DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

Schedule: @daily ⓘ   Next Run: 2023-10-17, 05:30:00

| Grid | Graph | Calendar | ⌛ Task Duration | Task Tries | Landing Times | Gantt | ⚠ Details | <> Code | Audit Log | ▶ | 🗑 |

📅 2023-10-17 16:31:32+05   Runs 25 ▾   Update   ☐ Cumulative Duration

Legend: ●building_db ●checking_raw_data_schema ●loading_data ●mapping_city_tier ●mapping_categorical_vars ●mapping_interactions ●checking_model_inputs_schema

Y-axis labels: 0.13, 0.13, 0.12, 0.12, 0.11

Y-axis title: Duration (seconds)

```python
        pass
##############################################################################
# Define the relation between the tasks
# ##############################################################################
building_db_task = PythonOperator(
    task_id='building_db',
    python_callable=build_dbs,
    dag=ML_data_cleaning_dag
)

checking_raw_data_schema_task = PythonOperator(
    task_id='checking_raw_data_schema',
    python_callable=raw_data_schema_check,
    dag=ML_data_cleaning_dag
)

loading_data_task = PythonOperator(
    task_id='loading_data',
    python_callable=load_data_into_db,
    dag=ML_data_cleaning_dag
)

mapping_city_tier_task = PythonOperator(
    task_id='mapping_city_tier',
    python_callable=map_city_tier,
    dag=ML_data_cleaning_dag
)

mapping_categorical_vars_task = PythonOperator(
    task_id='mapping_categorical_vars',
    python_callable=map_categorical_vars,
    dag=ML_data_cleaning_dag
)
```

Filter files by name

/ ... / 01_data_pipeline / notebooks /

Name ▲	Last Modified
Data	25 minutes ago
Maps	a day ago
profile_report	a day ago
data_cleaning_template.ipynb	2 hours ago

DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

Schedule: @daily ⓘ   Next Run: 2023-10-17, 05:30:00

⊞ Grid   ⌗ Graph   ▦ Calendar   ⧗ Task Duration   ⇄ Task Tries   ⤓ Landing Times   ≡ Gantt   ⚠ Details   <> Code   🔍 Audit Log   ▶ 🗑

Toggle Wrap

```
1    ################################################################################
2    # Import necessary modules
3    # ################################################################################
4
5
6    from airflow import DAG
7    from airflow.operators.python import PythonOperator
8
9    from datetime import datetime, timedelta
10
11
12   ################################################################################
13   # Define default arguments and DAG
14   # ################################################################################
15
16   default_args = {
17       'owner': 'airflow',
18       'start_date': datetime(2022,7,30),
19       'retries' : 1,
20       'retry_delay' : timedelta(seconds=5)
21   }
22
23
24   ML_data_cleaning_dag = DAG(
25                   dag_id = 'Lead_Scoring_Data_Engineering_Pipeline',
26                   default_args = default_args,
```

/ Assignment / unit_test /

Name	Last Modified
city_tier_mapping.py	a day ago
constants.py	a day ago
data_validation_checks.py	a day ago
interaction_mapping.csv	a day ago
lead_scoring_data_pipeline.py	an hour ago
leadscoring_test.csv	a day ago
schema.py	a day ago
significant_categorical_level.py	a day ago
test_with_pytest.py	a day ago
unit_test_cases.db	a day ago
utils.py	a day ago

List Log

Search ⌄

«   <   **1**   2   3   4   >   »   Page size ⌄   ←   **Record Count:** 327

Id	Dttm	Dag Id	Task Id	Event	Logical Date	Owner	Extra
334	2023-10-17, 16:35:17	Lead_Scoring_Data_Engineering_Pipeline		gantt		Adityapal	[('dag_id', 'Lead_Scoring_Data_Engineering_Pipeline')]
333	2023-10-17, 16:34:56	Lead_Scoring_Data_Engineering_Pipeline		landing_times		Adityapal	[('days', '30'), ('dag_id', 'Lead_Scoring_Data_Engineering_Pipelin
332	2023-10-17, 16:34:32	Lead_Scoring_Data_Engineering_Pipeline		tries		Adityapal	[('days', '30'), ('dag_id', 'Lead_Scoring_Data_Engineering_Pipelin
331	2023-10-17, 16:33:44	Lead_Scoring_Data_Engineering_Pipeline		duration		Adityapal	[('days', '30'), ('dag_id', 'Lead_Scoring_Data_Engineering_Pipelin
330	2023-10-17, 16:33:16	Lead_Scoring_Data_Engineering_Pipeline		calendar		Adityapal	[('dag_id', 'Lead_Scoring_Data_Engineering_Pipeline')]
329	2023-10-17, 16:32:48	Lead_Scoring_Data_Engineering_Pipeline		graph		Adityapal	[('dag_id', 'Lead_Scoring_Data_Engineering_Pipeline')]
328	2023-10-17, 16:31:42	Lead_Scoring_Data_Engineering_Pipeline	checking_model_inputs_schema	success	2023-10-17, 16:31:32	airflow	
327	2023-10-17, 16:31:42	Lead_Scoring_Data_Engineering_Pipeline	checking_model_inputs_schema	cli_task_run		root	{"host_name": "91ceab47785b", "full_command": "['/opt/conda/bir 'tasks', 'run', 'Lead_Scoring_Data_Engineering_Pipeline', 'checking_model_inputs_schema', 'manual__2023-10-17T11:01:32.831455+00:00', '--local', '--subdir', 'DAGS_FOLDER/Lead_scoring_data_pipeline/lead_scoring_data

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

DAG Details

success 56

Schedule Interval	@daily
Catchup	False
Started	True
End Date	None
Max Active Runs	0 / 16
Concurrency	16
Default Args	{'owner': 'airflow', 'retries': 1, 'retry_delay': datetime.timedelta(seconds=5), 'start_date': DateTime(2022, 7, 30, 0, 0, 0, tzinfo=Timezone('UTC'))}
Tasks Count	7
Task IDs	['building_db', 'checking_raw_data_schema', 'loading_data', 'mapping_city_tier', 'mapping_categorical_vars', 'mapping_interactions', 'checking_model_inputs_schema']
Relative file location	Lead_scoring_data_pipeline/lead_scoring_data_pipeline.py
Owner	airflow
DAG Run Timeout	None

Airflow

DAGs   Security   Browse   Admin   Docs   16:37 IST (+05:30)   AP

DAGs

All `33`   Active `1`   Paused `32`   Filter DAGs by tag   Search DAGs

	DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Action
🔵	**Lead_Scoring_Data_Engineering_Pipeline**	airflow	⟳ 17	@daily	2023-10-17, 16:31:32	2023-10-17, 05:30:00	7	▶ 🗑

«   ‹   1   ›   »   Showing **1-1** of **1** DAGs

Version: **v2.3.3**

11.00

10.95

Landing Time (hours)

10.90

10.85

10.80

16:21, 17 Oct   16:23, 17 Oct   16:25, 17 Oct   16:26, 17 Oct   16:28, 17 Oct   16:30, 17 Oct   **16:31, 17 Oct 2023**

▦ Grid | ⬛ Graph | 🗓 Calendar | ⏳ Task Duration | ⇄ Task Tries | ⬏ Landing Times | ≡ Gantt | ⚠ Details | <> Code | 🔍 Audit Log

▶ 🗑

📅 2023-10-17T16:31:33+0! | Runs | 25 ⌄ | Run | manual__2023-10-17T11:01:32.831455+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 ⟳

```
building_db → checking_raw_data_schema → loading_data → mapping_categorical_vars → mapping_interactions → checking_model_inputs_schema
                                          loading_data → mapping_city_tier → mapping_interactions
```

DAG: **Lead_Scoring_Data_Engineering_Pipeline** DAG to run data pipeline for lead scoring

Schedule: @daily   Next Run: 2023-10-17, 05:30:00

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

2023-10-17T16:31:33+05   Runs   25   Run   manual__2023-10-17T11:01:32.831455+00:00   Update

```python
30  )
31
32  ###########################################################################
33  # Create a task for build_dbs() function with task_id 'building_db'
34  # ###########################################################################
35  def build_dbs():
36      # Your logic for building databases
37      pass
38  ###########################################################################
39  # Create a task for raw_data_schema_check() function with task_id 'checking_raw_data_schema'
40  # ###########################################################################
41  def raw_data_schema_check():
42      # Your logic for checking raw data schema
43      pass
44  ###########################################################################
45  # Create a task for load_data_into_db() function with task_id 'loading_data'
46  # ###########################################################################
47  def load_data_into_db():
48      # Your logic for loading data into a database
49      pass
50  ###########################################################################
51  # Create a task for map_city_tier() function with task_id 'mapping_city_tier'
52  # ###########################################################################
53  def map_city_tier():
54      # Your logic for mapping city tiers
55      pass
56  ###########################################################################
57  # Create a task for map_categorical_vars() function with task_id 'mapping_categorical_vars'
58  # ###########################################################################
59  def map_categorical_vars():
60      # Your logic for mapping categorical variables
61      pass
62  ###########################################################################
```

/

Name	▲	Last Modified
📁 airflow		in a few seconds
📁 Assignment		an hour ago
📁 dags		a day ago
📁 dashboard		38 minutes ago
📁 data		a day ago
📁 database		a day ago
📁 notebooks		a day ago
📁 scripts		a day ago

Filter files by name

Terminal 1 ✕   Terminal 2 ✕   ☰ lead_scoring_data_pipeline.py ✕   +

```
108        task_id='mapping_interactions',
109        python_callable=interactions_mapping,
110        dag=ML_data_cleaning_dag
111    )
112
113    checking_model_inputs_schema_task = PythonOperator(
114        task_id='checking_model_inputs_schema',
115        python_callable=model_input_schema_check,
116        dag=ML_data_cleaning_dag
117    )
118
119    # Define the task dependencies
120    building_db_task >> checking_raw_data_schema_task >> loading_data_task
121    loading_data_task >> mapping_city_tier_task
122    loading_data_task >> mapping_categorical_vars_task
123    mapping_city_tier_task >> mapping_interactions_task
124    mapping_categorical_vars_task >> mapping_interactions_task
125    mapping_interactions_task >> checking_model_inputs_schema_task
126
127
```

/ ··· / 01_data_pipeline / scripts /

Name ▲	Last Modified
📁 data	a day ago
🔖 city_tier_mapping.py	a day ago
🔖 constants.py	a day ago
🔖 data_validation_chec...	a day ago
⊞ interaction_mapping....	a day ago
🔖 lead_scoring_data_pi...	an hour ago
⊞ leadscoring_test.csv	a day ago
🔖 schema.py	a day ago
🔖 significant_categorica...	a day ago
🔖 test_with_pytest.py	a day ago
📄 unit_test_cases.db	a day ago
🔖 utils.py	a day ago

17-10-2023 16:31:30 📅   25 ⌄   All Run Types ⌄   All Run States ⌄   **Clear Filters**

deferred   failed   queued   running   scheduled   skipped   success   up_for_reschedule   up_for_retry   upstream_failed   no_status

Auto-refresh ⬤   →☰

DAG
Lead_Scoring_Data_Engineering_Pipeline

Duration   Oct 17, 15:59   Oct 17, 16:21

DAG Details

DAG Runs Summary	
Total Runs Displayed	17
■ Total success	17
First Run Start	2023-10-17, 15:59:29 IST
Last Run Start	2023-10-17, 16:31:33 IST
Max Run Duration	00:00:10
Mean Run Duration	00:00:04
Min Run Duration	00:00:00

building_db
checking_raw_data_schema
loading_data
mapping_city_tier
mapping_categorical_vars
mapping_interactions
checking_model_inputs_schema

DAG Summary	
Total Tasks	7
PythonOperators	7

Terminal 1 ✕   Terminal 2 ✕   ☰ lead_scoring_data_pipeline.py ✕   +

```
69    # Create a task for model_input_schema_check() function with task_id 'checking_model_inputs_schema'
70    ##################################################################################
71    def model_input_schema_check():
72        # Your logic for checking model input schema
73        pass
74    ##################################################################################
75    # Define the relation between the tasks
76    ##################################################################################
77    building_db_task = PythonOperator(
78        task_id='building_db',
79        python_callable=build_dbs,
80        dag=ML_data_cleaning_dag
81    )
82
83    checking_raw_data_schema_task = PythonOperator(
84        task_id='checking_raw_data_schema',
85        python_callable=raw_data_schema_check,
86        dag=ML_data_cleaning_dag
87    )
88
89    loading_data_task = PythonOperator(
90        task_id='loading_data',
91        python_callable=load_data_into_db,
92        dag=ML_data_cleaning_dag
93    )
94
95    mapping_city_tier_task = PythonOperator(
96        task_id='mapping_city_tier',
97        python_callable=map_city_tier,
98        dag=ML_data_cleaning_dag
99    )
100
101   mapping_categorical_vars_task = PythonOperator(
102       task_id='mapping_categorical_vars',
103       python_callable=map_categorical_vars,
104       dag=ML_data_cleaning_dag
```

File browser

/ ⋯ / 01_data_pipeline / scripts /

Name	Last Modified
📁 data	a day ago
🔲 city_tier_mapping.py	a day ago
🔲 constants.py	a day ago
🔲 data_validation_chec...	a day ago
▦ interaction_mapping....	a day ago
🔲 lead_scoring_data_pi...	an hour ago
▦ leadscoring_test.csv	a day ago
🔲 schema.py	a day ago
🔲 significant_categorica...	a day ago
🔲 test_with_pytest.py	a day ago
🗎 unit_test_cases.db	a day ago
🔲 utils.py	a day ago

Simple   3 ⑤ 0 ⚙ Python   Ln 1, Col 1   Spaces: 4   lead_scoring_data_pipeline.py

Terminal 1 | Terminal 2 | constants.py | constants.py

```python
1   DB_PATH = '/home/Assignment'
2   DB_FILE_NAME = 'database'
3
4   DB_FILE_MLFLOW = 'airflow.cfg'
5
6   TRACKING_URI = 'https://91ceab47785b1.notebookse.jarvislabs.ai/dags/Lead_Scoring_Data_Engineering_Pipeline/grid'
7   EXPERIMENT = 'Lead_scoring_data_pipeline'
8
9
10  # model config imported from pycaret experimentation
11  model_config = 'leadscoring.csv'
12
13  # list of the features that needs to be there in the final encoded dataframe
14  ONE_HOT_ENCODED_FEATURES = 'Lead_scoring_mlflow_production'
15  # list of features that need to be one-hot encoded
16  FEATURES_TO_ENCODE = 'lead_scoring_data_cleaning.db'
17
```

Filter files by name

/ ... / 03_inference_pipeline / scripts /

Name	Last Modified
constants.py	a day ago
lead_scoring_data_pi...	an hour ago
utils.py	a day ago

Simple   3   0   Python   Ln 17, Col 1   Spaces: 4   constants.p

Filter files by name

📁 / ··· / 01_data_pipeline / scripts /

Name ▲	Last Modified
📁 data	a day ago
📄 city_tier_mapping.py	a day ago
📄 constants.py	a day ago
📄 data_validation_chec...	a day ago
⊞ interaction_mapping....	a day ago
📄 lead_scoring_data_pi...	an hour ago
⊞ leadscoring_test.csv	a day ago
📄 schema.py	a day ago
📄 significant_categorica...	a day ago
📄 test_with_pytest.py	a day ago
📄 unit_test_cases.db	a day ago
📄 utils.py	a day ago

```python
1  ################################################################################
2  # Import necessary modules
3  # ##############################################################################
4
5
6  from airflow import DAG
7  from airflow.operators.python import PythonOperator
8
9  from datetime import datetime, timedelta
10
11
12 ################################################################################
13 # Define default arguments and DAG
14 # ##############################################################################
15
16 default_args = {
17     'owner': 'airflow',
18     'start_date': datetime(2022,7,30),
19     'retries' : 1,
20     'retry_delay' : timedelta(seconds=5)
21 }
22
23
24 ML_data_cleaning_dag = DAG(
25             dag_id = 'Lead_Scoring_Data_Engineering_Pipeline',
26             default_args = default_args,
27             description = 'DAG to run data pipeline for lead scoring',
28             schedule_interval = '@daily',
29             catchup = False
30 )
31
32 ################################################################################
33 # Create a task for build_dbs() function with task_id 'building_db'
34 # ##############################################################################
35 def build_dbs():
36     # Your logic for building databases
```

Simple ⬤○   3   ⑤   0   ⚙   Python   ◈   Ln 1, Col 1   Spaces: 4   lead_scoring_data_pipeline.py

DAG: Lead_Scoring_Data_Engineering_Pipeline DAG to run data pipeline for lead scoring

Schedule: @daily Next Run: 2023-10-17, 05:30:00

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

success ▮▮▮▮▮▮▮▮▮▮ failed   ▮ running   · planned   ☐ no_status