SCREENSHOT Document for UpGrad's Assignment Submission

Prepared By,

Sanjeev Surendran

Contents

Airflow Main UI Page 4
Initial run of Lead_Scoring_Data_Engineering_Pipeline 4
Initial run of Lead_scoring_training_pipeline 6
Initial run of Lead_scoring_inference_pipeline 7
Next run of Lead_Scoring_Data_Engineering_Pipeline 9
Next run of Lead_scoring_inference_pipeline 10
MLFlow of Baseline_model_exp01 12
MLFlow of Baseline_model_exp02 13
MLFlow of Lead_scoring_mlflow_production 14
Pytest of Data Pipeline 19

Table of Figures

Figure 1: Airflow Main UI Page 4
Figure 2: Grid view of Lead_Scoring_Data_Engineering_Pipeline 5
Figure 3: Graph view of Lead_Scoring_Data_Engineering_Pipeline 5
Figure 4: Grid view of Lead_scoring_training_pipeline 6
Figure 5: Graph view of Lead_scoring_training_pipeline 7
Figure 6: Grid view of Lead_scoring_inference_pipeline 8
Figure 7: Graph view of Lead_scoring_inference_pipeline 8
Figure 8: Grid view of Lead_Scoring_Data_Engineering_Pipeline 9
Figure 9: Graph view of Lead_Scoring_Data_Engineering_Pipeline 10
Figure 10: Grid view of Lead_scoring_inference_pipeline 11
Figure 11: Graph view of Lead_scoring_inference_pipeline 12
Figure 12: Main MLFlow UI of Baseline_model_exp01 12
Figure 13: MLFlow Artifacts of Baseline_model_exp01 13
Figure 14: Main MLFlow UI of Baseline_model_exp02 13
Figure 15: MLFlow Artifacts of Baseline_model_exp02 14
Figure 16: Main MLFlow UI of Lead_scoring_mlflow_production 15
Figure 17: MLFlow Artifacts of Version 1 model 16
Figure 18: MLFlow Artifacts of Version 2 model 17
Figure 19: MLFlow Artifacts of Version 3 model 19
Figure 20: Version 3 model registered as Production Stage 19
Figure 21: Pytest Test Results 19

Airflow Main UI Page

This is the main Airflow UI page that contains all the registered DAGS. We can see our 3 DAGS:

- Lead_Scoring_Data_Engineering_Pipeline
- Lead_scoring_training_pipeline
- Lead_scoring_inference_pipeline

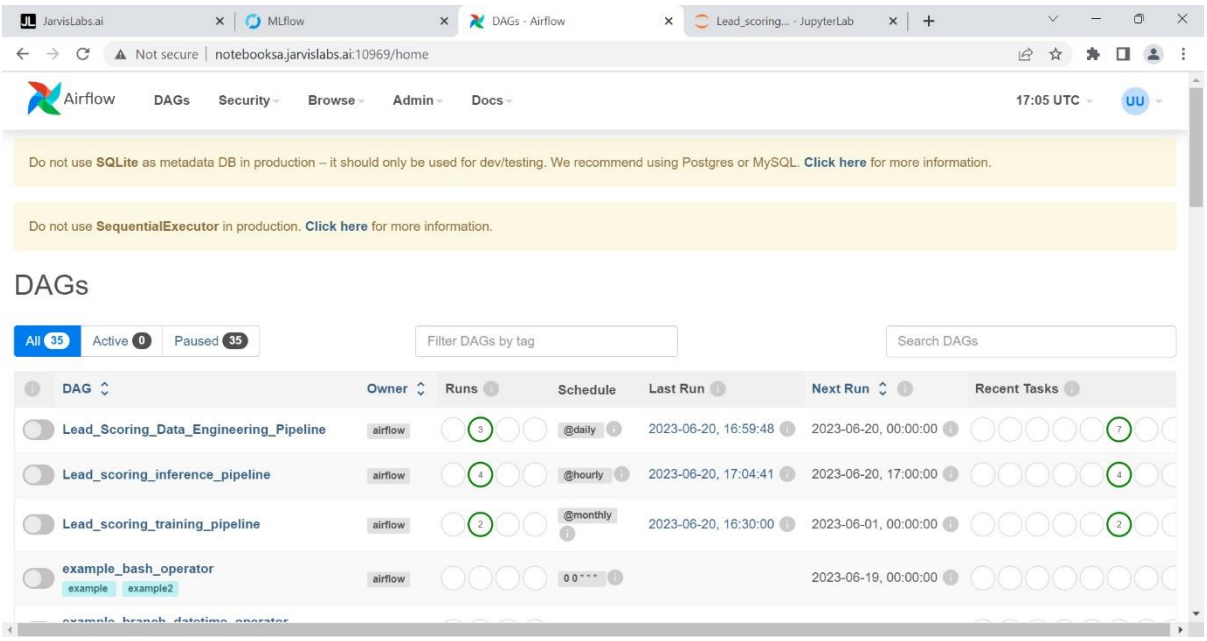

Figure 1: Airflow Main UI Page

Initial run of Lead_Scoring_Data_Engineering_Pipeline

We have triggered manual run of Lead_Scoring_Data_Engineering_Pipeline for data engineering and cleaning data.

Both manual and scheduled runs can be seen in picture. Manual runs have white coloured arrow on top of green bars in Grid view.

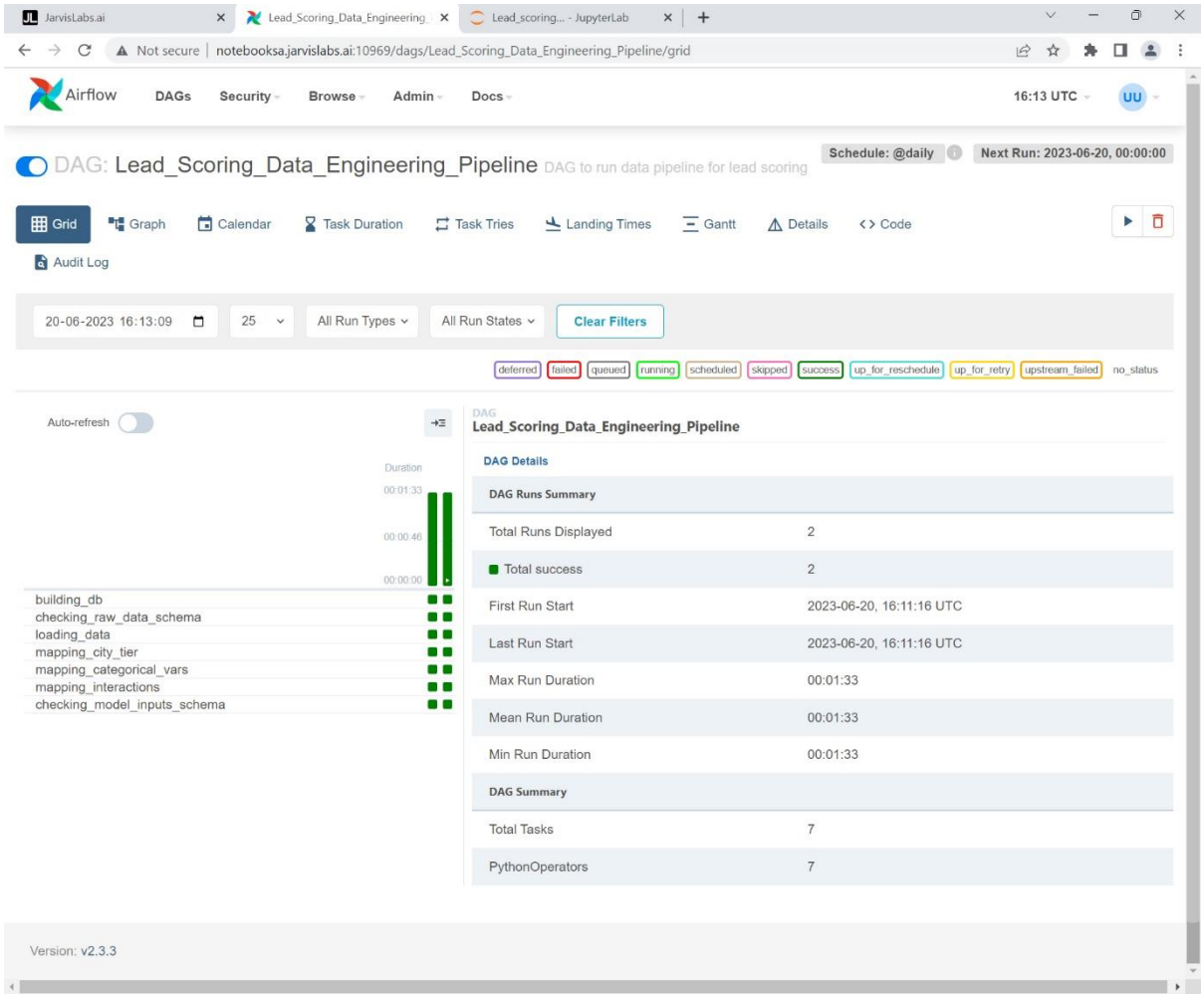

Figure 2: Grid view of Lead_Scoring_Data_Engineering_Pipeline

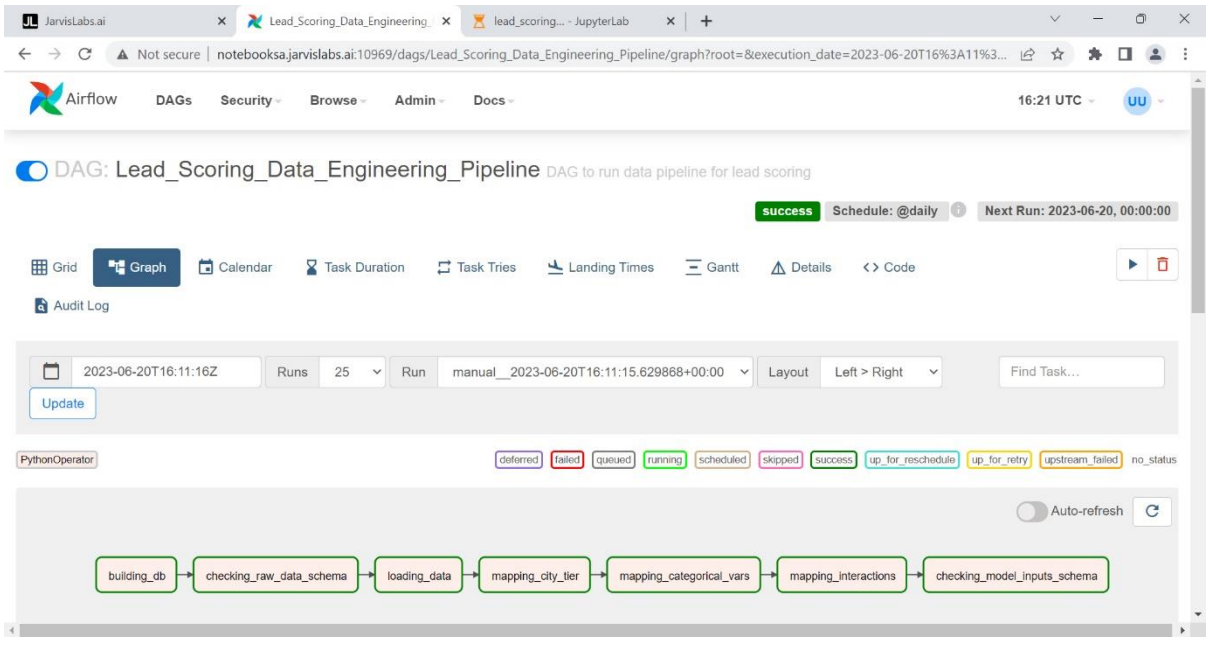

Figure 3: Graph view of Lead_Scoring_Data_Engineering_Pipeline

Initial run of Lead_scoring_training_pipeline

We have triggered manual run of Lead_scoring_training_pipeline for training cleaned data.

Both manual and scheduled runs can be seen in picture. Manual runs have white coloured arrow on top of green bars in Grid view.

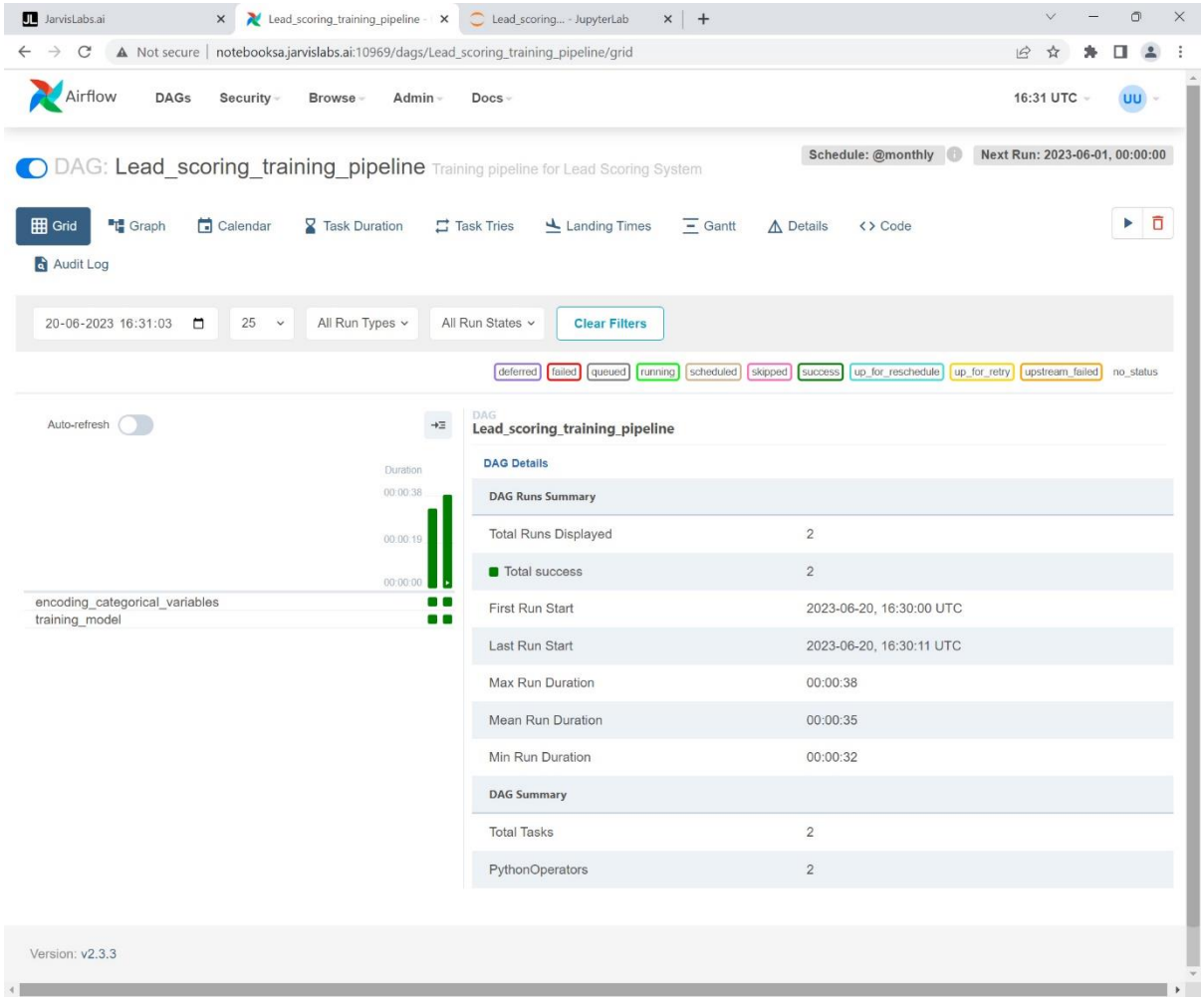

Figure 4: Grid view of Lead_scoring_training_pipeline

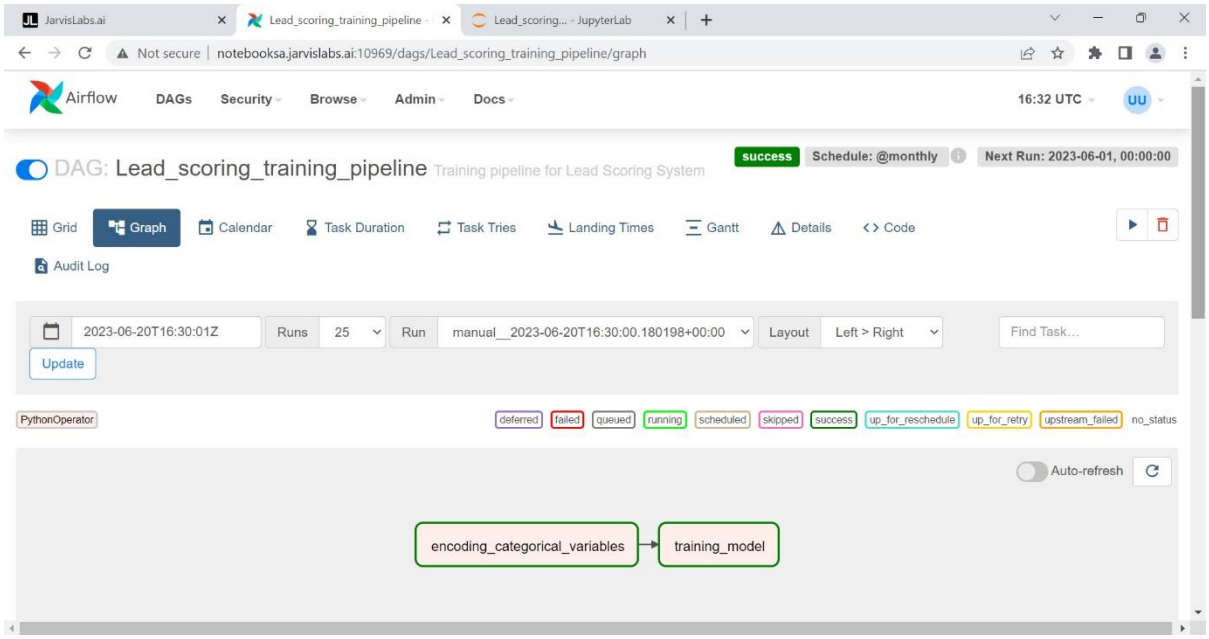

Figure 5: Graph view of Lead_scoring_training_pipeline

Initial run of Lead_scoring_inference_pipeline

We have triggered manual run of Lead_scoring_inference_pipeline for inferencing incoming data and predict outcome.

Both manual and scheduled runs can be seen in picture. Manual runs have white coloured arrow on top of green bars in Grid view.

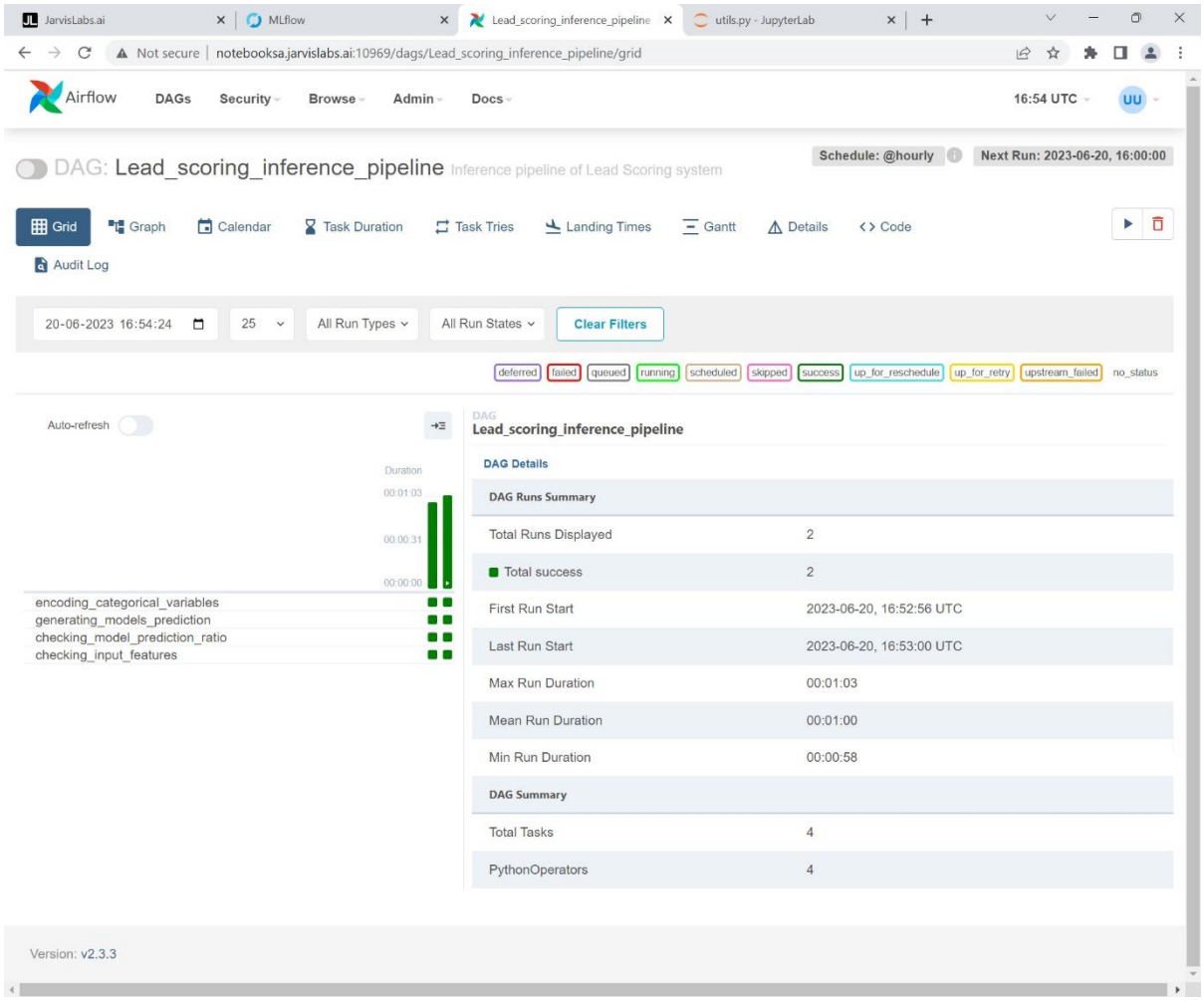

Figure 6: Grid view of Lead_scoring_inference_pipeline

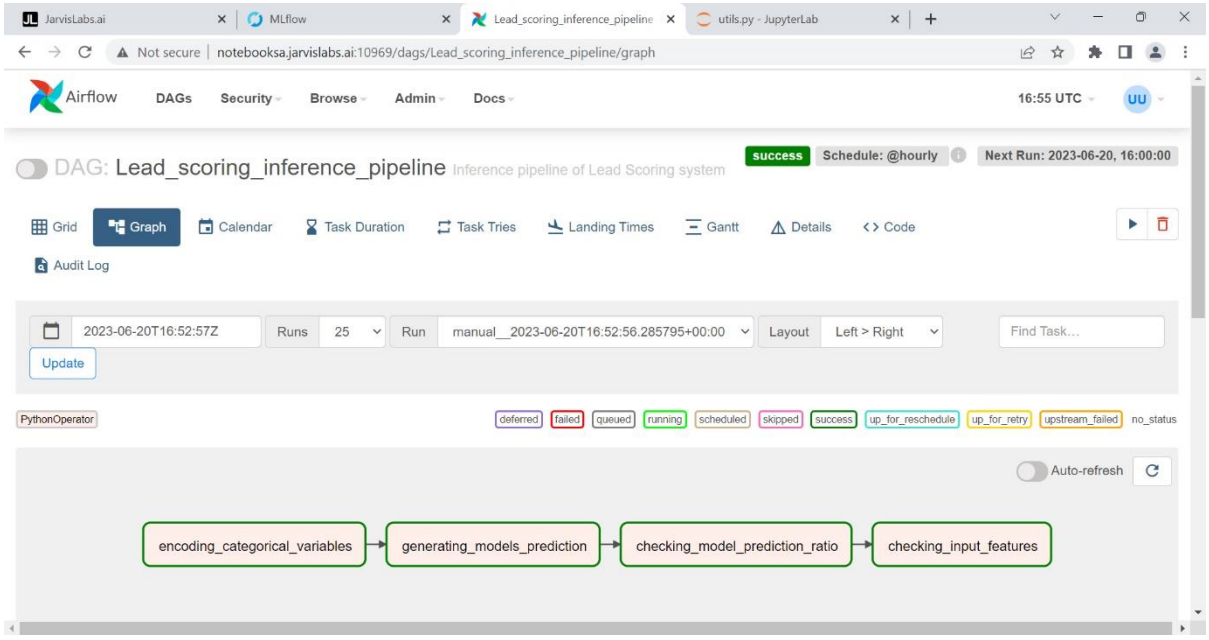

Figure 7: Graph view of Lead_scoring_inference_pipeline

Next run of Lead_Scoring_Data_Engineering_Pipeline

We have triggered manual run of Lead_Scoring_Data_Engineering_Pipeline for cleaning **leadscoring_inference.csv**.

Both manual and scheduled runs can be seen in picture. Manual runs have white coloured arrow on top of green bars in Grid view.

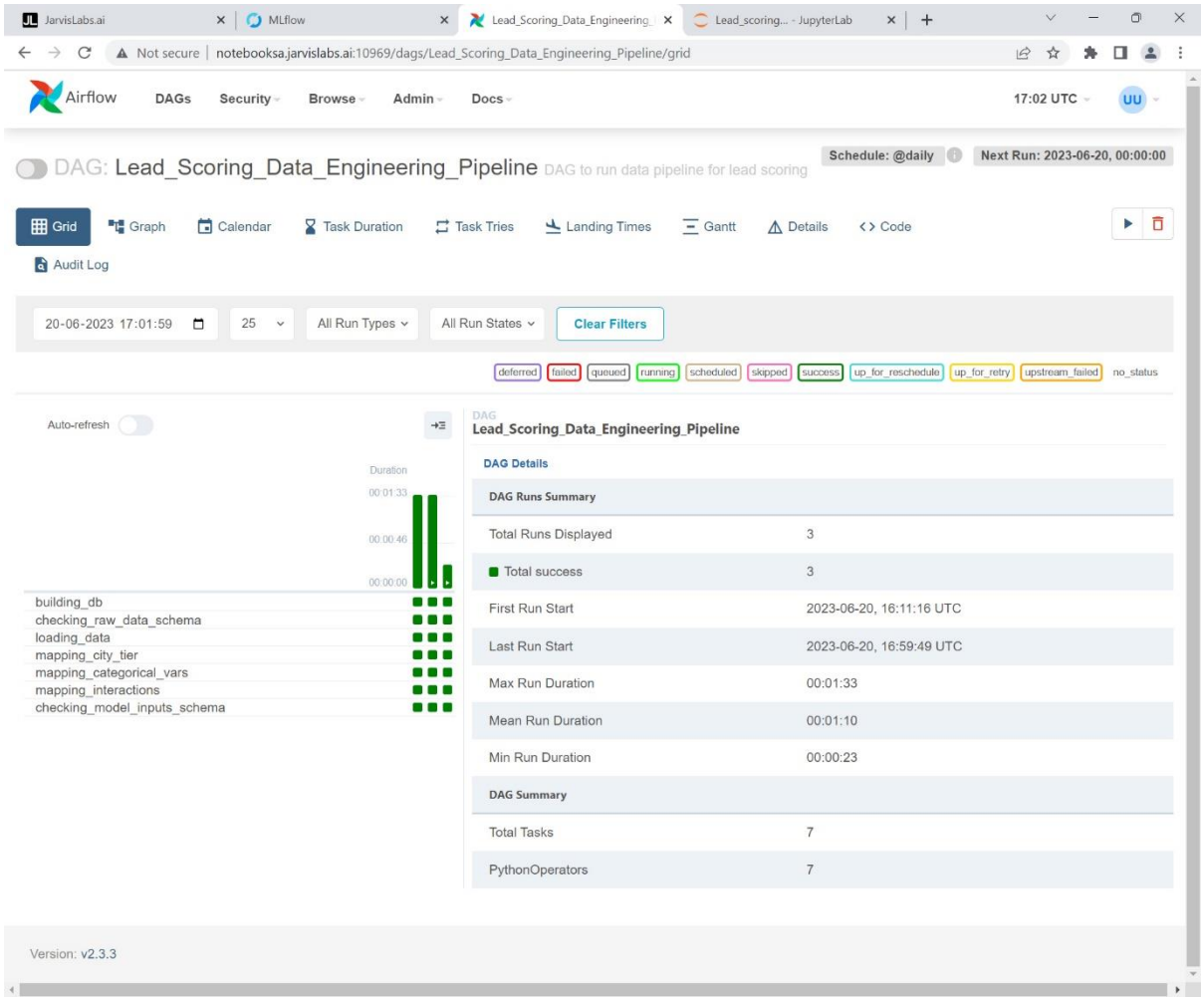

Figure 8: Grid view of Lead_Scoring_Data_Engineering_Pipeline

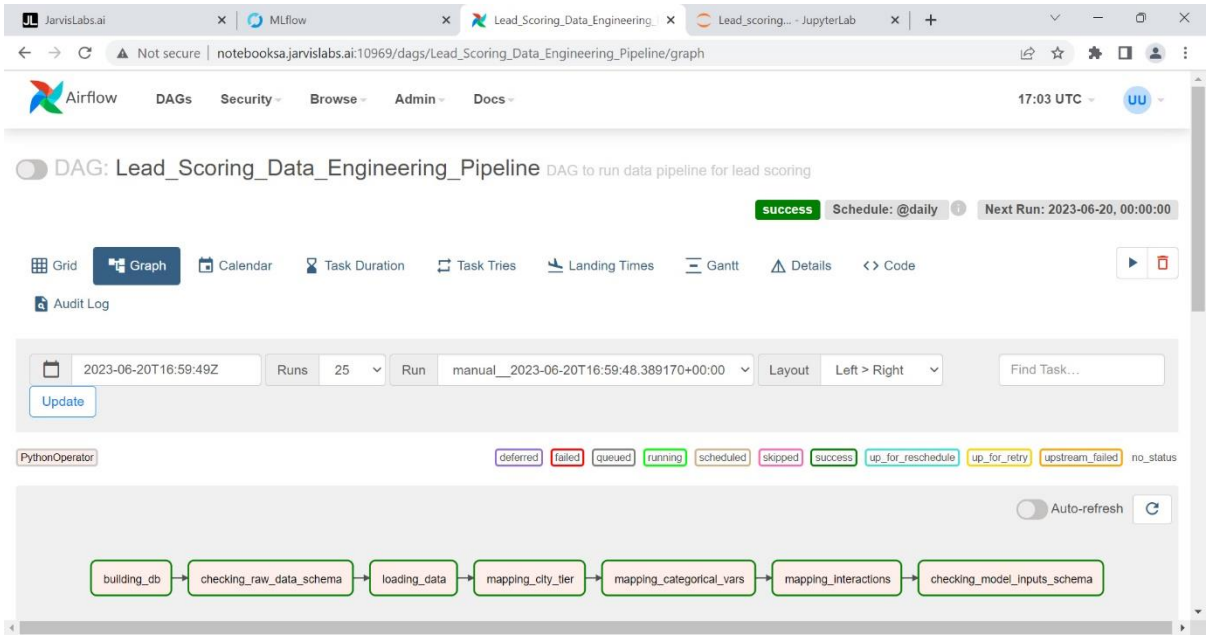

Figure 9: Graph view of Lead_Scoring_Data_Engineering_Pipeline

Next run of Lead_scoring_inference_pipeline

We have triggered manual run of Lead_scoring_inference_pipeline for inferencing incoming data and predict outcome.

Both manual and scheduled runs can be seen in picture. Manual runs have white coloured arrow on top of green bars in Grid view.

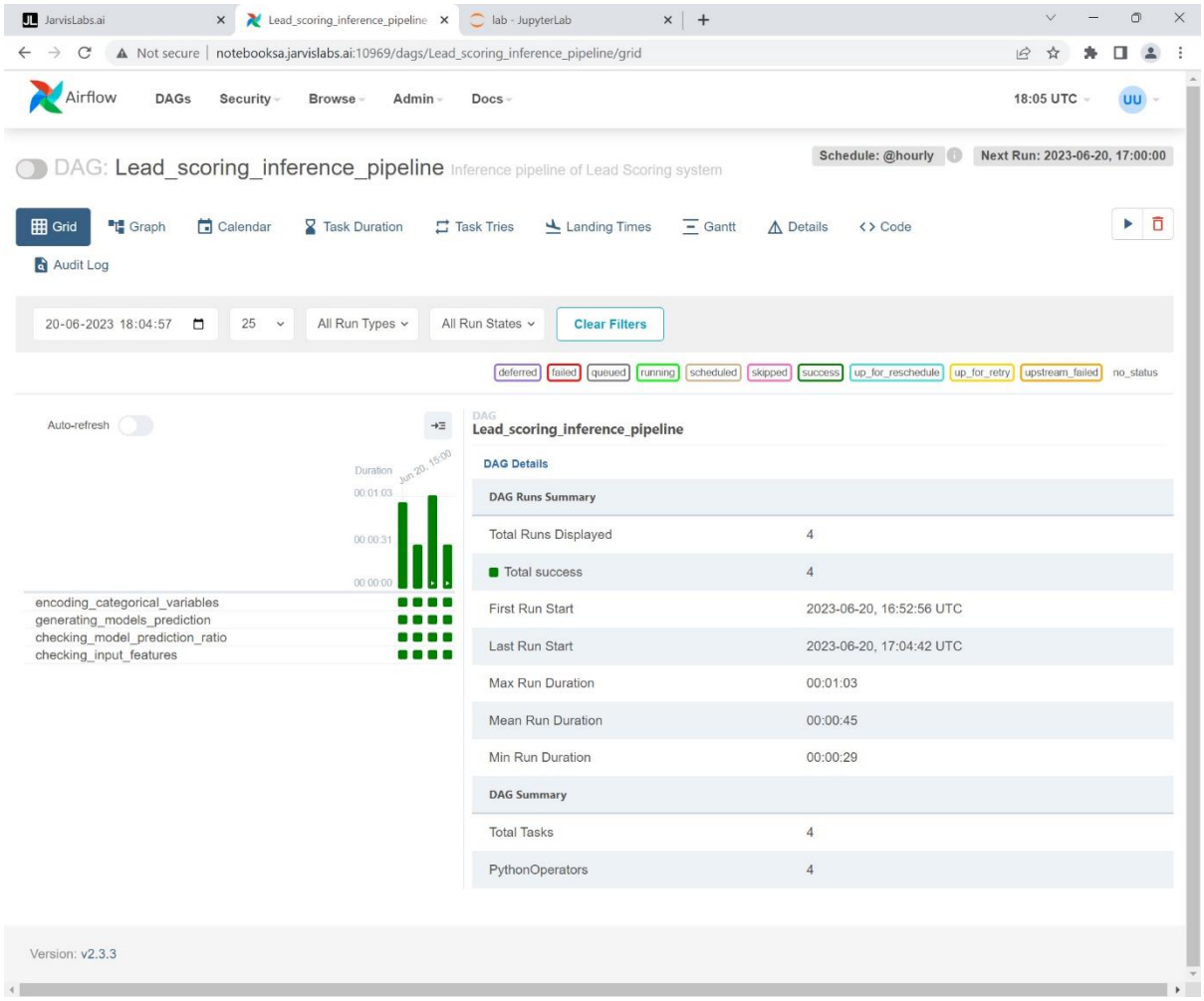

Figure 10: Grid view of Lead_scoring_inference_pipeline

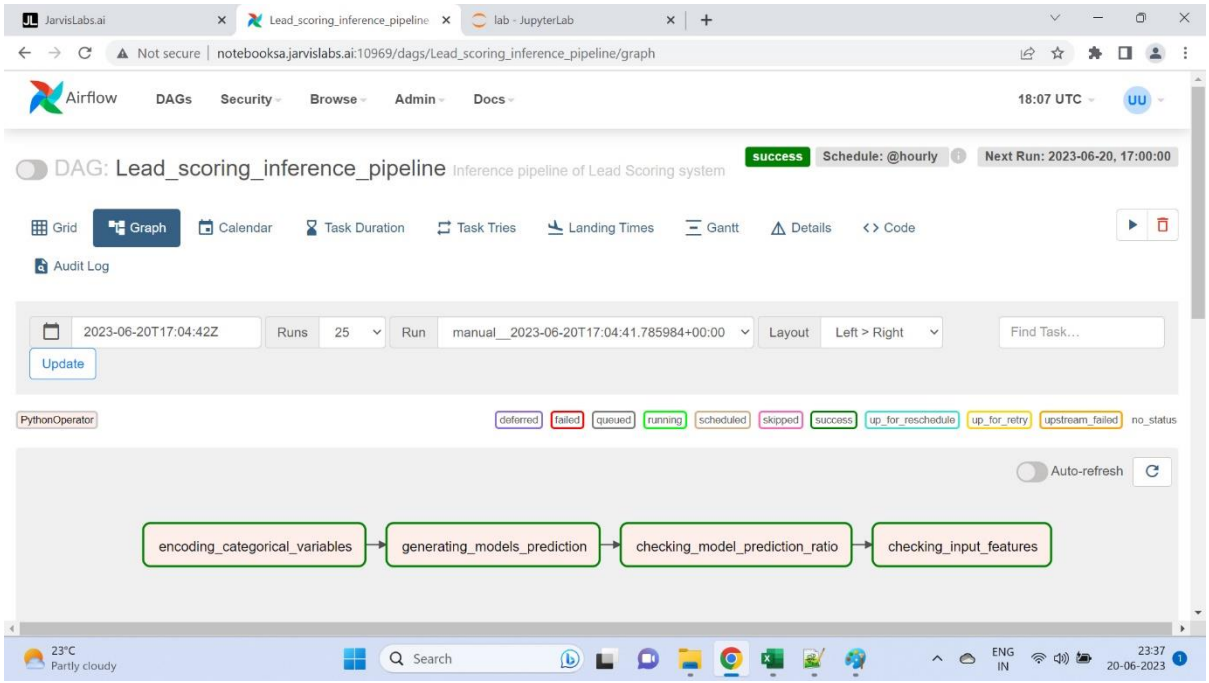

Figure 11: Graph view of Lead_scoring_inference_pipeline

MLFlow of Baseline_model_exp01

Baseline_model_exp01 is the initial pycaret setup experiment done for training model.

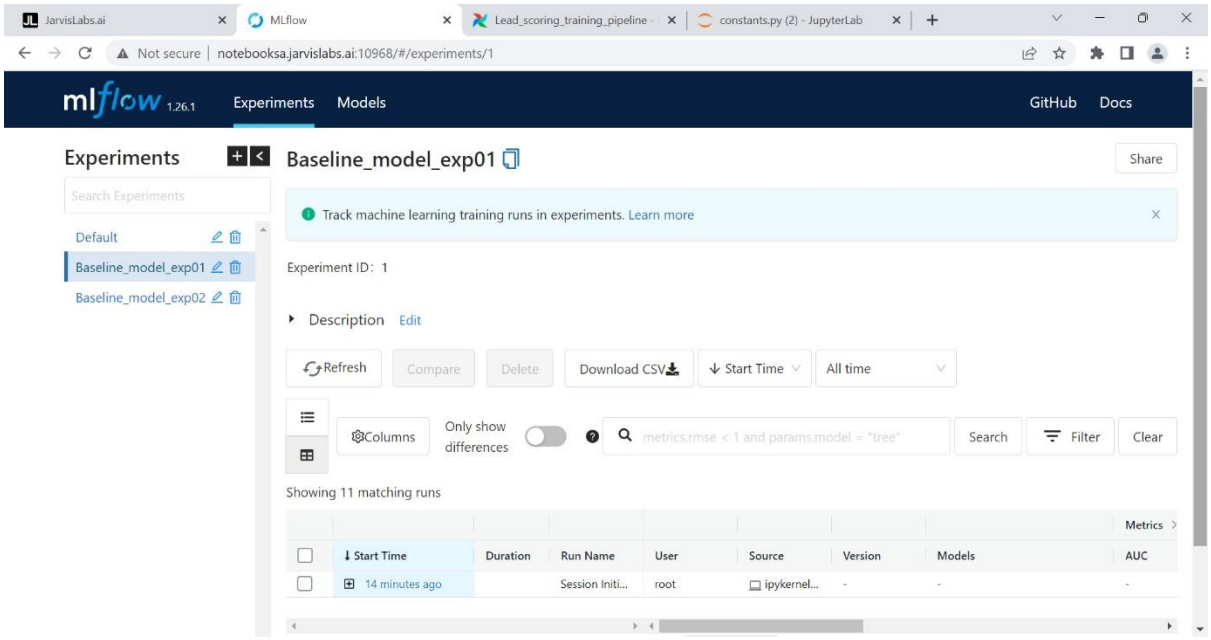

Figure 12: Main MLFlow UI of Baseline_model_exp01

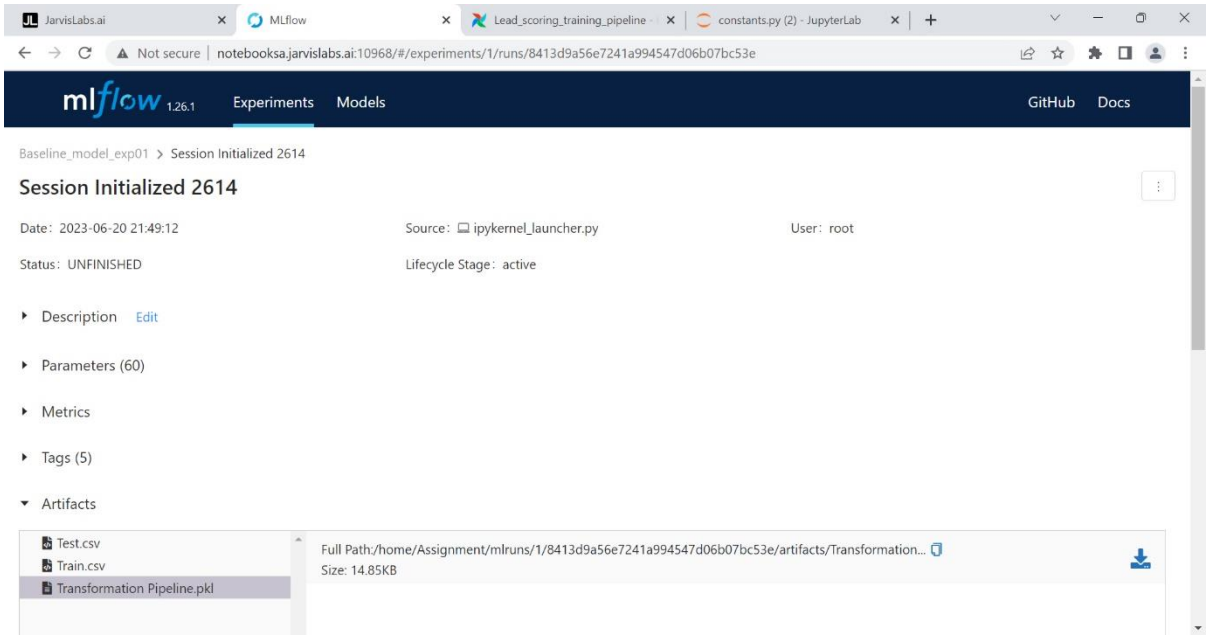

Figure 13: MLFlow Artifacts of Baseline_model_exp01

MLFlow of Baseline_model_exp02

Baseline_model_exp02 is the pycaret setup experiment done for training model **after removing certain features**.

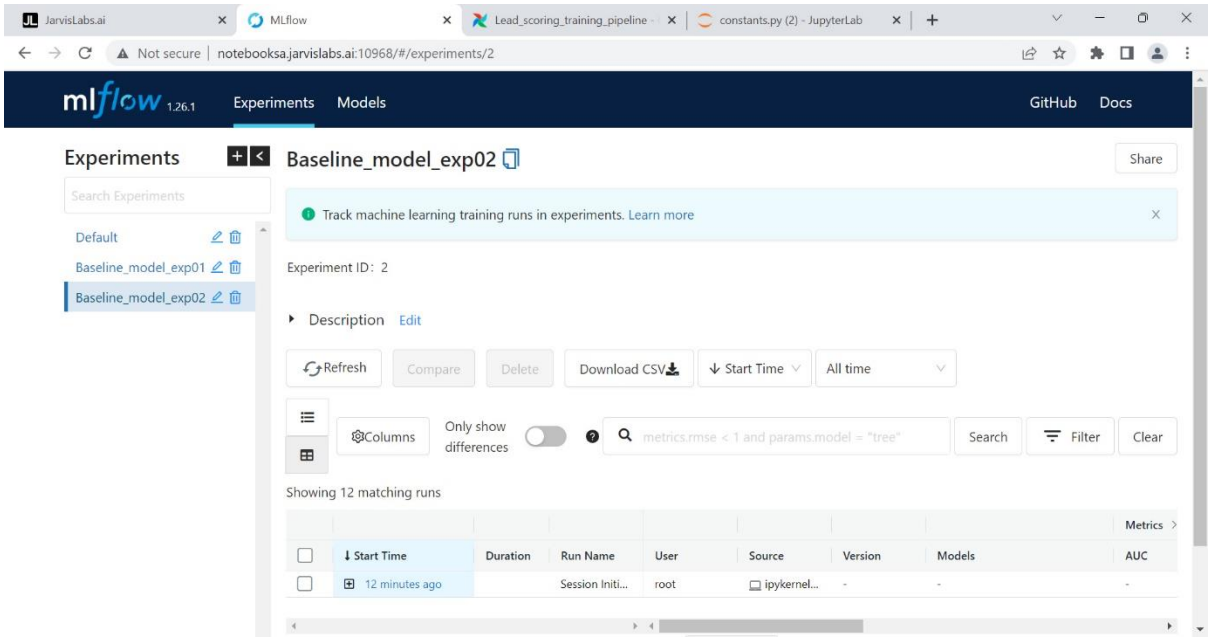

Figure 14: Main MLFlow UI of Baseline_model_exp02

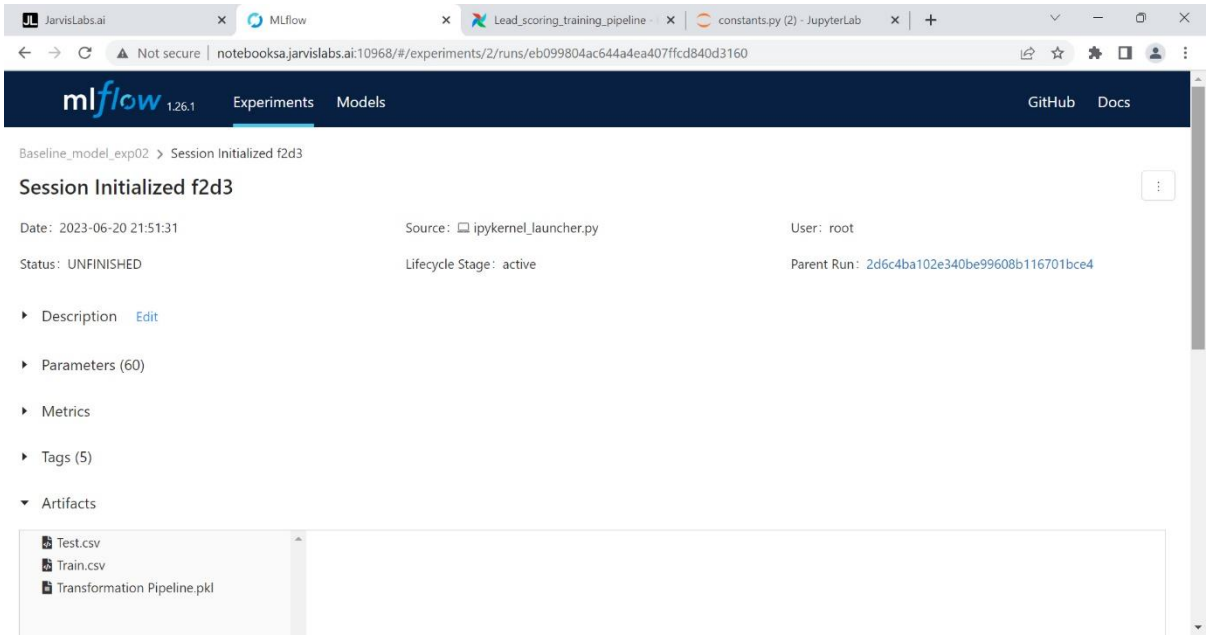

Figure 15: MLFlow Artifacts of Baseline_model_exp02

MLFlow of Lead_scoring_mlflow_production

Lead_scoring_mlflow_production is the final model we have selected. We can see 3 models are generated:

- **Version 1** - One was created when dummy notebook was run to check the working of utils.py functions.
- **Version 2** - One was created during schedule run of airflow (automatic).
- **Version 3** - One was created manually from airflow UI. This is our final model and we have moved this version to **PRODUCTION** stage.

Please check the figure captions to understand the following screenshots.

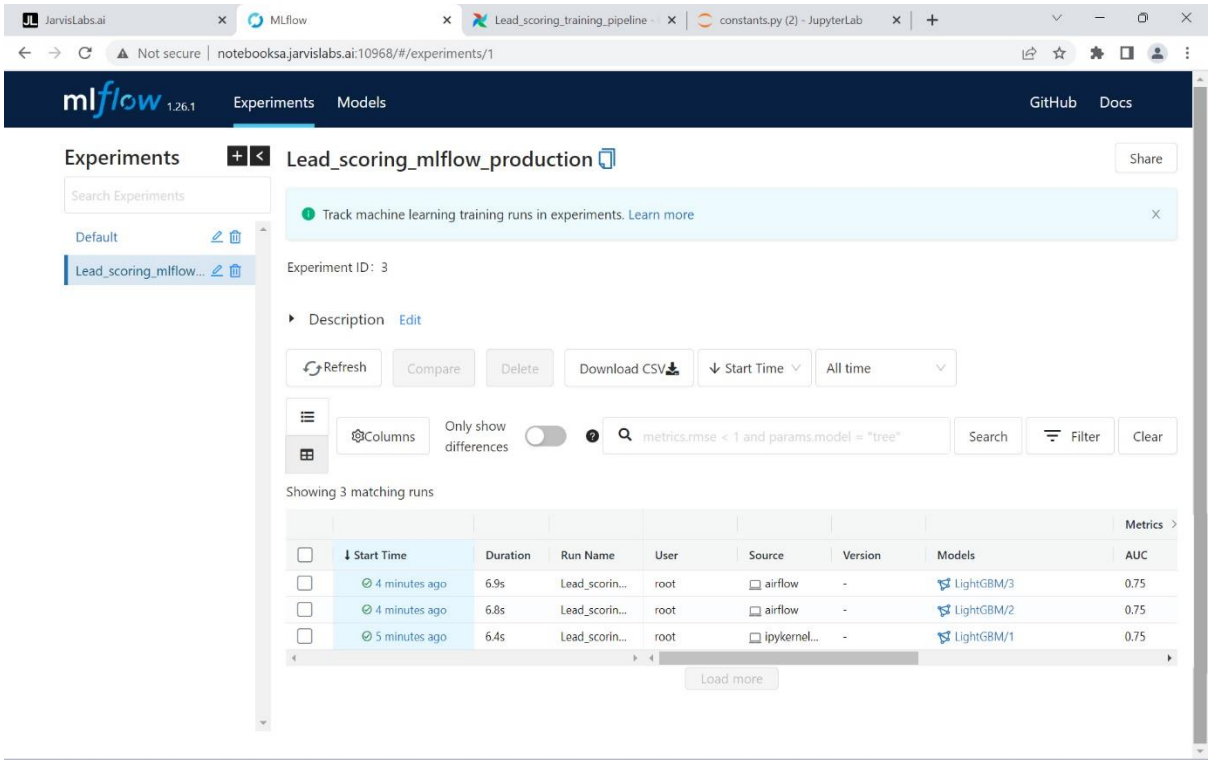

Figure 16: Main MLFlow UI of Lead_scoring_mlflow_production

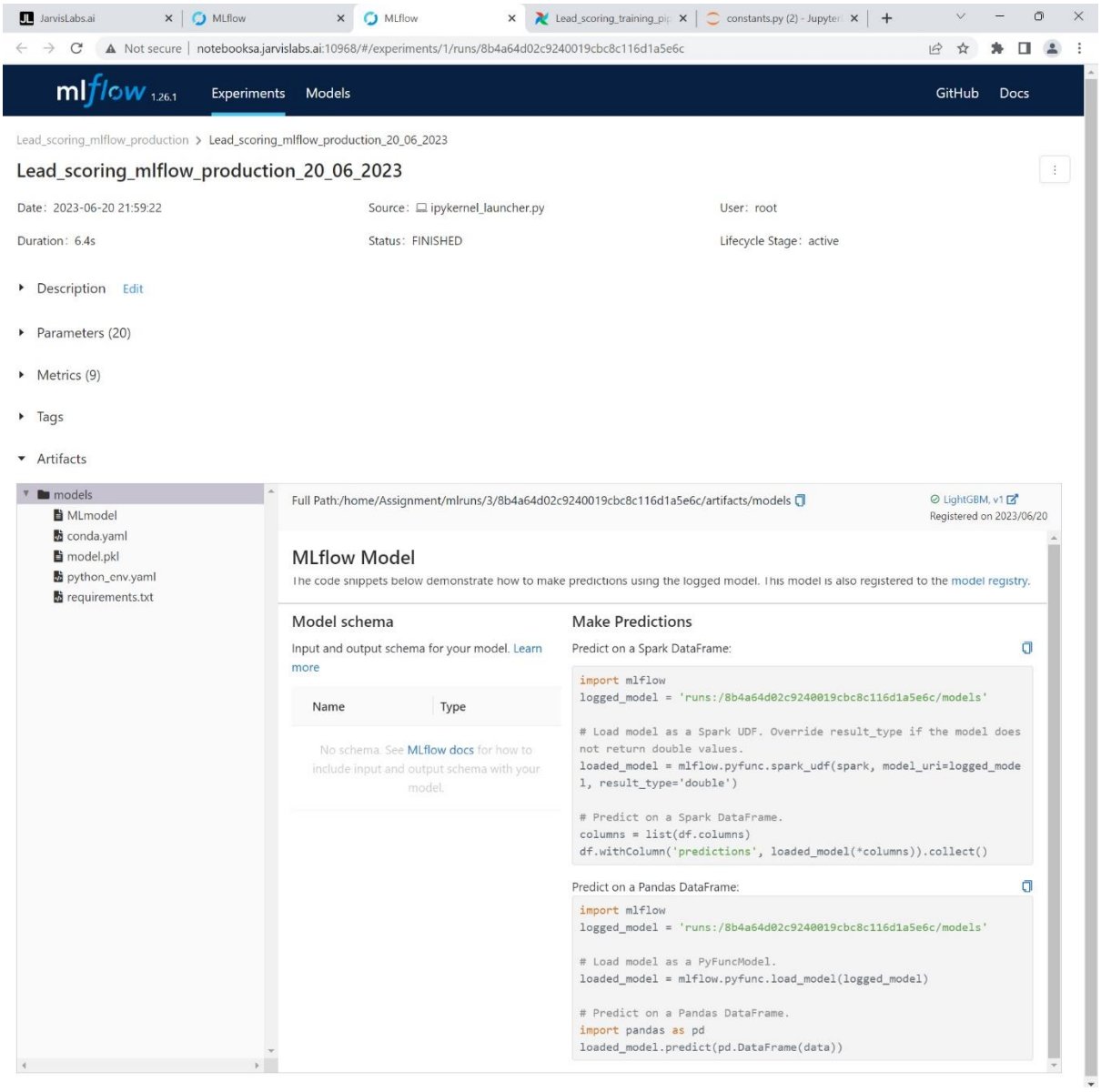

Figure 17: MLFlow Artifacts of Version 1 model

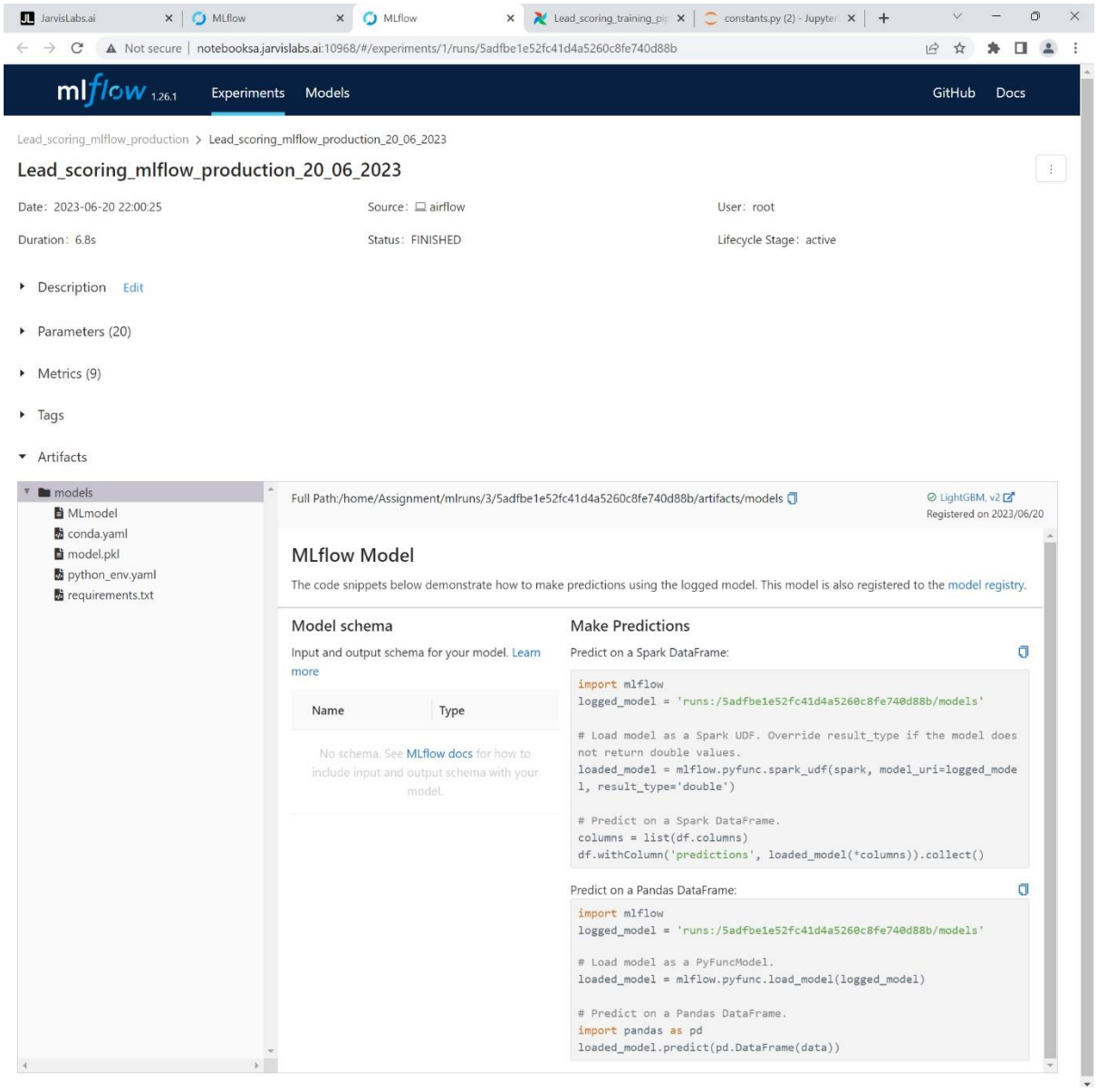

Figure 18: MLFlow Artifacts of Version 2 model

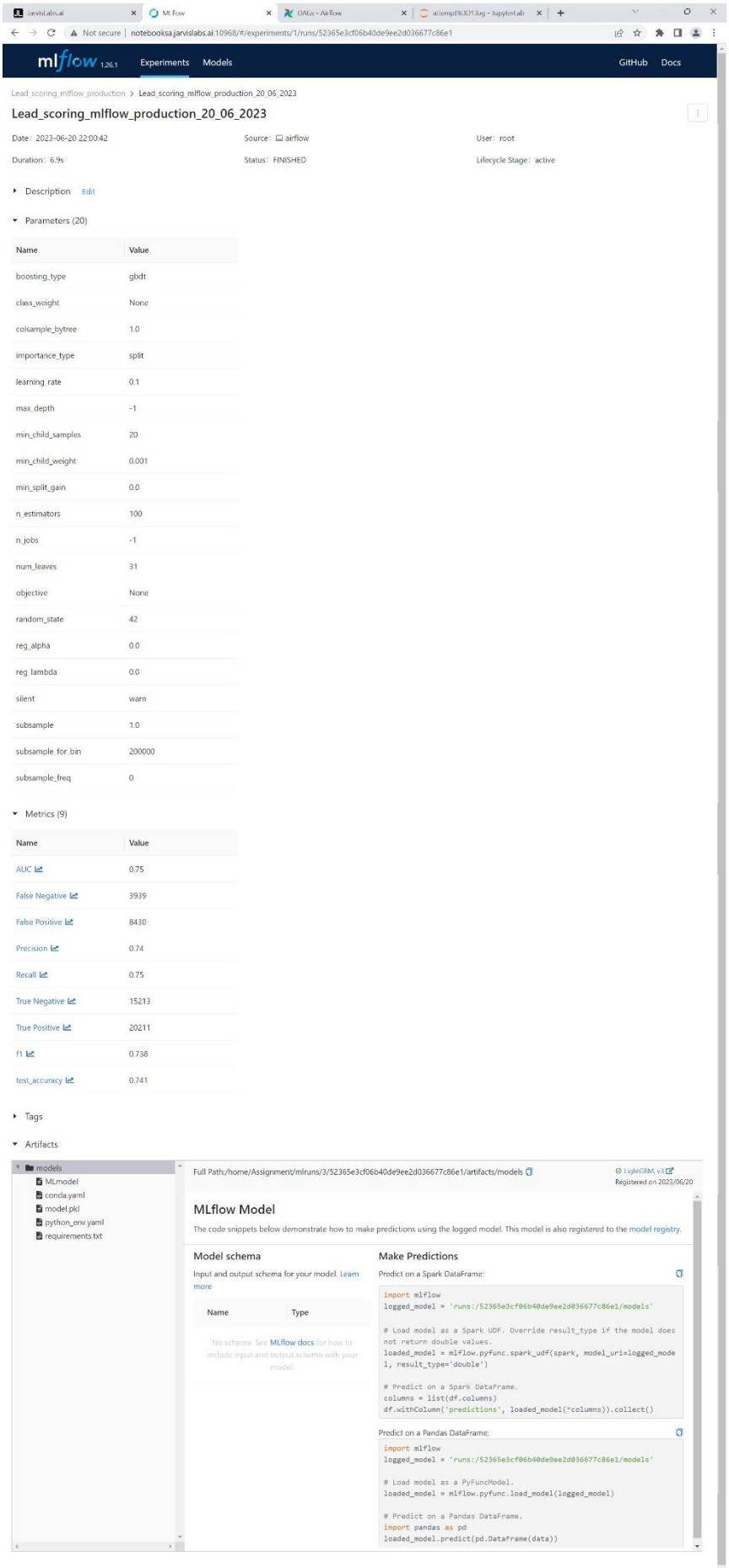
JarvisLabs.ai
MLflow
DAGs - Airflow
Not secure | notebooksa.jarvislabs.ai:10968/#/experiments/1/runs/52365e3cf06b40de9ee2d036677c86e1
mlflow 1.26.1
Experiments
Models
GitHub
Docs
Lead_scoring_mlflow_production > Lead_scoring_mlflow_production_20_06_2023
Lead_scoring_mlflow_production_20_06_2023
Date: 2023-06-20 22:00:42
Source: airflow
User: root
Duration: 6.9s
Status: FINISHED
Lifecycle Stage: active
Description Edit
Parameters (20)
Name Value
boosting_type gbdt
class_weight None
colsample_bytree 1.0
importance_type split
learning_rate 0.1
max_depth -1
min_child_samples 20
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
Metrics (9)
Name Value
AUC 0.75
False Negative 3939
False Positive 8430
Precision 0.74
Recall 0.75
True Negative 15213
True Positive 20211
f1 0.738
test_accuracy 0.741
Tags
Artifacts
models
MLmodel
conda.yaml
model.pkl
python_env.yaml
requirements.txt
Full Path:/home/Assignment/mlruns/3/52365e3cf06b40de9ee2d036677c86e1/artifacts/models
LightGBM, v3
Registered on 2023/06/20
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:/52365e3cf06b40de9ee2d036677c86e1/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:/52365e3cf06b40de9ee2d036677c86e1/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))

Figure 19: MLFlow Artifacts of Version 3 model

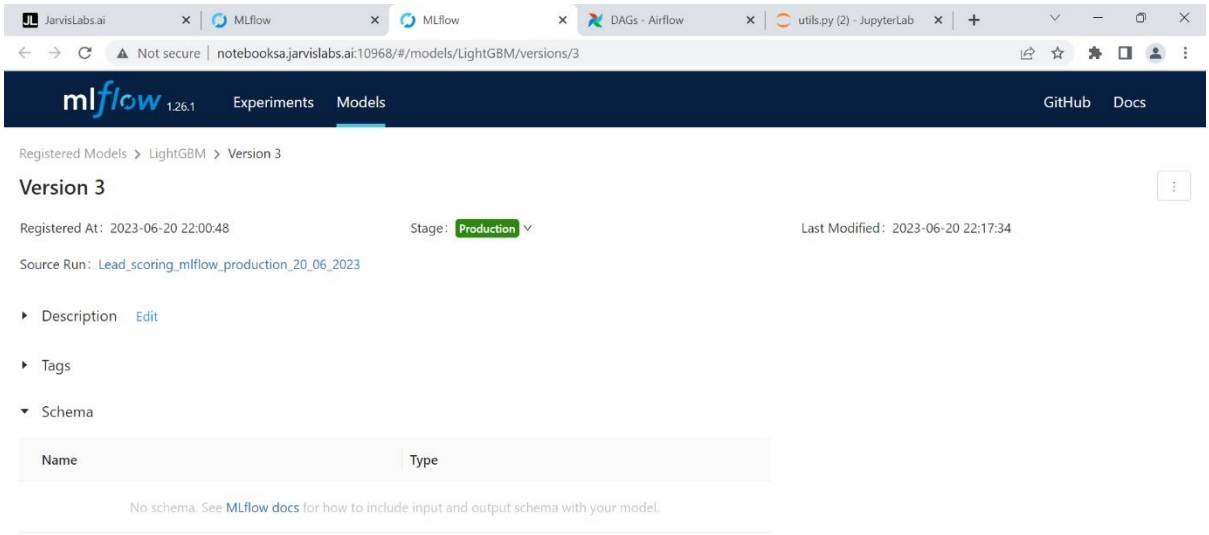

Figure 20: Version 3 model registered as Production Stage

Pytest of Data Pipeline

Screenshot of pytest done for data pipeline.

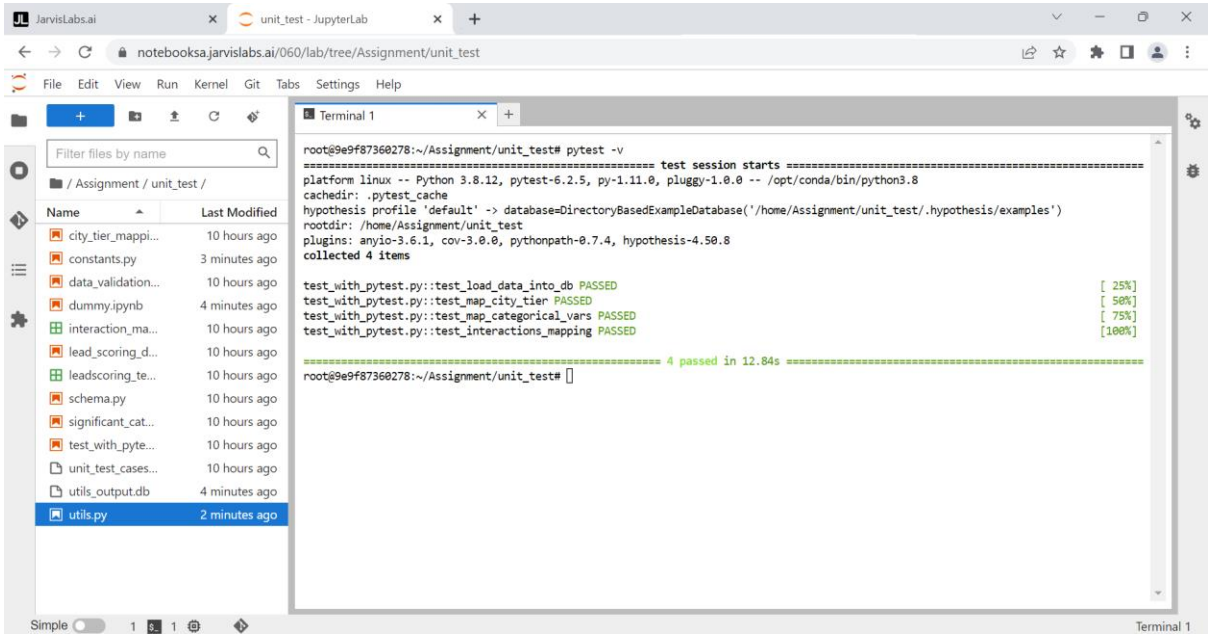

Figure 21: Pytest Test Results