

➤ Model Building with all variables

The screenshot shows the mlflow Experiments page for 'Baseline_model_exp01'. The interface includes a search bar, a list of experiments, and a table of runs. The table has columns for Start Time, Duration, Run Name, User, Source, Version, Models, Metrics (AUC, Accuracy, F1, C), and Parameters (CPU Jobs, Categorical). The runs are sorted by Start Time, showing 11 matching runs.

Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy	F1	C	CPU Jobs	Categorical
4 minutes ago		Session Init...	root	ipykernel...	-	-	-	-	-	-	-1	4
1 minute ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739	0.762	-	-	-
3 minutes ago		Naive Bayes	root	ipykernel...	-	sklearn	0.734	0.663	0.727	-	-	-
3 minutes ago		Linear Discr...	root	ipykernel...	-	sklearn	0.773	0.701	0.728	-	-	-
3 minutes ago		Ridge Classi...	root	ipykernel...	-	sklearn	0	0.701	0.728	-	-	-
3 minutes ago		Logistic Reg...	root	ipykernel...	-	sklearn	0.784	0.71	0.74	1.0	-	-
3 minutes ago		Decision Tre...	root	ipykernel...	-	sklearn	0.817	0.736	0.758	-	-	-
3 minutes ago		Extra Trees C...	root	ipykernel...	-	sklearn	0.818	0.737	0.758	-	-	-
3 minutes ago		Random For...	root	ipykernel...	-	sklearn	0.819	0.737	0.759	-	-	-
3 minutes ago		Extreme Gra...	root	ipykernel...	-	sklearn	0.821	0.738	0.762	-	-	-
3 minutes ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739	0.762	-	-	-

➤ LGBM Model with artifacts

The screenshot shows the mlflow Models page for 'Baseline_model_exp01' with the title 'Light Gradient Boosting Machine'. It displays the model's date (2023-08-15 07:59:19), status (UNFINISHED), source (ipykernel_launcher.py), user (root), and parent run (b856b96641549de855a8b2eadc02a3). The page includes sections for Description, Parameters (20), Metrics (8), Tags (5), and Artifacts. The Artifacts section shows a list of files: Mlmodel,conda.yaml,model.pkl,python_env.yaml,requirements.txt,AUC.png,Confusion Matrix.png,Feature Importance.png, and Holdout.html. The 'MLflow Model' section provides a description and a 'Make Predictions' section with a code snippet for loading and predicting with the model.

```
import mlflow
logged_model = 'runs:/b856b96641549de855a8b2eadc02a3/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')

# Predict on a Spark DataFrame.
columns = list(df.columns)
```

➤ Model Building after dropping variables

The screenshot shows the mlflow Experiments page for 'Baseline_model_exp02'. The interface includes a search bar, a list of experiments, and a table of runs. The table has columns for Start Time, Duration, Run Name, User, Source, Version, Models, Metrics (AUC, Accuracy, F1, C), and Parameters (CPU Jobs, Categorical). The runs are sorted by Start Time, showing 10 matching runs.

Start Time	Duration	Run Name	User	Source	Version	Models	AUC	Accuracy	F1	C	CPU Jobs	Categorical
4 minutes ago		Session Init...	root	ipykernel...	-	-	-	-	-	-	-1	4
34 seconds ago		Naive Bayes	root	ipykernel...	-	sklearn	0.734	0.67	0.723	-	-	-
34 seconds ago		Linear Discr...	root	ipykernel...	-	sklearn	0.773	0.7	0.728	-	-	-
35 seconds ago		Ridge Classi...	root	ipykernel...	-	sklearn	0	0.7	0.728	-	-	-
35 seconds ago		Logistic Reg...	root	ipykernel...	-	sklearn	0.784	0.71	0.74	1.0	-	-
35 seconds ago		Decision Tre...	root	ipykernel...	-	sklearn	0.817	0.736	0.758	-	-	-
36 seconds ago		Extra Trees C...	root	ipykernel...	-	sklearn	0.817	0.737	0.758	-	-	-
36 seconds ago		Random For...	root	ipykernel...	-	sklearn	0.818	0.737	0.759	-	-	-
36 seconds ago		Extreme Gra...	root	ipykernel...	-	sklearn	0.821	0.738	0.762	-	-	-
38 seconds ago		Light Gradie...	root	ipykernel...	-	sklearn	0.821	0.739	0.762	-	-	-

➤ LGBM Model with artifacts, after dropping variables

The screenshot shows the MLflow Model Registry interface. At the top, there's a navigation bar with 'mlflow 1.26.1', 'Experiments', and 'Models'. Below this, the breadcrumb 'Baseline_model_exp02 > Light Gradient Boosting Machine' is visible. The main title is 'Light Gradient Boosting Machine'. Metadata includes 'Date: 2023-08-15 08:14:24', 'Source: ipykernel_launcher.py', 'User: root', 'Status: UNFINISHED', 'Lifecycle Stage: active', and 'Parent Run: 3bd56af2260340d898737e0c76bc3962'. On the left, there's a sidebar with 'Description', 'Parameters (20)', 'Metrics (8)', 'Tags (5)', and 'Artifacts'. The 'Artifacts' section is expanded, showing a file tree for the model. The main content area displays the 'MLflow Model' details, including a 'Model schema' table (empty) and a 'Make Predictions' section with code snippets for loading and predicting with the model.

➤ Airflow DAG for Data Pipeline

The screenshot shows the Airflow web interface. The top navigation bar includes 'Airflow', 'DAGs', 'Security', 'Browse', 'Admin', and 'Docs'. A warning banner states: '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.' Below this, another warning says: 'Do not use SequentialExecutor in production. Click here for more information.' The 'DAGs' section shows a list of DAGs with columns for DAG name, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, Actions, and Links. The DAGs listed are 'Lead_Scoring_Data_Engineering_Pipeline', 'example_branch_operator', 'example_branch_datetime_operator', and 'example_branch_datetime_operator_2'. The 'Lead_Scoring_Data_Engineering_Pipeline' DAG is selected, showing its details. The DAG is in a 'running' state with a schedule of '@daily' and a next run of '2023-08-25, 00:00:00'. The DAG graph shows a sequence of tasks: 'building_db' -> 'checking_raw_data_schema' -> 'loading_data' -> 'mapping_city_tier' -> 'mapping_categorical_vars' -> 'mapping_interactions' -> 'checking_model_inputs_schema'. The 'PythonOperator' is highlighted, and a legend shows various task states: deferred, failed, queued, running, scheduled, skipped, success, up_for_reschedule, up_for_retry, upstream_failed, and no_status.

➤ Lead_Scoring_Training_Pipeline experiment with artifacts

The MLflow Experiments page displays the 'run_LightGBM' experiment. The top navigation bar includes 'mlflow 1.26.1', 'Experiments', and 'Models'. The experiment details show it was run on 2023-08-26 at 20:32:28, with a duration of 4.6s, source 'ipykernel_launcher.py', user 'root', and status 'FINISHED'. The lifecycle stage is 'active'. Below the details are links for Description, Parameters (20), Metrics (1), Tags, and Artifacts. The Artifacts section shows a file tree for 'models' containing 'MLmodel', 'conda.yaml', 'model.pkl', 'python_env.yaml', and 'requirements.txt'. The 'MLflow Model' section provides the full path and a link to the model registry. The 'Model schema' section is empty, and the 'Make Predictions' section shows a code snippet for loading the model and making predictions on a Spark DataFrame.

➤ LightGBM model in Production

The MLflow Models page displays the 'LightGBM' model in production. The top navigation bar includes 'mlflow 1.26.1', 'Experiments', and 'Models'. The model details show it is 'Version 4', registered at 2023-08-26 20:32:32, with a stage of 'Production' and last modified at 2023-08-26 20:33:32. The source run is 'run_LightGBM'. Below the details are links for Description, Tags, and Schema. The Schema section is empty, and the 'Make Predictions' section shows a code snippet for loading the model and making predictions on a Spark DataFrame.

➤ Airflow DAG for Training Pipeline

The Airflow DAGs page displays a list of DAGs. The top navigation bar includes 'Airflow', 'DAGs', 'Security', 'Browse', 'Admin', and 'Docs'. The DAGs section shows a table with columns for DAG, Owner, Runs, Schedule, Last Run, Next Run, Recent Tasks, and Actions. The 'Lead_Scoring_Data_Engineering_Pipeline' DAG is highlighted, showing it is active and has a daily schedule. The 'Lead_scoring_training_pipeline' DAG is also shown, with a monthly schedule. Other DAGs include 'example_branch_operator', 'example_branch_datetime_operator', 'example_branch_datetime_operator_2', 'example_branch_dop_operator_v3', and 'example_branch_labels'.

Airflow DAGs Security Browse Admin Docs 01:45 UTC SJ

Triggered Lead_scoring_training_pipeline, it should start any moment now.

DAG: Lead_scoring_training_pipeline Training pipeline for Lead Scoring System **queued** Schedule: @monthly Next Run: 2023-08-01, 00:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details Code Audit Log

2023-08-26T01:44:22Z Runs 25 Run manual__2023-08-26T01:44:21.045543+00:00 Layout Left > Right Update Find Task...

PythonOperator

deferred failed queued running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

Auto-refresh

```

graph LR
    A[encoding_categorical_variables] --> B[training_model]
  
```

➤ Airflow DAG for Inference Pipeline

Airflow DAGs Security Browse Admin Docs 01:57 UTC SJ

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.

Do not use **SequentialExecutor** in production. [Click here](#) for more information.

DAGs

At 35 Active 1 Paused 64 Filter DAGs by tag Search DAGs

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions
Lead_Scoring_Data_Engineering_Pipeline	airflow	1	@daily	2023-08-25, 03:37:25	2023-08-26, 00:00:00	1	▶ 🗑
Lead_scoring_inference_pipeline	airflow	1	@hourly	2023-08-26, 00:00:00	2023-08-26, 01:00:00	2 1 1	▶ 🗑
Lead_scoring_training_pipeline	airflow	1	@monthly	2023-08-26, 01:44:21	2023-08-01, 00:00:00	2	▶ 🗑
example_bash_operator	airflow	1	@0***		2023-08-25, 00:00:00		▶ 🗑
example_branch_datetime_operator	airflow	1	@daily		2023-08-25, 00:00:00		▶ 🗑

With leadscoring.csv:

Airflow DAGs Security Browse Admin Docs 15:07 UTC SJ

DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system **success** Schedule: @hourly Next Run: 2023-08-26, 15:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details Code Audit Log

2023-08-26T15:05:42Z Runs 25 Run manual__2023-08-26T15:05:41.438802+00:00 Layout Left > Right Update Find Task...

PythonOperator

deferred failed queued running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

Auto-refresh

```

graph LR
    A[encoding_categorical_variables] --> B[checking_input_features]
    B --> C[generating_models_prediction]
    C --> D[checking_model_prediction_ratio]
  
```

With leadscoring_inference.csv:

Airflow DAGs Security Browse Admin Docs 15:19 UTC 6.1

Triggered Lead_scoring_inference_pipeline, it should start any moment now.

DAG: Lead_scoring_inference_pipeline Inference pipeline of Lead Scoring system queued Schedule: @hourly Next Run: 2023-08-26, 15:00:00

Grid Graph Calendar Task Duration Task Tries Landing Times Gantt Details <> Code Audit Log

2023-08-26T15:18:57Z Runs 25 Run manual_2023-08-26T15:18:56.481308+00:00 Layout Left > Right Update Find Task...

PythonOperator deferred failed queued running scheduled skipped success up_for_reschedule up_for_retry upstream_failed no_status

Auto-refresh

```
graph LR; encoding_categorical_variables --> checking_input_features; checking_input_features --> generating_models_prediction; generating_models_prediction --> checking_model_prediction_ratio;
```