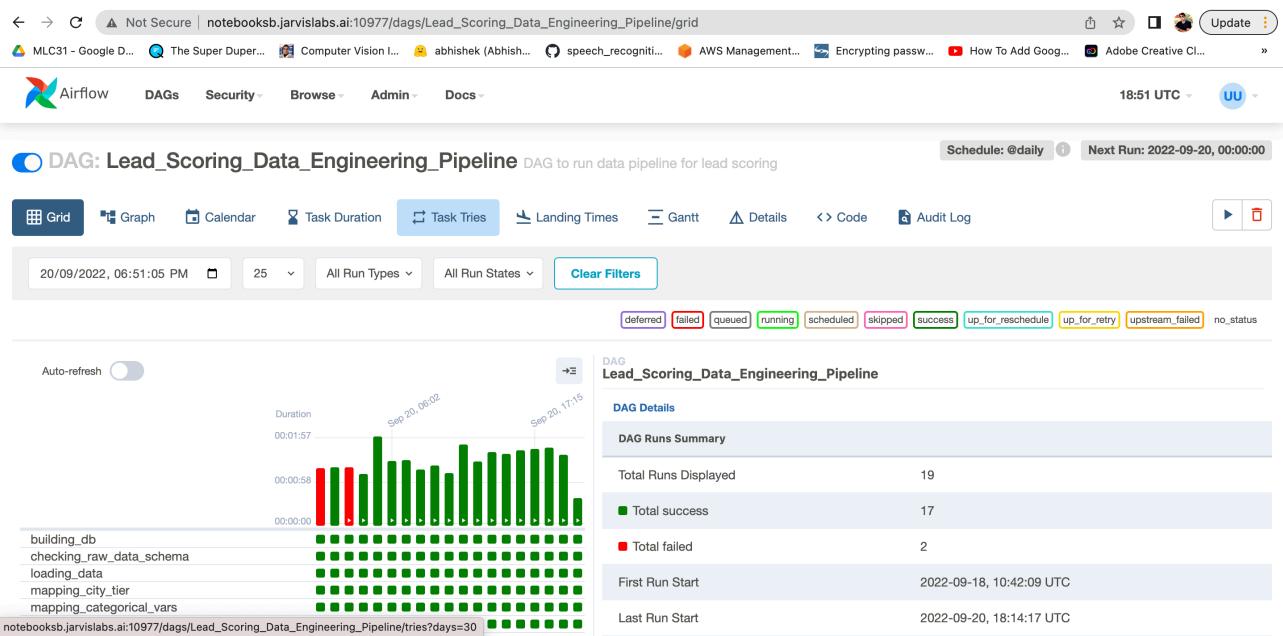


# After making changes in Data Pipeline for Inference file

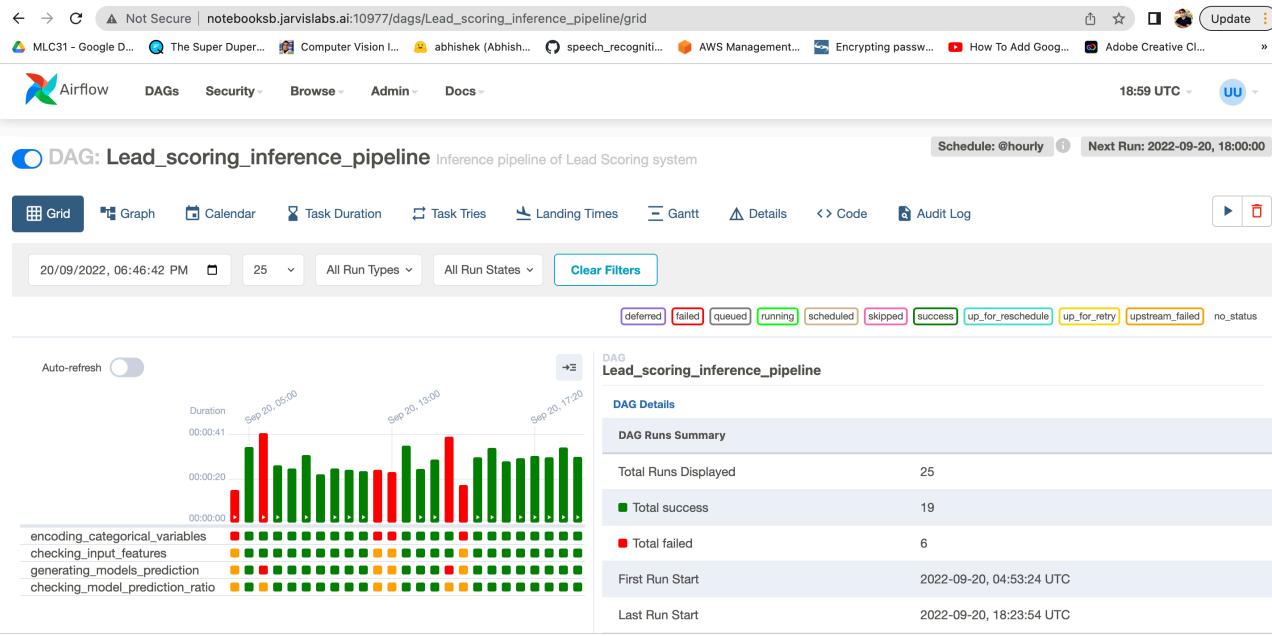


```
# You can create more variables according to your project. The following are the basic variables that have been provided to you
DB_PATH = '/home/airflow/dags/lead_scoring_data_pipeline/'
FILE_TYPE = 'parquet'
INDEX_COLUMNS = ['created_date', 'city_tier', 'first_platform_c', 'first_utm_medium_c', 'referring_source_c', 'total_leads_dropped', 'referred_lead']
INTERACTION_MAPPING = '/home/airflow/dags/lead_scoring_data_pipeline/mapping/mapping.csv'
DATA_DIRECTORY = '/home/airflow/dags/lead_scoring_data_pipeline/data'
# #DFT FEATURES =
LEAD_SCORING_FILE='leadscore_inference_final_v2.parquet'
```

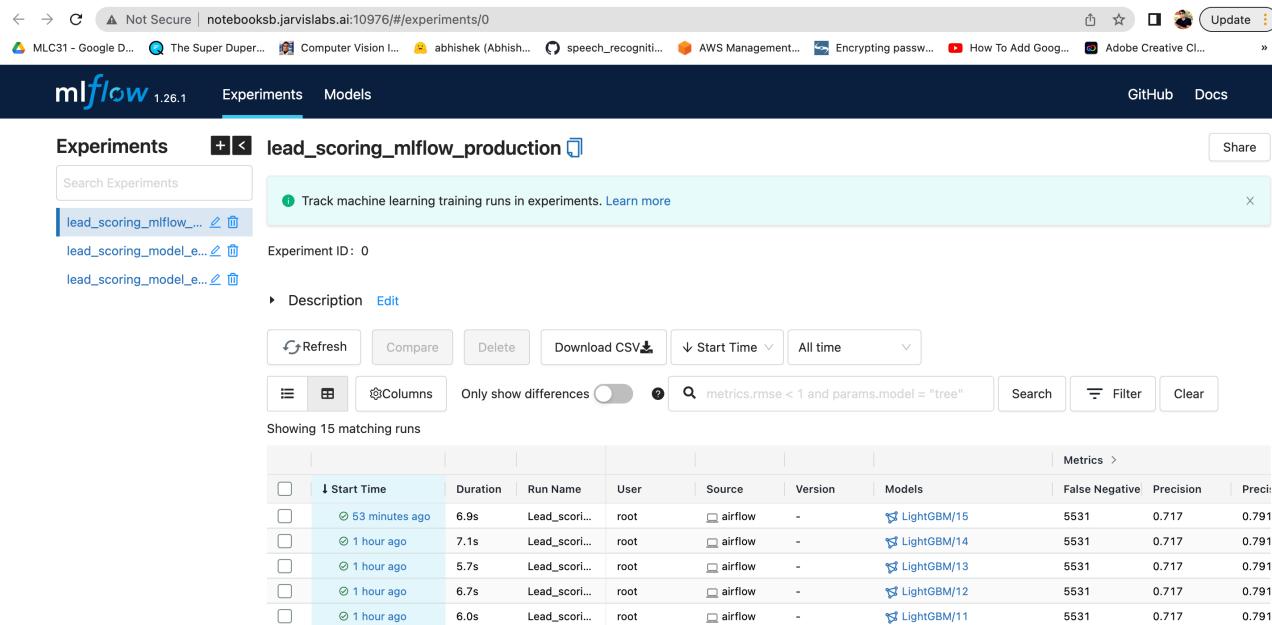
## Data Pipeline



## Data Inference



## ML Flow Server



The screenshot shows the mlflow web interface at the URL <https://notebooksb.jarvislabs.ai:10976/#/models>. The page title is "Registered Models". A header bar includes links for "Experiments" and "Models", and buttons for "GitHub" and "Docs". The main content area displays a table of registered models:

Name	Latest Version	Staging	Production	Last Modified	Tags
LightGBM	Version 15	-	Version 15	2022-09-20 23:25:01	-

Below the table are navigation buttons for pages and a search/filter bar.

## Models:

The screenshot shows the mlflow web interface at the URL <https://notebooksb.jarvislabs.ai:10976/#/experiments/0/runs/2fc45d56b03244d1882cd9035b8c737f/artifacts/1>. The page title is "MLflow Model". The left sidebar has sections for "Tags" and "Artifacts". The "Artifacts" section is expanded, showing a folder named "models" containing files: MLmodel, conda.yaml, model.pkl, python\_env.yaml, and requirements.txt. The main content area displays the "MLflow Model" details:

**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 <a href="#">MLflow docs</a> for how to include input and output schema with your model.	

**Make Predictions**  
Predict on a Spark DataFrame:

```
import mlflow
logged_model = 'runs:/2fc45d56b03244d1882cd9035b8c737f/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:/2fc45d56b03244d1882cd9035b8c737f/models'
```

# Metrics

The screenshot shows the MLflow UI for an experiment named 'Lead\_scoring\_mlflow\_production'. The experiment was run on 2022-09-20 at 23:22:33 by user root. It has a duration of 6.9s and is in an active lifecycle stage. The status is FINISHED. The UI displays 12 metrics:

Name	Value
False Negative	0.531
Precision	0.717
Precision_0	0.791
Precision_1	0.674
Recall	0.732
Recall_0	0.588
Recall_1	0.846
True Negative	20946
f1_0	0.675
f1_1	0.75
roc_auc	0.717
test_accuracy	0.717

Below the metrics, there are sections for Tags and Artifacts. The Artifacts section shows a folder structure for a 'MLflow Model' containing files like MLmodel, conda.yaml, model.pkl, python-env.yaml, and requirements.txt.

# Parameters

The screenshot shows the MLflow UI for the same experiment. The parameters listed are:

Name	Value
boosting_type	gbdt
class_weight	None
criterion	gini
max_depth	-1
max_features	None
max_leaf_nodes	None
max_samples	None
min_child_weight	0.001
min_split_gain	0.0
n_estimators	100
n_jobs	-1
num_leaves	31
objective	None
random_state	42
reg_alpha	0.0
reg_lambda	0.0
silent	warn
subsample	1.0
subsample_for_bin	200000
subsample_freq	0

Below the parameters, there are sections for Metrics (12), Tags, and Artifacts.

## Please Note : - Regarding Unit Test and Scripts Files

- Data Pipeline Unit test ipynb file location : Lead\_scoring\_data\_pipeline/AssignmentFolderFiles/scripts/test\_utils\_and\_validation\_check.ipynb
- Data Training pipeline Unit test ipynb file location : Lead\_scoring\_training\_pipeline/AssignmentFolderFiles/Scripts/TestTrainingPipeline.ipynb
- Data Inference pipeline Unit test ipynb file : Lead\_scoring\_inference\_pipeline/AssignmentFolderFiles/scripts/Test\_Inference\_Function.ipynb

All the relevant Scripts files are placed in AssignmentFolder/scripts under respective Pipelines folder