# DAG: walmart-ml-workflow A simple Machine Learning workflow for Walmart Sales

**Schedule: 1 day, 0:00:00 (/dagrun/list/?_flt_3_dag_id=walmart-ml-workflow)** **Next Run: 2022-08-26, 17:00:00**

Grid (/dags/walmart-ml-workflow/grid?root=) Graph (/dags/walmart-ml-workflow/graph?root=)

Calendar (/dags/walmart-ml-workflow/calendar)

Task Duration (/dags/walmart-ml-workflow/duration?days=30&root=)

Task Tries (/dags/walmart-ml-workflow/tries?days=30&root=)

Landing Times (/dags/walmart-ml-workflow/landing-times?days=30&root=)

Gantt (/dags/walmart-ml-workflow/gantt?root=) Details (/dags/walmart-ml-workflow/details)

Code (/dags/walmart-ml-workflow/code?root=)

Audit Log (/dags/walmart-ml-workflow/audit_log?root=)

▶ (/delete?dag_id=walmart-ml-workflow&redirect_url=%2Fdags%2Fwalmart-ml-workflow%2Fcode)

## Parsed at: 2023-01-13, 10:49:25

Toggle Wrap

```python
from datetime import datetime, timedelta

from airflow import DAG
from airflow.operators.bash import BashOperator

# instantiates a directed acyclic graph
with DAG(
    'walmart-ml-workflow',
    default_args={
        'owner': 'Francisco Estevez', # YOUR NAME HERE
        'depends_on_past': False,
        'email': ['francisco@estevez.work'], # YOUR EMAIL HERE
        'email_on_failure': False,
        'email_on_retry': False,
        'retries': 1,
        'retry_delay': timedelta(minutes=5),
        },
    description='A simple Machine Learning workflow for Walmart Sales',
    schedule_interval=timedelta(days=1),
    start_date=datetime(2022, 8, 27),
    tags=['walmart', 'ml', 'workflow'] # OPTIONAL: tags
) as dag:

    # instantiate tasks using Operators.
    # BashOperator defines tasks that execute bash scripts. In this case, we run Pythor
    get_store_data = BashOperator(
        task_id='get_store_data',
        bash_command='python ~/airflow/scripts/get_store_data.py',
    )
    train = BashOperator(
        task_id='train',
        depends_on_past=False,
        bash_command='python ~/airflow/scripts/train.py',
        retries=3,
    )

    # sets the ordering of the DAG. The >> directs the 2nd task to run after the 1st ta
    # get the store data first, then train.
    get_store_data >> train
```

Version: v2.5.0 (https://pypi.python.org/pypi/apache-airflow/2.5.0)
Git Version: **.release:2.5.0+fa2bec042995004f45b914dd1d66b466ccced410**