

1. Screenshot of all the experiments

The screenshot shows the mlflow UI interface. At the top, there is a navigation bar with the mlflow logo, version 1.26.1, and links for Experiments and Models. On the right side of the header are GitHub and Docs links. Below the header, the main content area is titled "Experiments" and shows a list of experiments under the heading "Lead_scoring_model_experimentation". A search bar labeled "Search Experiments" is present. Below the search bar, there are two experiment entries: "Default" and "Lead_scoring_model...". The "Lead_scoring_model..." entry is selected, indicated by a blue border. Below the experiment list, there is a section titled "Description" with "Edit" and "Compare" buttons, and a "Delete" button. There are also buttons for "Download CSV" and "Start Time" (set to "All time"). A search bar and filter buttons are also present. The main table displays 22 matching runs, with columns for Start Time, Duration, Run Name, User, Source, Version, Models, AUC, Accuracy, F1, C, and CPU Jobs. The table is sorted by Start Time.

2. Screenshot of one experiments with all the artifacts visible

The screenshot shows a detailed view of an experiment named "Light Gradient Boosting Machine" under the "Lead_Scoring_Model_Experimentation" experiment. At the top, there is a navigation bar with the mlflow logo, version 1.26.1, and links for Experiments and Models. On the right side of the header are GitHub and Docs links. The main content area shows the experiment details: Status: UNFINISHED, Source: ipykernel\launched.py, User: root, Lifecycle Stage: active, and Parent Run: a7db22659cd548a1af6cf26feaf94194. Below the details, there are sections for Description (with Edit link), Parameters (20), Metrics (8), Tags (5), and Artifacts. The Artifacts section is expanded, showing a file tree with items: model (MLmodel, conda.yaml, model.pkl, python_env.yaml, requirements.txt), AUC.png, Confusion Matrix.png, Feature Importance.png, and Holdout.html. The AUC.png file is selected and its content is displayed as a graph showing the ROC curve with an AUC of 0.821. The graph has an x-axis and y-axis ranging from 0.0 to 1.0.

mlflow 1.26.1 Experiments Models

Lead_Scoring_Model_Experimentation > Light Gradient Boosting Machine

Light Gradient Boosting Machine

Status: UNFINISHED Source: ipykernel_launcher.py User: root

Lifecycle Stage: active Parent Run: a7db22659cd548a1af6cf26feaf94194

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

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	Make Predictions
Input and output schema for your model. Learn more	Predict on a Spark DataFrame:
Name Type	<pre>import mlflow logged_model = 'runs:/3ec9233365d043ada37e648fd6b-d6752/model'</pre>
No schema. See MLflow docs for how to include input and output schema with your model.	<pre># 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')</pre>
	<pre># Predict on a Spark DataFrame. columns = list(df.columns) df.withColumn('predictions', loaded_model(*columns)).collect()</pre>
	Predict on a Pandas DataFrame:
	<pre>df = spark.read.parquet('.../data.parquet')</pre>

After Dropping Features

1. Screenshot of all the experiments

mlflow 1.26.1 Experiments Models

Experiments + Lead_Scoring_Model_Experimentation_After_Dropping_Features

Search Experiments

Default

Lead_scoring_model... Lead_Scoring_Model_Experimentation_After_Dropping_Features

Experiment ID: 3

Description Edit

Refresh Compare Delete Download CSV Start Time All time

Only show differences Metrics > Parameters >

Showing 12 matching runs

Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy	F1	C	CPU Jobs
26 minutes ago		Session Init...	root	ipykernel...	-	sklearn	0.821	0.739	0.76	-	-1
19 minutes ago		Light Grade...	root	ipykernel...	-	sklearn	0.821	0.739	0.762	-	-
22 minutes ago		Light Grade...	root	ipykernel...	-	sklearn	0.734	0.673	0.725	-	-
22 minutes ago		Naive Bayes	root	ipykernel...	-	sklearn	0.773	0.7	0.727	-	-
22 minutes ago		Linear Discr...	root	ipykernel...	-	sklearn	0	0.7	0.727	-	-
22 minutes ago		Ridge Classif...	root	ipykernel...	-	sklearn	0.784	0.71	0.74	1.0	-
22 minutes ago		Logistic Reg...	root	ipykernel...	-	sklearn	0.817	0.736	0.758	-	-
22 minutes ago		Decision Tre...	root	ipykernel...	-	sklearn	0.818	0.736	0.758	-	-
22 minutes ago		Extra Trees C...	root	ipykernel...	-	sklearn	0.819	0.737	0.759	-	-
22 minutes ago		Random For...	root	ipykernel...	-	sklearn	0.821	0.738	0.761	-	-
22 minutes ago		Extreme Gra...	root	ipykernel...	-	sklearn					

2. Screenshot of one experiments with all the artifacts visible

mlflow 1.26.1 Experiments Models GitHub Docs

Lead_Scoring_Model_Experimentation_After_Dropping_Features > Light Gradient Boosting Machine

Light Gradient Boosting Machine

Status: UNFINISHED Source: ipykernel_launcher.py User: root

Lifecycle Stage: active Parent Run: e2136f015c174733b6021972553c4d3

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/airflow/mlruns/3/7d5758ff397c49e99d0a1171166b2a08/artifacts/Confusion Matrix.png Size: 7.03KB

	22892	12931
0		
1	5875	29992

mlflow 1.26.1 Experiments Models GitHub Docs

Lead_Scoring_Model_Experimentation_After_Dropping_Features > Light Gradient Boosting Machine

Light Gradient Boosting Machine

Status: UNFINISHED Source: ipykernel_launcher.py User: root

Lifecycle Stage: active Parent Run: e2136f015c174733b6021972553c4d3

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/airflow/mlruns/3/7d5758ff397c49e99d0a1171166b2a08/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.

Register Model

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:

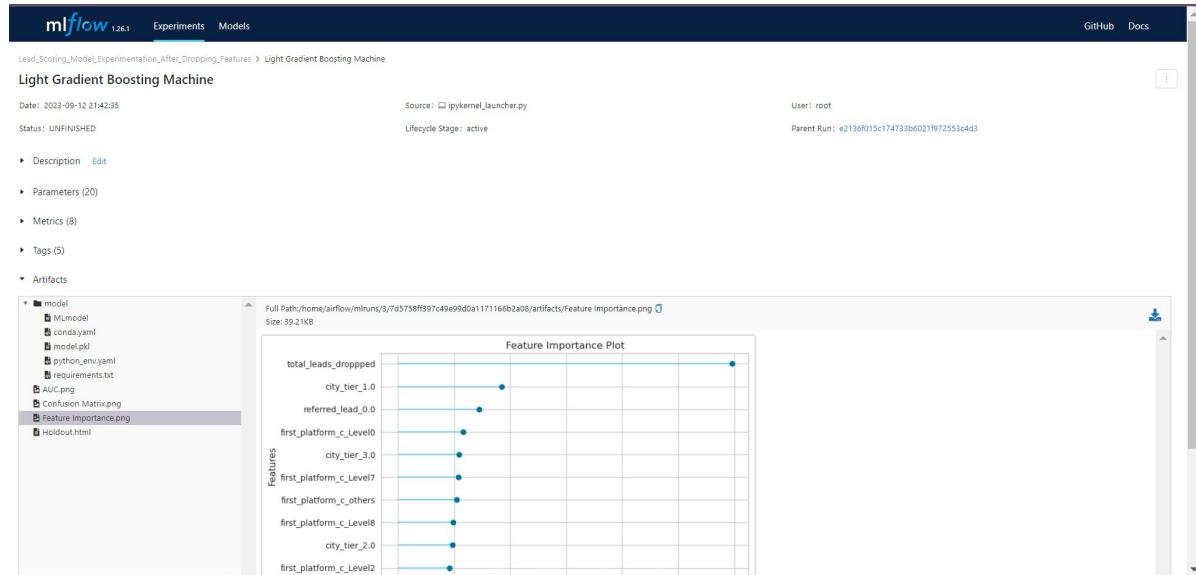
```
import mlflow
logged_model = 'runs:/7d5758ff397c49e99d0a1171166b2a08/model'

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



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

Name	Latest Version	Stage	Last Modified	Tags
LightGBM	Version 1	Production	2025-04-09 22:16:25	-

Versions	All	Active 1	Compare	
<input type="checkbox"/> Version	Registered at	Created by	Stage	Description
<input checked="" type="checkbox"/> Version 1	2025-04-09 21:12:21		Production	

Screenshot of successful execution Airflow DAG in List

The screenshot shows the Airflow web interface with the following details:

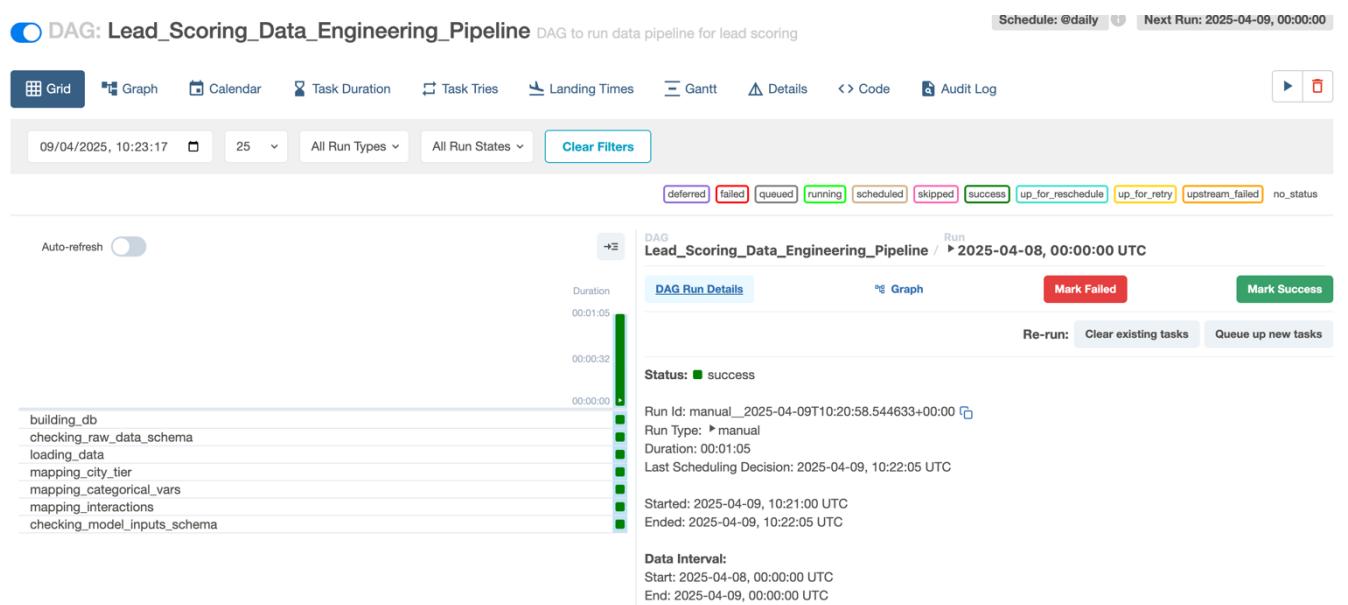
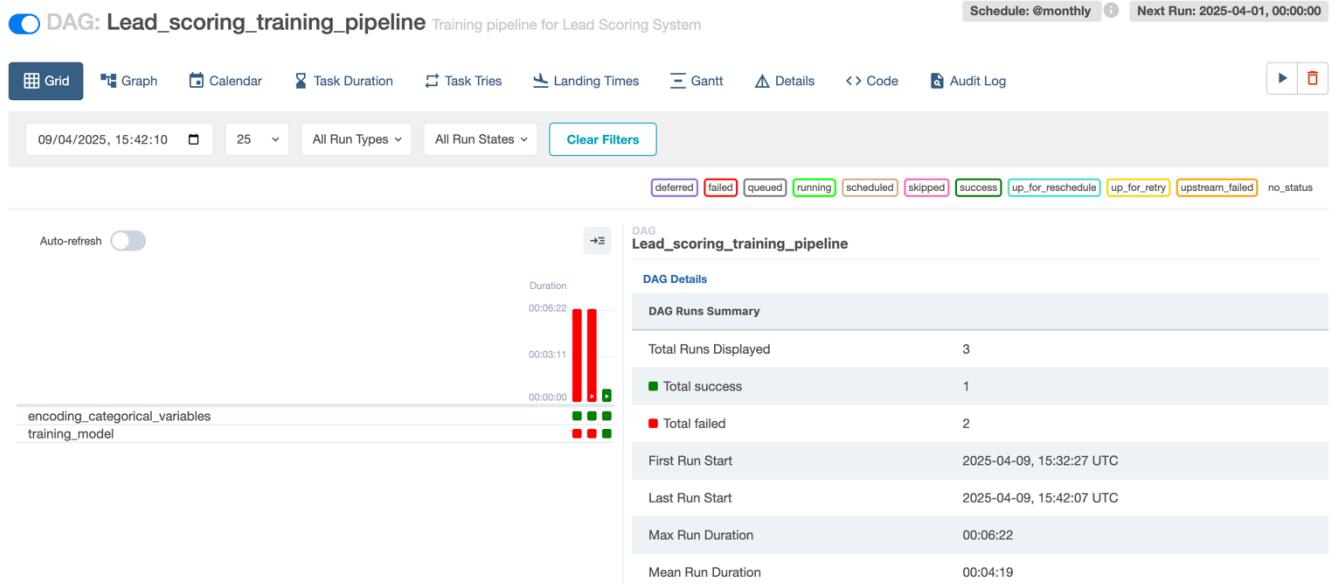
- Header:** Airflow, DAGs, Security, Browse, Admin, Docs, 16:58 UTC, UU.
- Alert Bar:** Triggered Lead_scoring_inference_pipeline, it should start any moment now.
- Warning Bar:** Do not use SQLite as metadata DB in production – it should only be used for dev/testing. We recommend using Postgres or MySQL. [Click here](#) for more information.
- Info Bar:** Do not use SequentialExecutor in production. [Click here](#) for more information.
- DAGs Section:** All 35, Active 1, Paused 34. Filter DAGs by tag and Search DAGs.
- DAG List:** A table showing the following DAGs:

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks
Lead_Scoring_Data_Engineering_Pipeline	airflow	1	@daily	2025-04-09, 10:20:58	2025-04-09, 00:00:00	7
Lead_scoring_inference_pipeline	airflow	1	@hourly	2025-04-09, 16:58:31	2025-04-09, 16:00:00	1
Lead_scoring_training_pipeline	airflow	1	@monthly	2025-04-09, 15:42:06	2025-04-01, 00:00:00	2
example_bash_operator	airflow	0	0 0 * * *		2025-04-08, 00:00:00	
example_branch_datetime_operator	airflow	0	@daily		2025-04-08, 00:00:00	
example_branch_datetime_operator_2		0				

Screenshot of successful execution Airflow DAG in Grid

The screenshot shows the Airflow web interface with the following details:

- Header:** DAG: Lead_scoring_inference_pipeline, Inference pipeline of Lead Scoring system, Schedule: @hourly, Next Run: 2025-04-09, 16:00:00.
- Toolbar:** Grid, Graph, Calendar, Task Duration, Task Tries, Landing Times, Gantt, Details, Code, Audit Log.
- Filter Bar:** 09/04/2025, 16:58:52, 25, All Run Types, All Run States, Clear Filters.
- Legend:** deferred, failed, queued, running, scheduled, skipped, success, up_for_reschedule, up_for_retry, upstream_failed, no_status.
- Auto-refresh:** On.
- Log View:** Shows log entries for tasks: encoding_categorical_variables, checking_input_features, generating_models_prediction, checking_model_prediction_ratio.
- DAG Details Panel:** DAG Lead_scoring_inference_pipeline, DAG Runs Summary, Total Runs Displayed: 2, Total success: 2, First Run Start: 2025-04-09, 16:58:10 UTC, Last Run Start: 2025-04-09, 16:58:32 UTC, Max Run Duration: 00:00:15, Mean Run Duration: 00:00:15.



Screenshot of successful execution Airflow DAG in Graph

