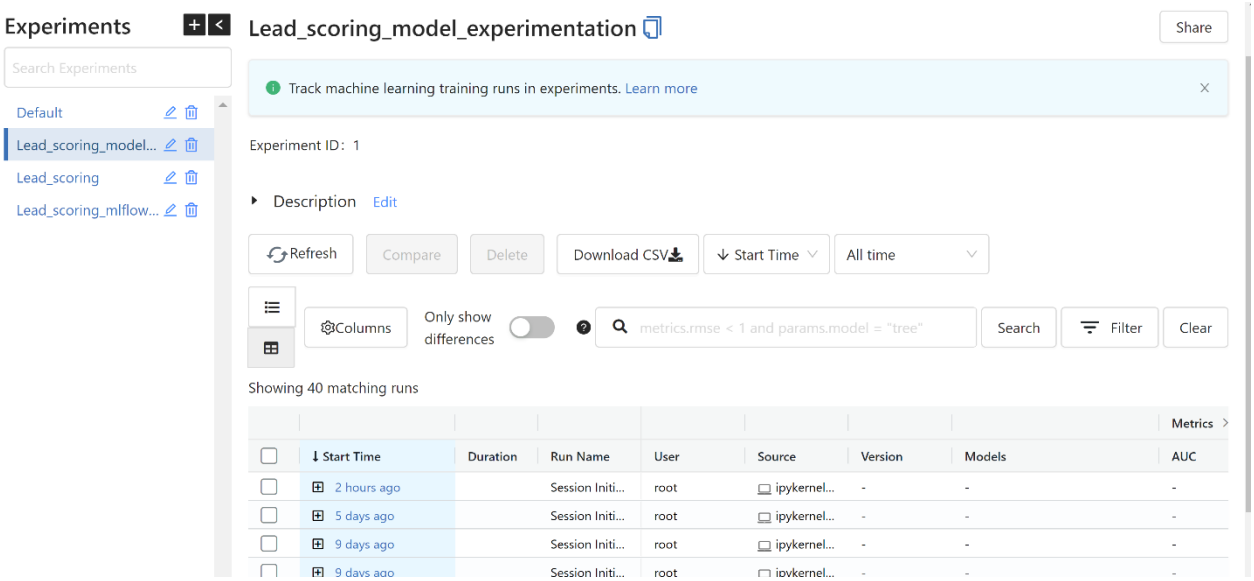
Model Experimentation Part

Screenshot of ml flow during experimentation



Experiment before features dropping

```
# setup pycaret
exp_lead_scoring = setup(data=dataset, target = 'app_complete_flag',
                  fold_shuffle=True,
                  session_id = 42,
                  normalize = True,
                  transformation = True,
                  remove_multicollinearity = True, multicollinearity_threshold = 0.95,
                  n_jobs=-1,use_gpu=True,
                  log_experiment=True,experiment_name='Lead_scoring_model_experimentation',
                  log_plots=True, log_data=True,
                  silent=True, verbose=True,
                  log_profile=False)
```

	Description	Value
0	session_id	42
1	Target	app_complete_flag
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(238964, 12)
5	Missing Values	False

Experiment setup after features dropping

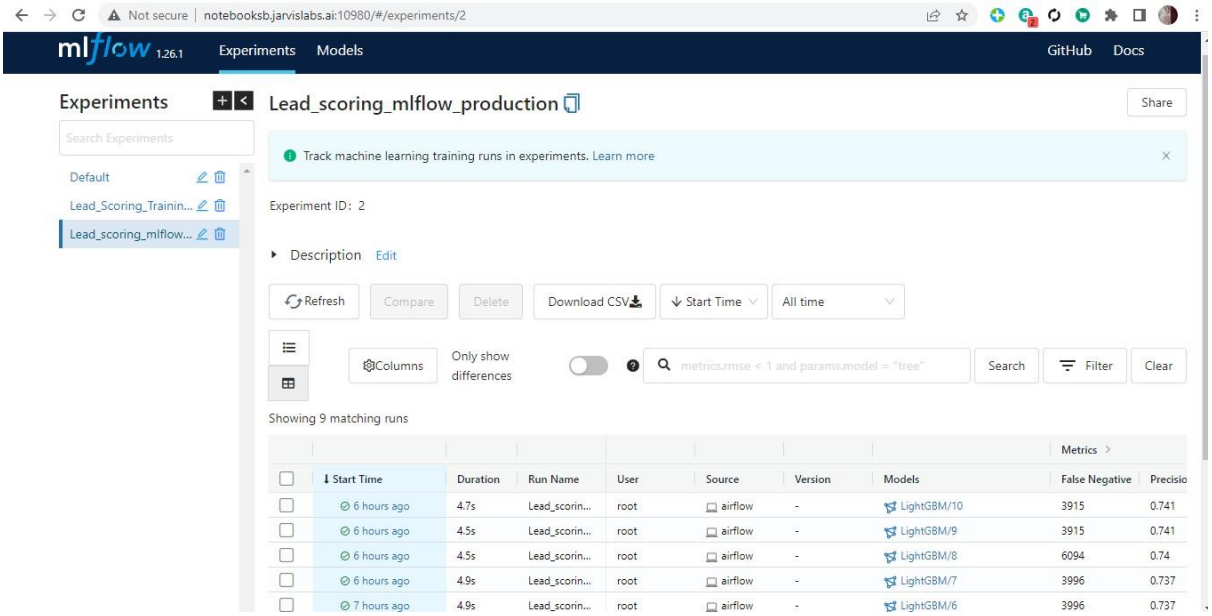
[14]:
```
# setup pycaret
from pycaret.classification import *

exp_lead_scoring = setup(data=dataset_cols, target = 'app_complete_flag',
                    remove_multicollinearity = True, multicollinearity_threshold = 0.95,
                    categorical_features = ['city_tier', 'first_platform_c','first_utm_medium_c', 'first_utm_
                    fold_shuffle=True,
                    session_id = 42,
                    n_jobs=-1,use_gpu=True,
                    log_experiment=True,experiment_name='Lead_scoring',
                    log_plots=True, log_data=True,
                    silent=True, verbose=True,
                    log_profile=False)
```

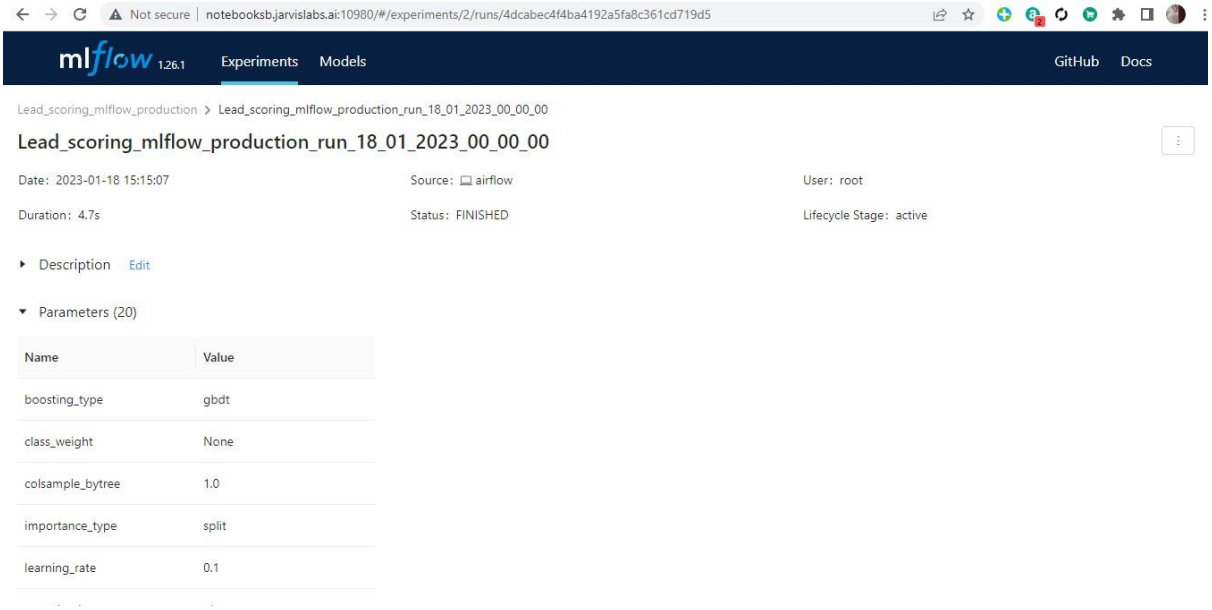
	Description	Value
0	session_id	42
1	Target	app_complete_flag
2	Target Type	Binary
3	Label Encoded	None
4	Original Data	(238964, 7)
5	Missing Values	False
6	Numeric Features	1

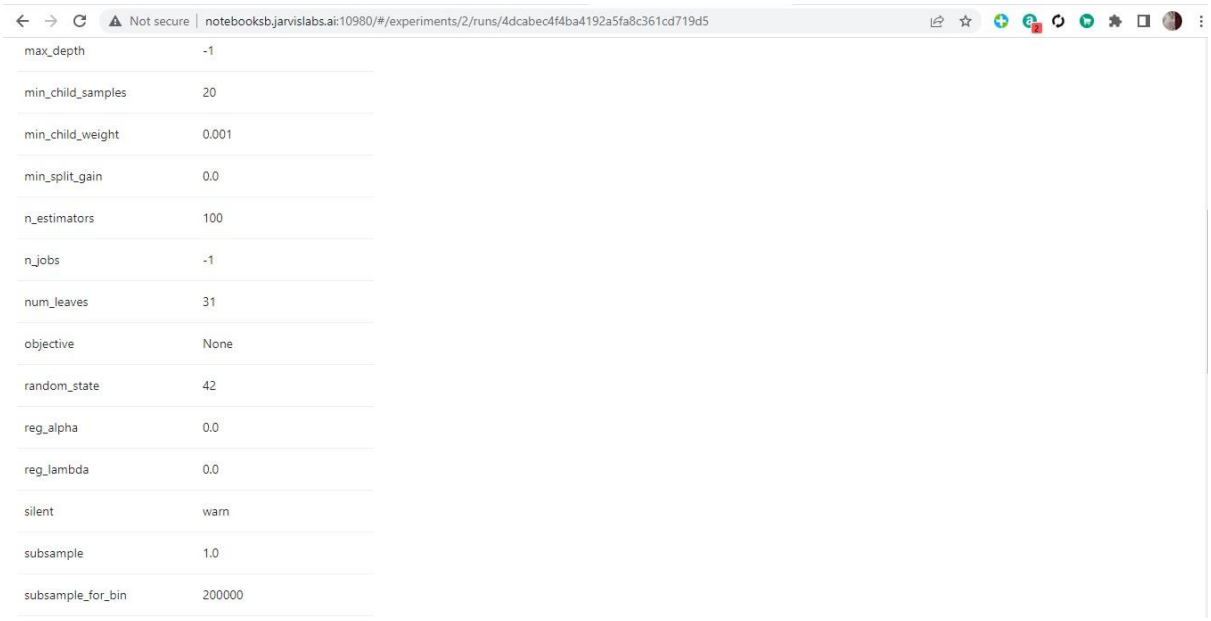
Model Training Part

After Running multiple times mlflow_production

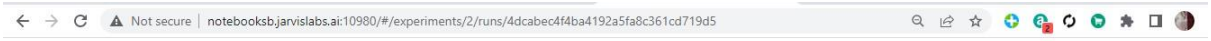


Below some screenshot show details of LightGBM/10 model from above mlflow production



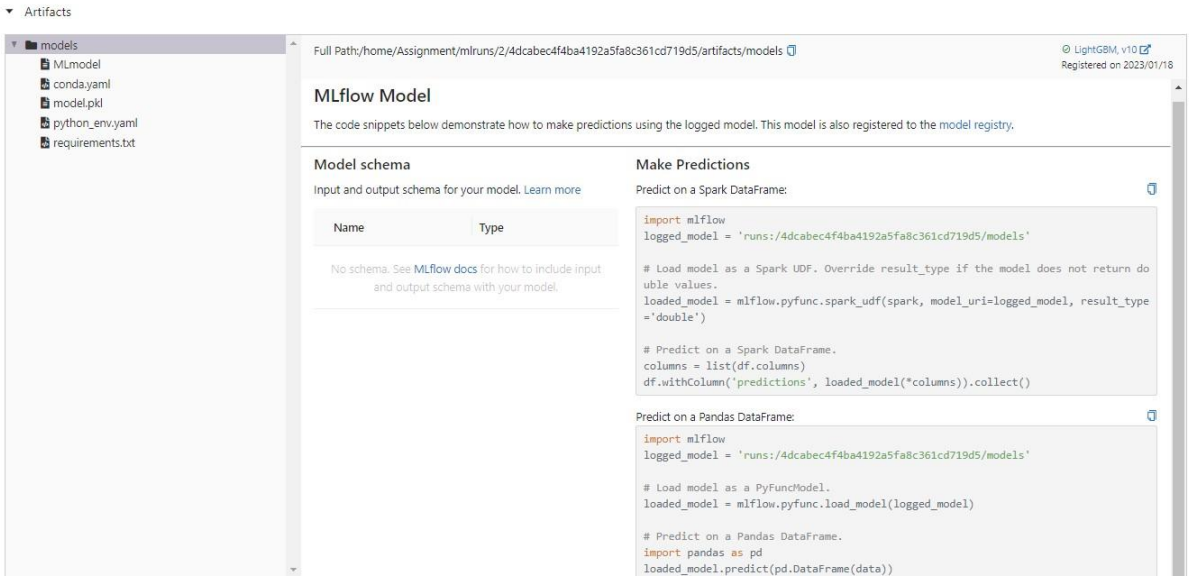


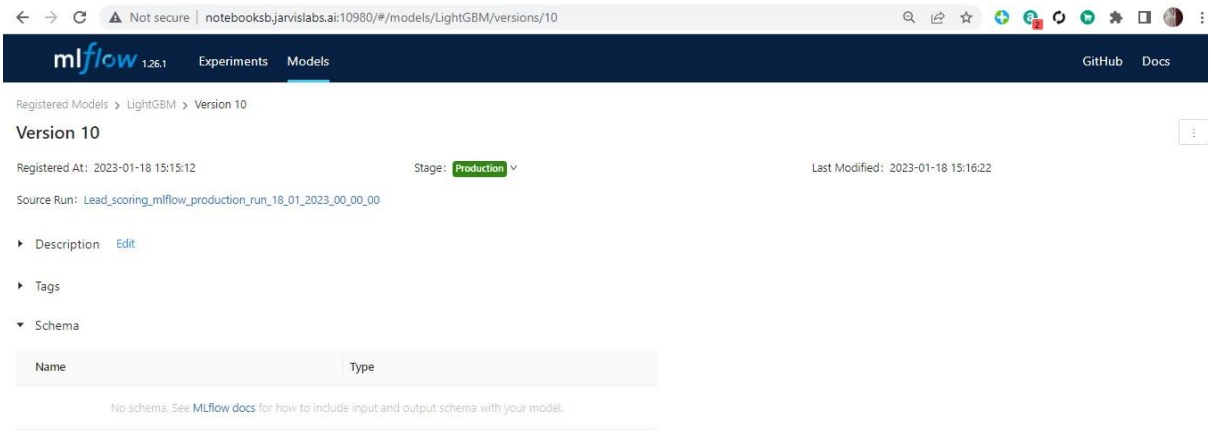
max_depth	-1
min_child_samples	20
min_child_weight	0.001
min_split_gain	0.0
n_estimators	100
n_jobs	-1
num_leaves	31
objective	None
random_state	42
reg_alpha	0.0
reg_lambda	0.0
silent	warn
subsample	1.0
subsample_for_bin	200000



▼ Metrics (13)

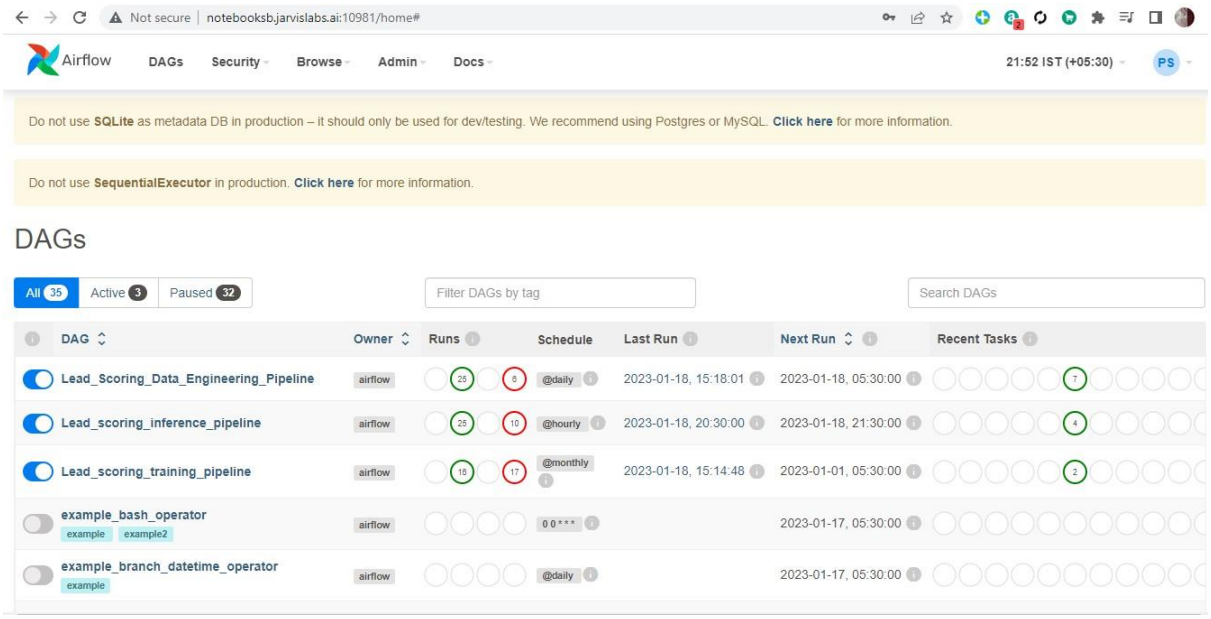
Name	Value
False Negative	3915
Precision	0.741
Precision_0	0.795
Precision_1	0.706
Recall	0.751
Recall_0	0.643
Recall_1	0.838
True Negative	15199
f1 score	0.739
f1_0	0.711
f1_1	0.766
roc_auc	0.751
test_accuracy	0.742


Artifacts
models
MLmodel
conda.yaml
model.pkl
python_env.yaml
requirements.txt
Full Path:/home/Assignment/mlruns/2/4dcabec4f4ba4192a5fa8c361cd719d5/artifacts/models
LightGBM, v10
Registered on 2023/01/18
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:/4dcabec4f4ba4192a5fa8c361cd719d5/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:/4dcabec4f4ba4192a5fa8c361cd719d5/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))

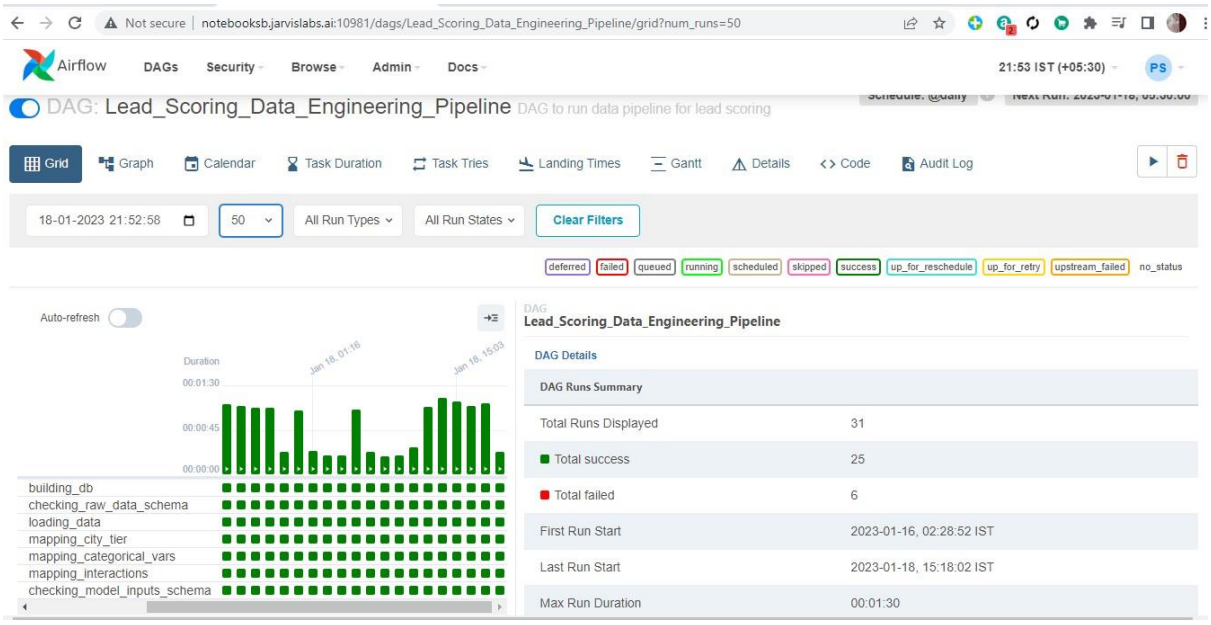

Not secure | notebooksb.jarvislabs.ai:10980/#/models/LightGBM/versions/10
mlflow 1.26.1
Experiments
Models
GitHub
Docs
Registered Models > LightGBM > Version 10
Version 10
Registered At: 2023-01-18 15:15:12
Stage: Production
Last Modified: 2023-01-18 15:16:22
Source Run: Lead_scoring_mlflow_production_run_18_01_2023_00_00_00
Description
Edit
Tags
Schema
Name
Type
No schema. See MLflow docs for how to include input and output schema with your model.

Airflow Part Screenshot

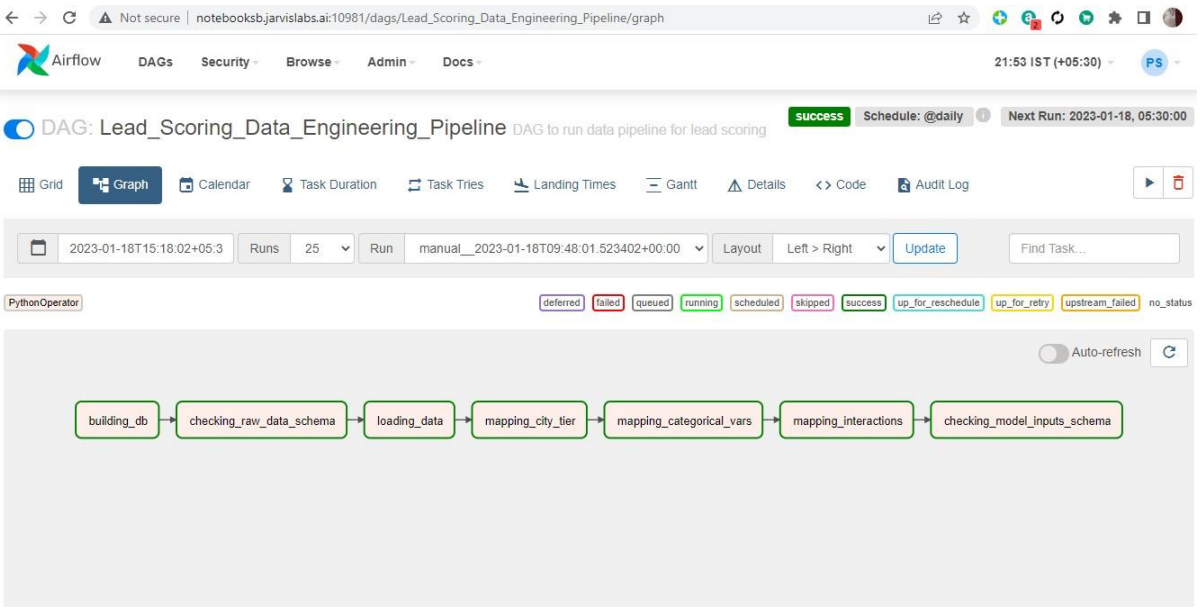
Airflow Home page



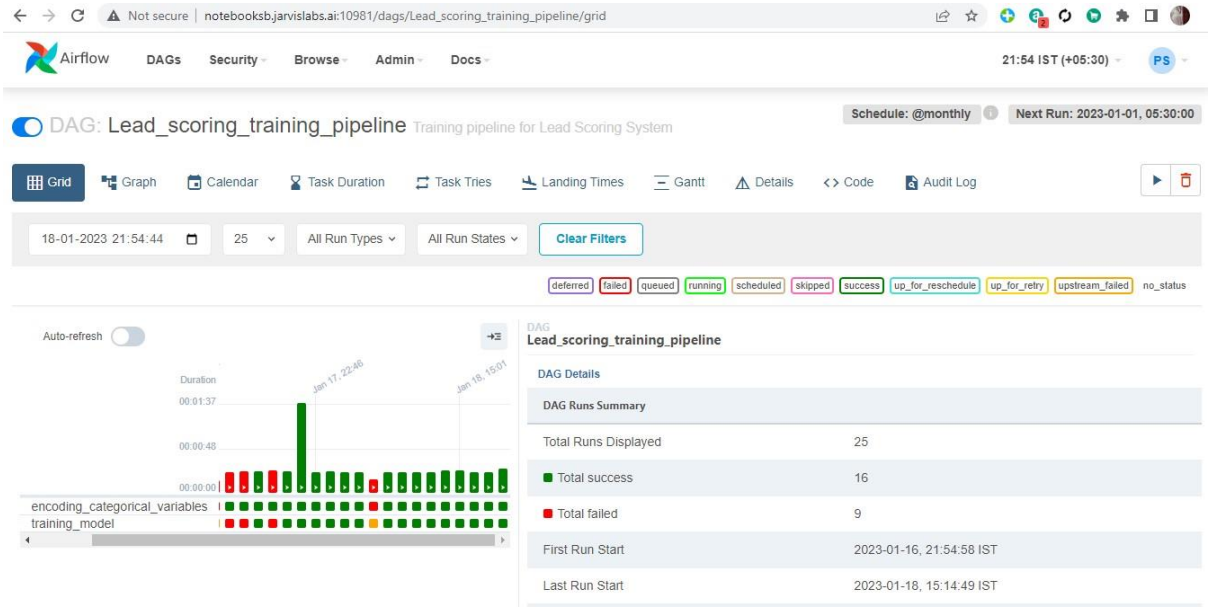
Data Engineering pipeline Grid view



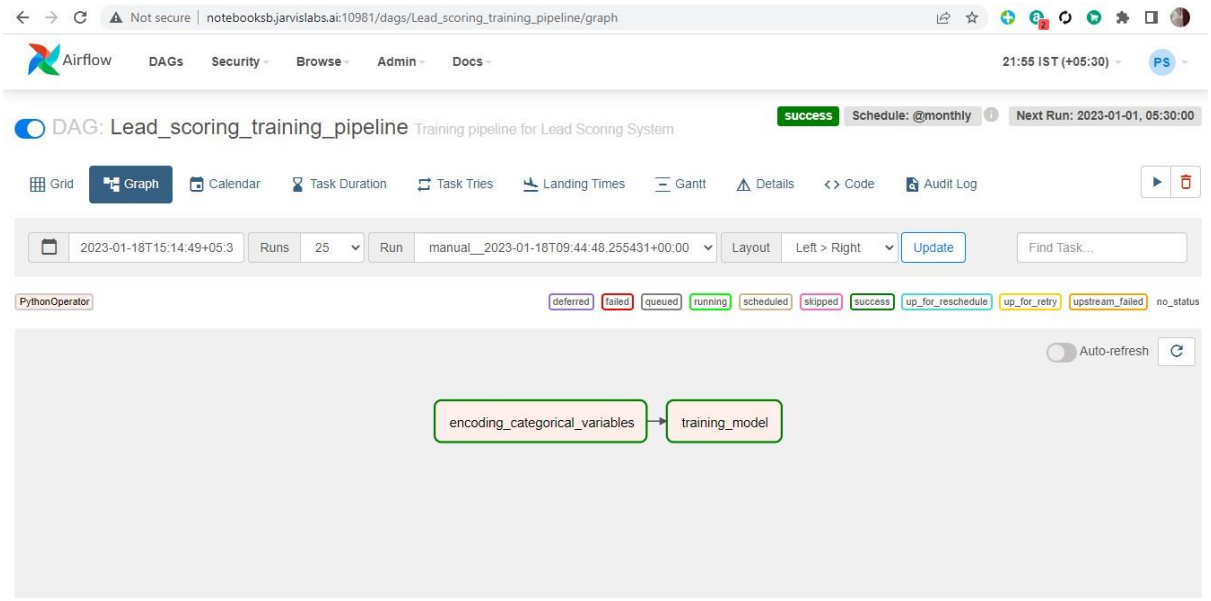
Data Engineering pipeline Graph view



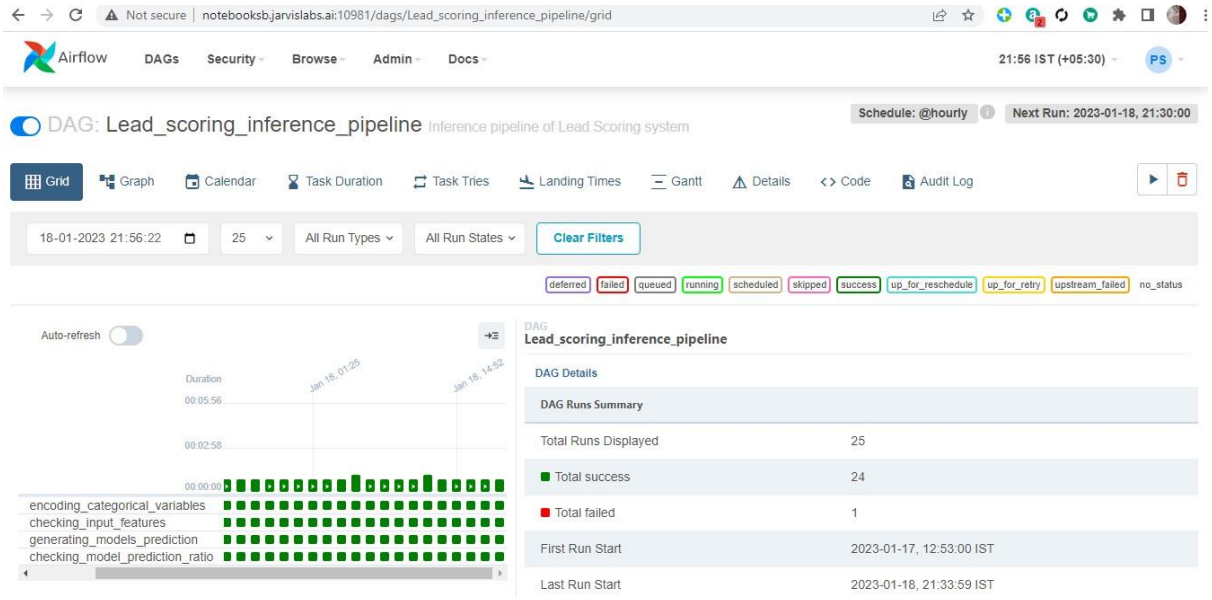
Model Training pipeline Grid view



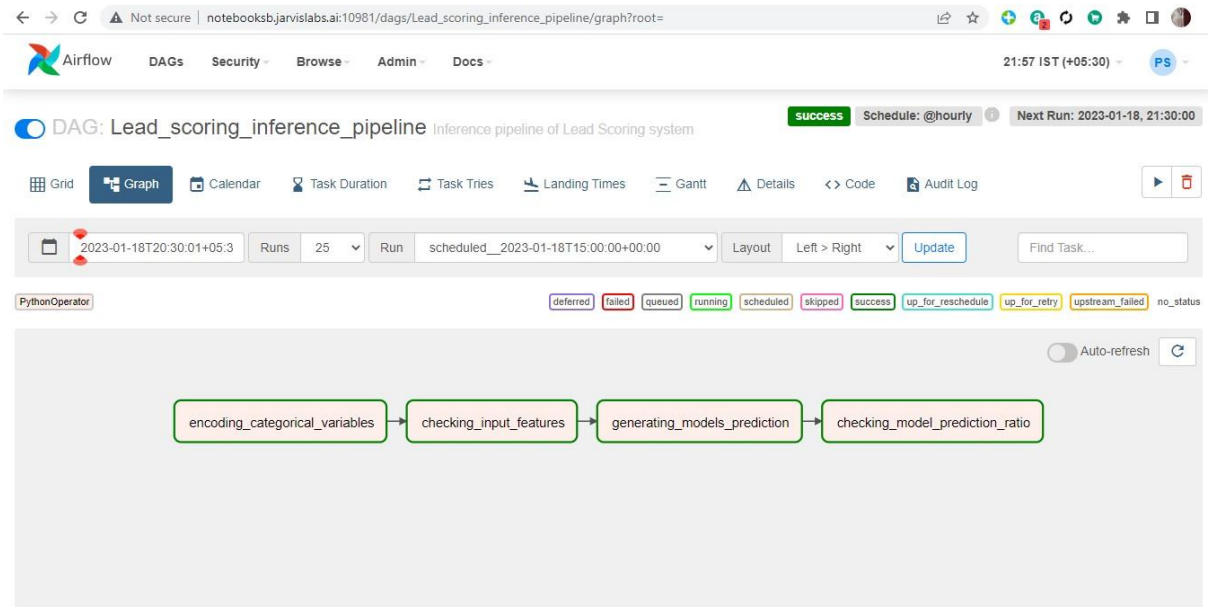
Model Training pipeline Graph view



Inference pipeline Grid view

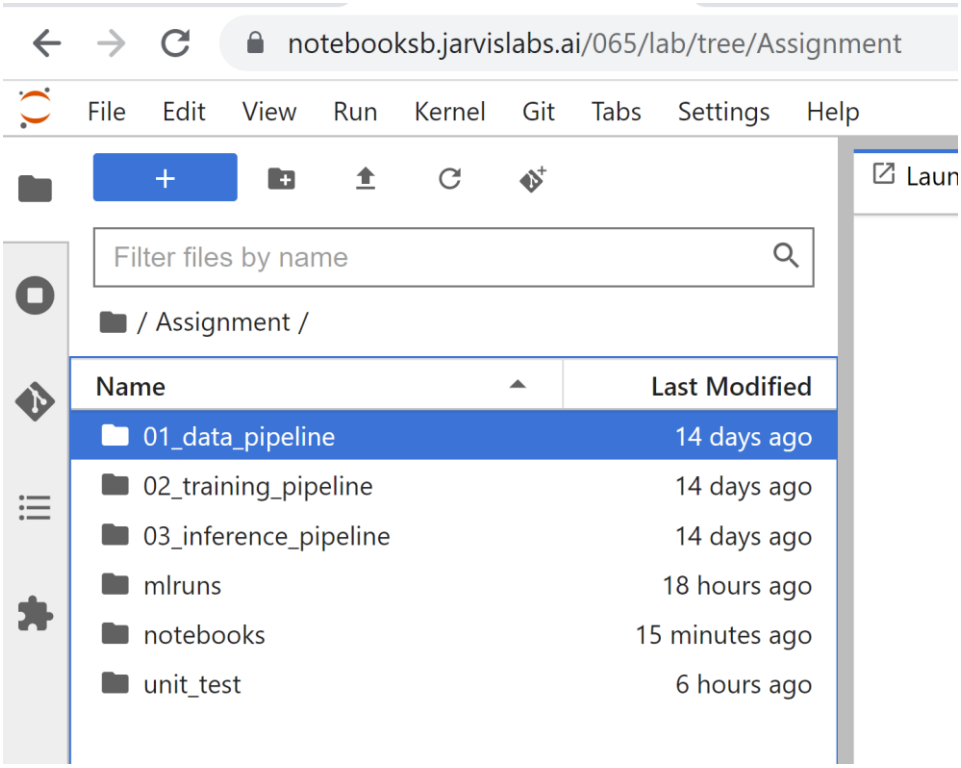


Inference pipeline Graph view

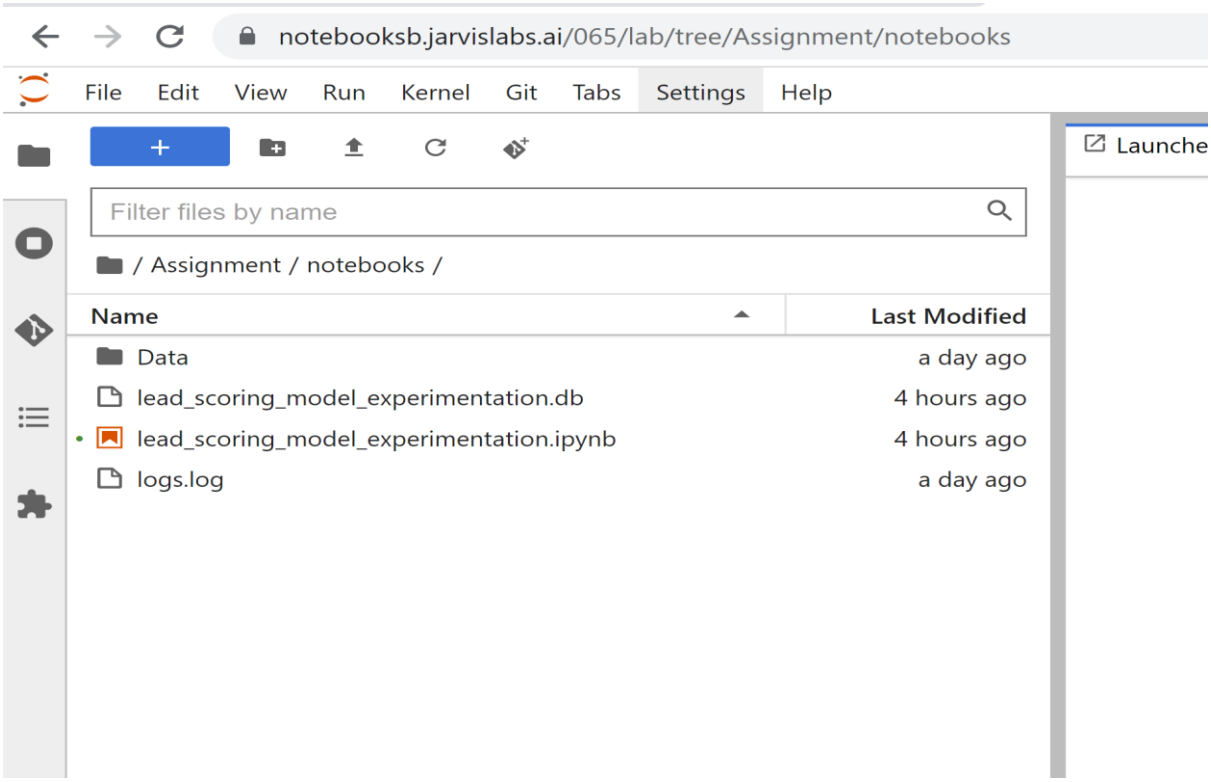


Jarvislab Notebook Location Screenshot

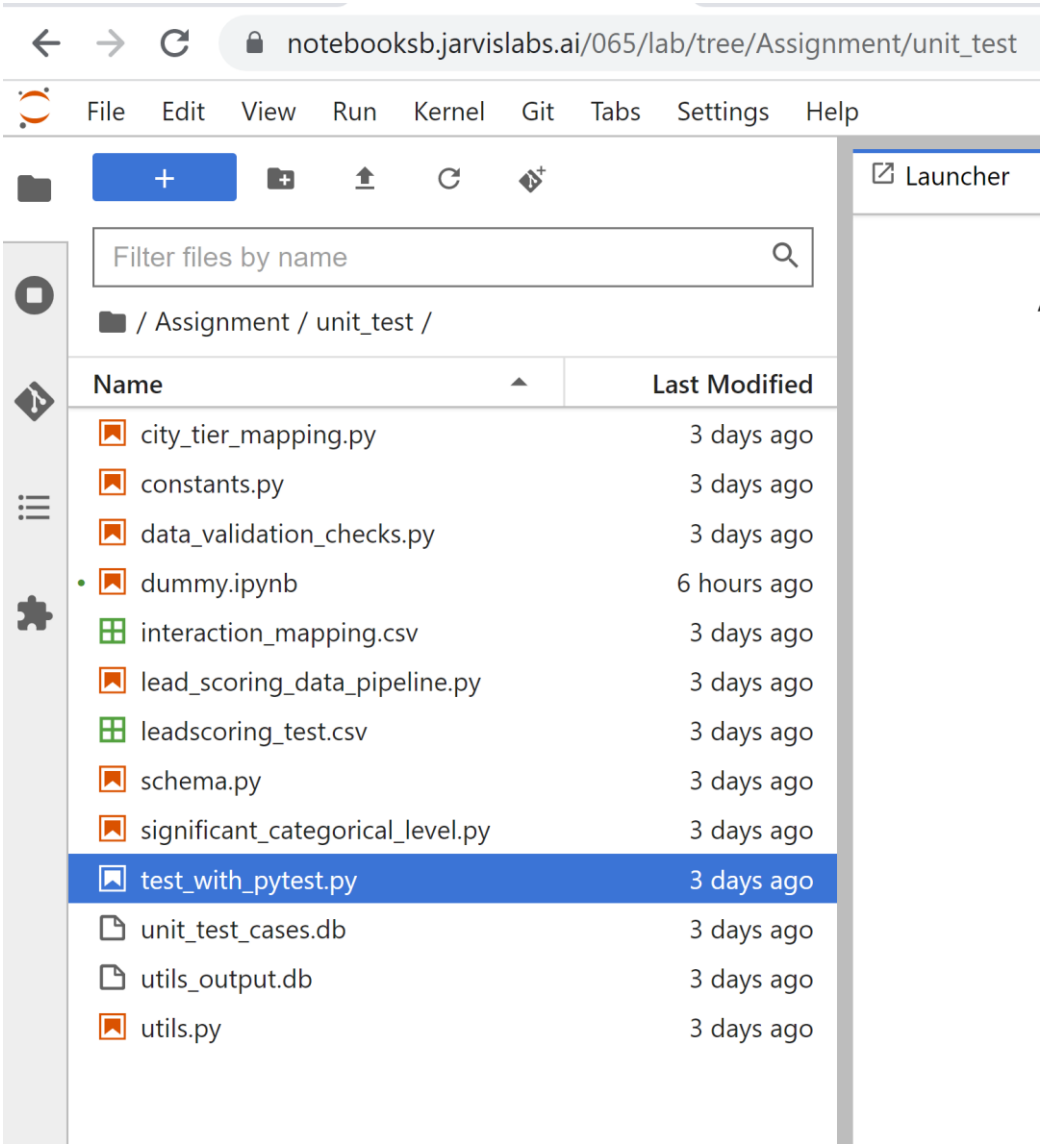
Assignment directory



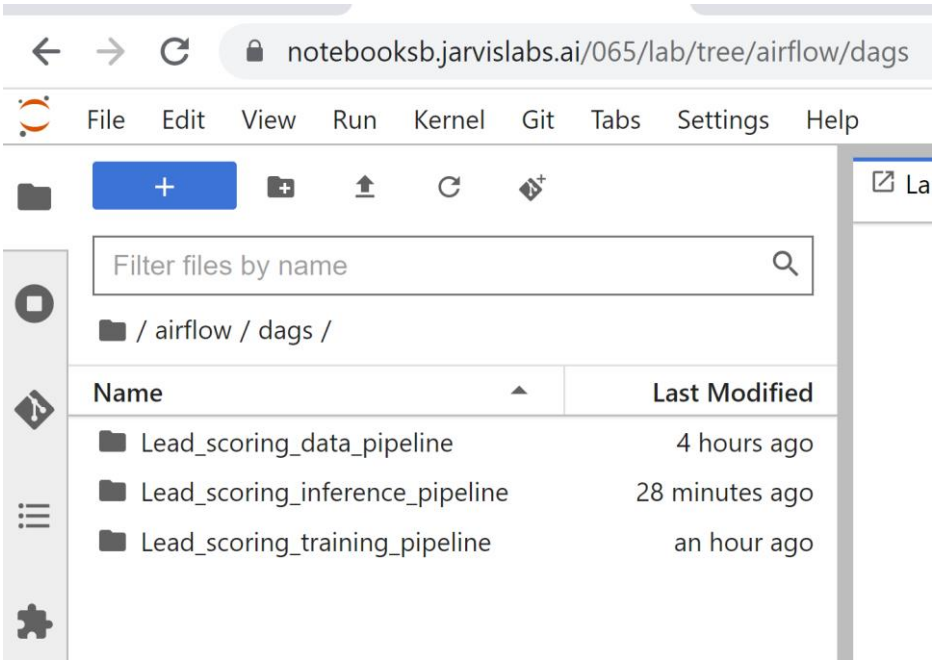
Notebooks directory

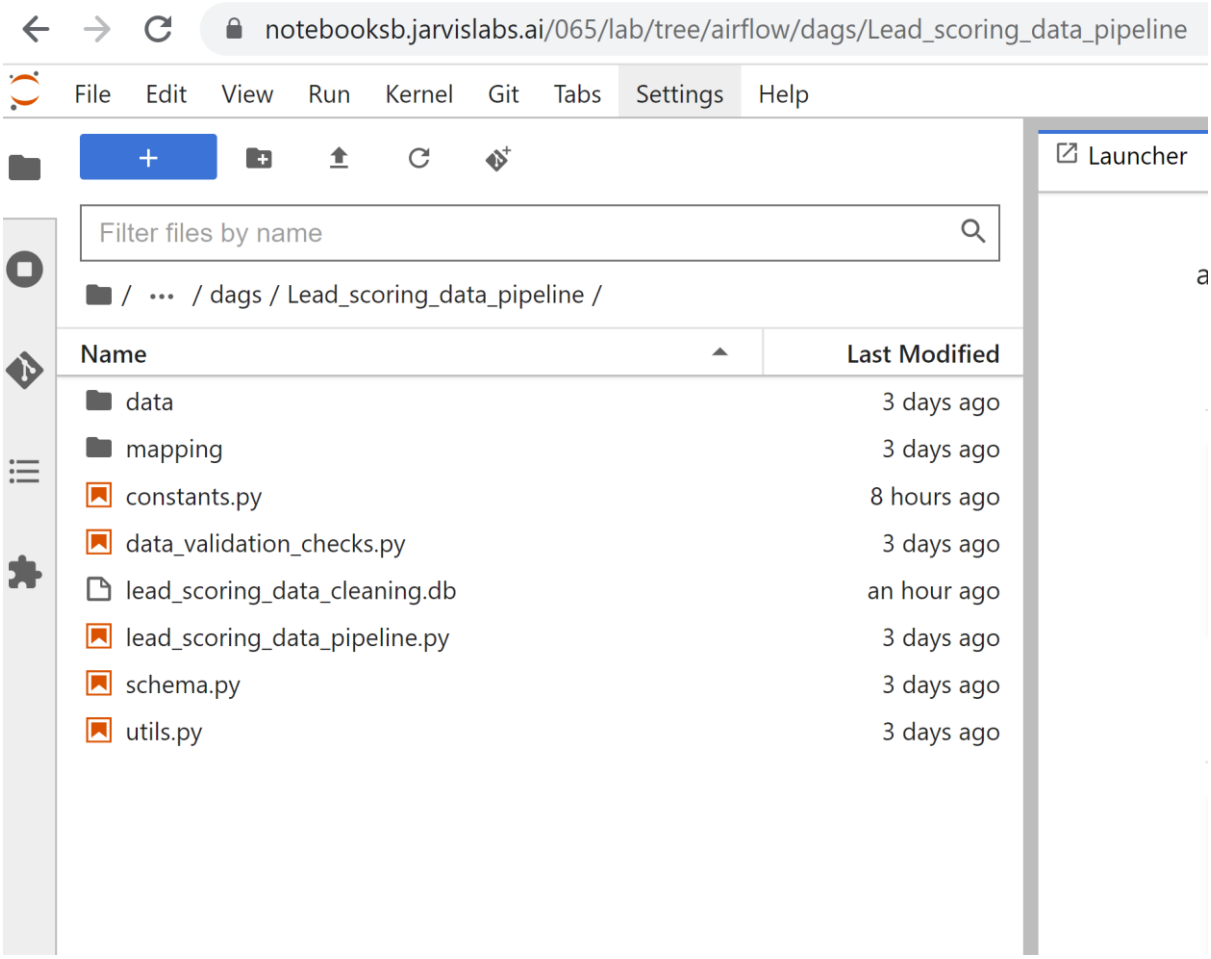


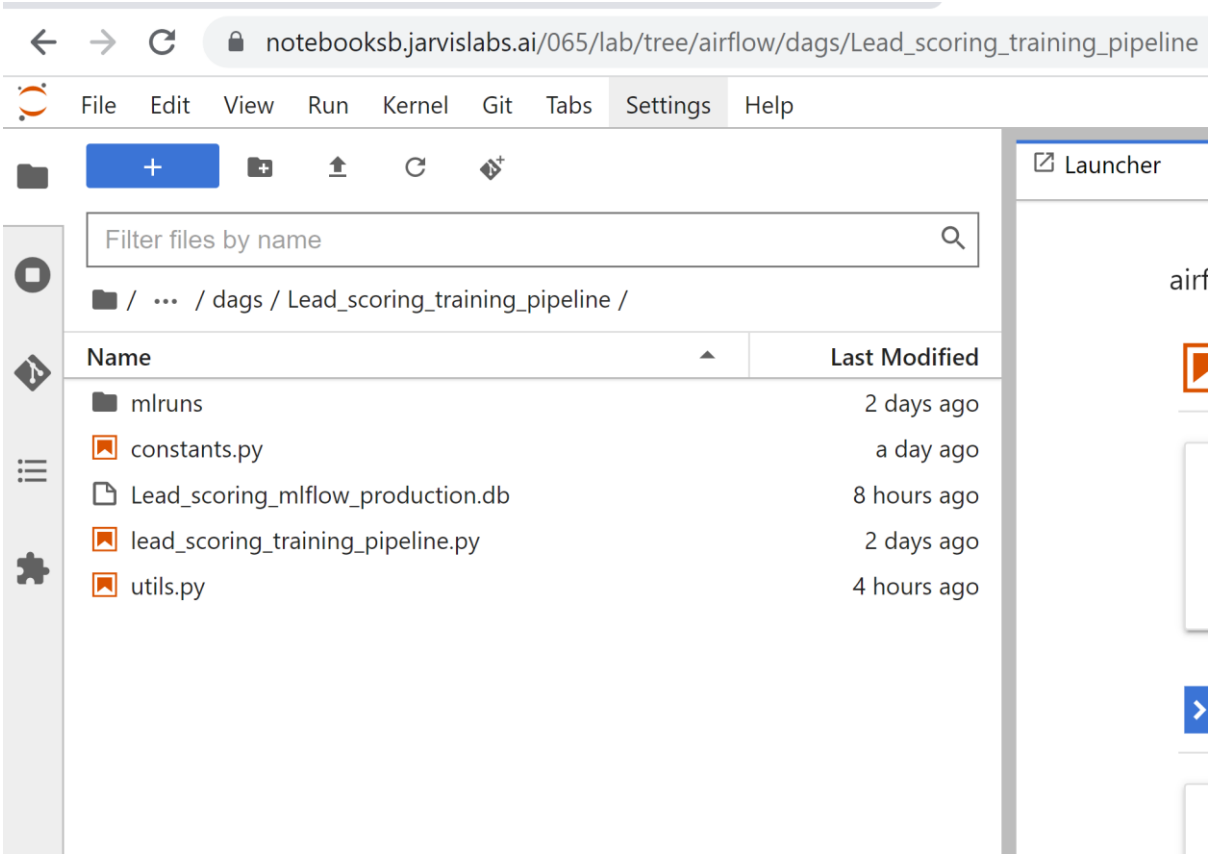
Unit_test directory

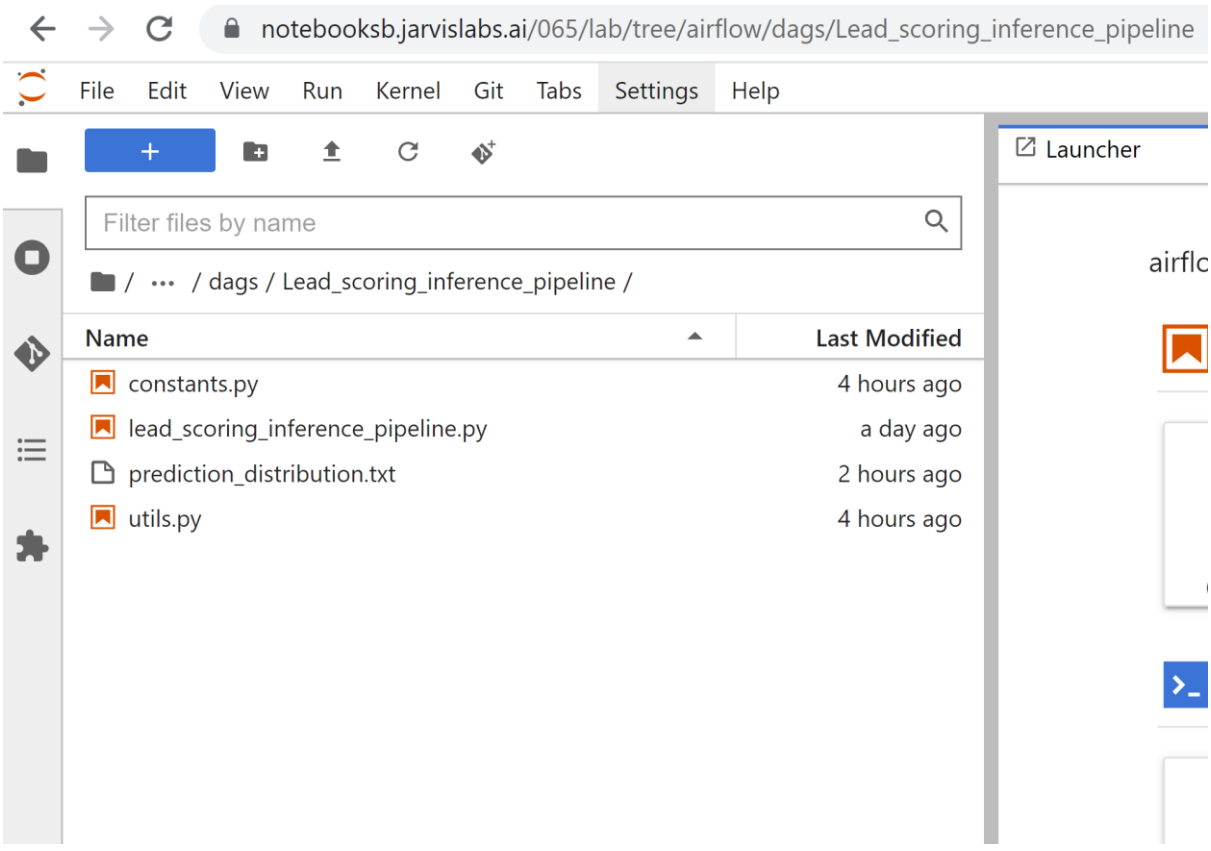


Airflow/dags directory






notebooksb.jarvislabs.ai/065/lab/tree/airflow/dags/Lead_scoring_training_pipeline
File Edit View Run Kernel Git Tabs Settings Help
Launcher
Filter files by name
/ ... / dags / Lead_scoring_training_pipeline /
Name
Last Modified
mlruns 2 days ago
constants.py a day ago
Lead_scoring_mlflow_production.db 8 hours ago
lead_scoring_training_pipeline.py 2 days ago
utils.py 4 hours ago


notebooksb.jarvislabs.ai/065/lab/tree/airflow/dags/Lead_scoring_inference_pipeline
File Edit View Run Kernel Git Tabs Settings Help
Launcher
Filter files by name
/ ... / dags / Lead_scoring_inference_pipeline /
Name
Last Modified
constants.py 4 hours ago
lead_scoring_inference_pipeline.py a day ago
prediction_distribution.txt 2 hours ago
utils.py 4 hours ago