10. Pipeline Orchestration

10.1 Orchestration Approach

The end-to-end data and machine learning pipeline is orchestrated using **Prefect**, a Python-native workflow orchestration framework that runs seamlessly on Windows and Linux environments.

Prefect is used to:

- Automate pipeline execution
- Define task dependencies
- Provide task-level logging and retries
- Monitor pipeline execution status

This orchestration layer ensures reliable and repeatable execution of the complete recommendation system pipeline.

10.2 Pipeline Workflow

The pipeline is executed as a single flow with the following sequence of stages:

```
Data Ingestion
  → Data Validation
    → Data Preparation and EDA
      → Feature Engineering
        → Feature Store Access
          → Model Training
            → Model Evaluation
```

Each stage corresponds to an independent task implemented in earlier pipeline steps.

10.3 Orchestration Implementation

The pipeline is implemented as a Prefect flow with multiple tasks, where each task invokes an existing processing script.

📄 Flow Definition

`src/orchestration/prefect_flow.py`

The flow coordinates execution of:

- Clickstream and product data ingestion
- Data validation checks
- Data preparation and exploratory analysis
- Feature engineering and storage in PostgreSQL
- Feature store access demonstration
- Model training and evaluation

10.4 Logging and Monitoring

Prefect provides built-in logging and monitoring capabilities:

- Each task logs execution start and completion status
- Failures are captured and retried automatically
- Downstream tasks are blocked on upstream task failures
- Execution status can be monitored via the Prefect UI or terminal logs

This enables effective observability and failure handling across the pipeline.

10.5 Execution and Verification

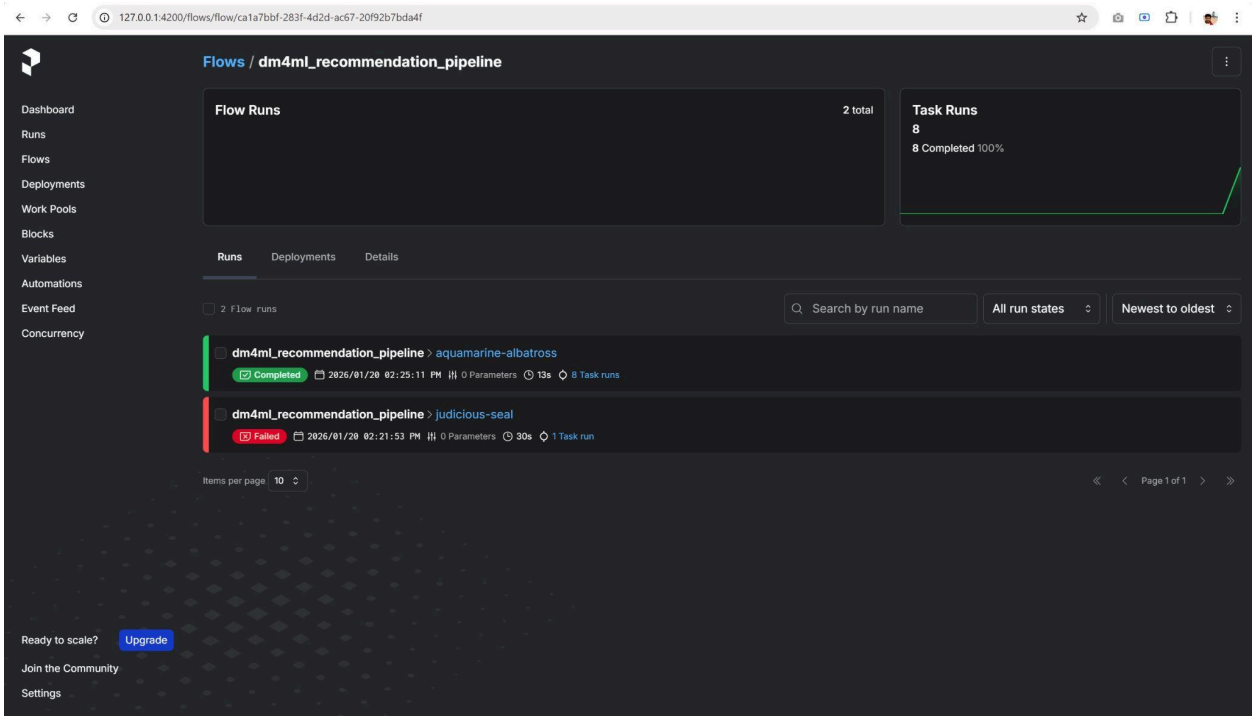
The pipeline is executed using:

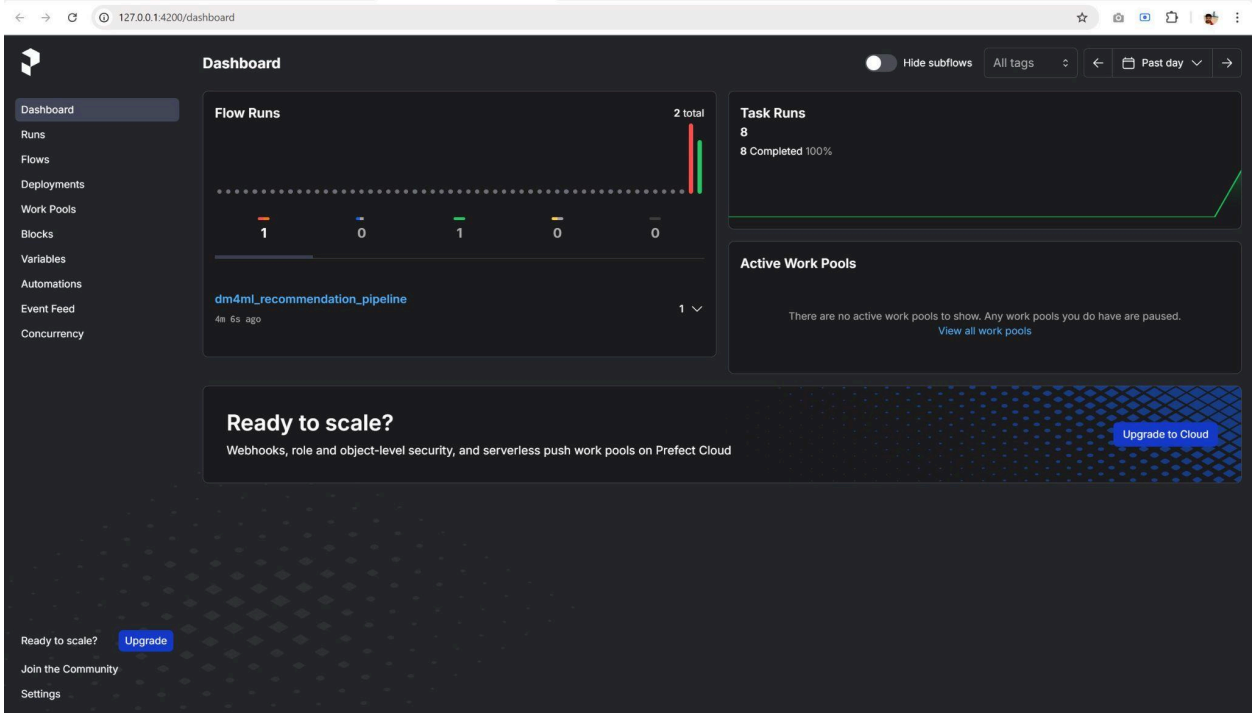
```
python -m src.orchestration.prefect_flow
```

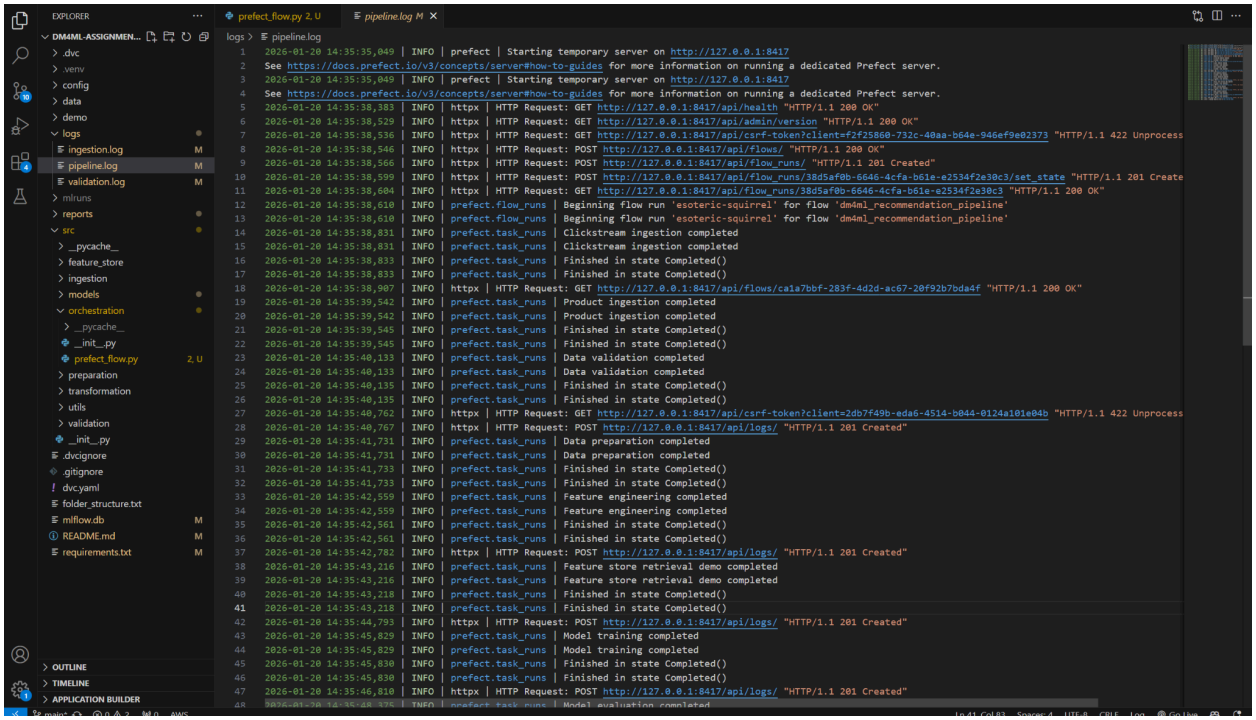
Successful execution is verified through:

- Console logs indicating completion of each task
- Prefect UI showing successful flow runs and task statuses

Screenshots or logs from successful executions are included as evidence of orchestration.


127.0.0.1:4200/flows/flow/ca1a7bbf-283f-4d2d-ac67-20f92b7bda4f
Flows / dm4ml_recommendation_pipeline
Dashboard
Runs
Flows
Deployments
Work Pools
Blocks
Variables
Automations
Event Feed
Concurrency
Flow Runs
2 total
Task Runs
8
8 Completed 100%
Runs
Deployments
Details
2 flow runs
Search by run name
All run states
Newest to oldest
dm4ml_recommendation_pipeline > aquamarine-albatross
Completed
2026/01/20 02:25:11 PM
0 Parameters
13s
8 Task runs
dm4ml_recommendation_pipeline > judicious-seal
Failed
2026/01/20 02:21:53 PM
0 Parameters
30s
1 Task run
Items per page
10
Page 1 of 1
Ready to scale?
Upgrade
Join the Community
Settings


127.0.0.1:4200/dashboard
Dashboard
Hide subflows
All tags
Past day
Dashboard
Runs
Flows
Deployments
Work Pools
Blocks
Variables
Automations
Event Feed
Concurrency
Flow Runs
2 total
1
0
1
0
0
dm4ml_recommendation_pipeline
4m 6s ago
1
Task Runs
8
8 Completed 100%
Active Work Pools
There are no active work pools to show. Any work pools you do have are paused.
View all work pools
Ready to scale?
Webhooks, role and object-level security, and serverless push work pools on Prefect Cloud
Upgrade to Cloud
Ready to scale?
Upgrade
Join the Community
Settings



10.6 Summary

The Prefect-based orchestration layer automates the entire data and machine learning workflow, ensuring structured execution, observability, and fault tolerance. This completes the end-to-end pipeline implementation for the recommendation system.