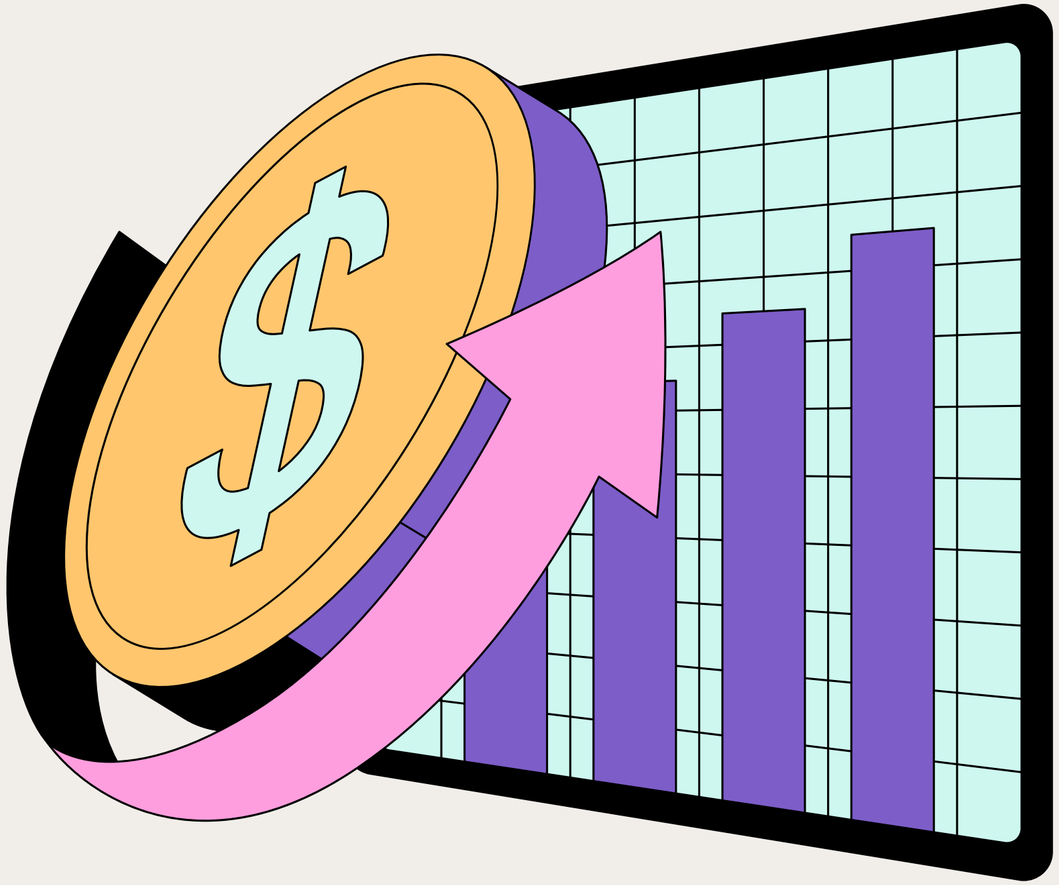
Global Market Index

Project Deliverables & Key



GMI_Analytics_V1.0

https://github.com/Workflomate/Global-Market-Project

Delivered: An End-to-End BI for Global Market Analysis

Goal: Develop a comprehensive data platform that collects and analyzes global stock index trends to support smart investment decisions.

Key Objectives Achieved:

- - Collected and analyzed global index time series from multiple financial sources.
- - Developed analytical models to evaluate index behavior, volatility, and forecast performance.
- - Created and deployed an interactive Dash web dashboard to visualize real-time market patterns and insights.
- - Automated end-to-end data workflows using Airflow to ensure accuracy, consistency, and scalability.

Project Workflow

1. Data & SQL Layer (sql/): create_tables.sql, load_data.py – ingest Kaggle CSV into MySQL staging.

2. DBT Models (dbt/): staging → intermediate → final (stg_global_index.sql, int_market_metrics.sql, fct_market_summary.sql).

3. Forecasting (forecasting/): model_forecast.py – 7-day predictions, MAE/RMSE metrics.

4. Airflow Orchestration (airflow/): global-market-dag.py – schedules ingestion, dbt runs, forecasting, dashboard update.

5. Dash Dashboard (dash/): app.py – interactive visualizations for regions, volatility, forecasts.

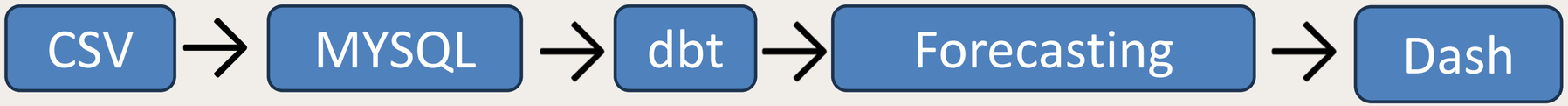
6. Architecture & Demo (docs/): architecture_diagram.png, Project_Documentation.md.

Project Deliverables

✓Technical documentation: README.md & docs/Project_Documentation.md.
✓Architecture diagram (docs/architecture_diagram.png).
✓Source code repository (structured, version-controlled).
✓dbt model lineage and artifacts (dbt/models, logs, manifest).
✓Forecasting script and evaluation (forecasting/model_forecast.py).
✓Interactive Dash dashboard (dash/app.py) with visualizations and filters.
✓Airflow DAG for orchestration (airflow/global-market-dag.py).
✓Setup & run instructions.

System Architecture

- Data Source: Kaggle Stock Exchange Dataset (CSV: open, high, low, close, volume).
- Ingestion: sql/load_data.py → MySQL staging tables (create_tables.sql).
- Transformation: dbt models (staging → intermediate → final) producing fct_market_summary.
- Forecasting: forecasting/model_forecast.py using Prophet for 7-day forecasts.
- Orchestration: airflow/global-market-dag.py automates inget data → run dbt → dbt test→ forecast → refresh dashboard .
- Visualization: dash/app.py serves interactive web UI with filters and alerts.
- Docs: docs/Project_Documentation.md and architecture_diagram.png describe architecture and runbook.



DBT Model Lineage

- Staging Layer (stg_global_index.sql)
  - Loads and cleans raw CSV data.
  - Converts data types and removes nulls.
- Intermediate Layer (int_market_metrics.sql)
  - Calculates daily returns, log returns, and rolling volatility.
  - Prepares data for analytical use.
- Final Layer (fct_market_summary.sql)
  - Aggregates metrics across indices and computes average volatility & return.
  - Provides data for dashboards and forecasting.

Forecasting Logic & Outputs

The script model_forecast.py trains a time-series model on historical close prices.

It outputs:

Next 7-day price forecasts

Confidence intervals

Forecast accuracy metrics (MAE, RMSE)

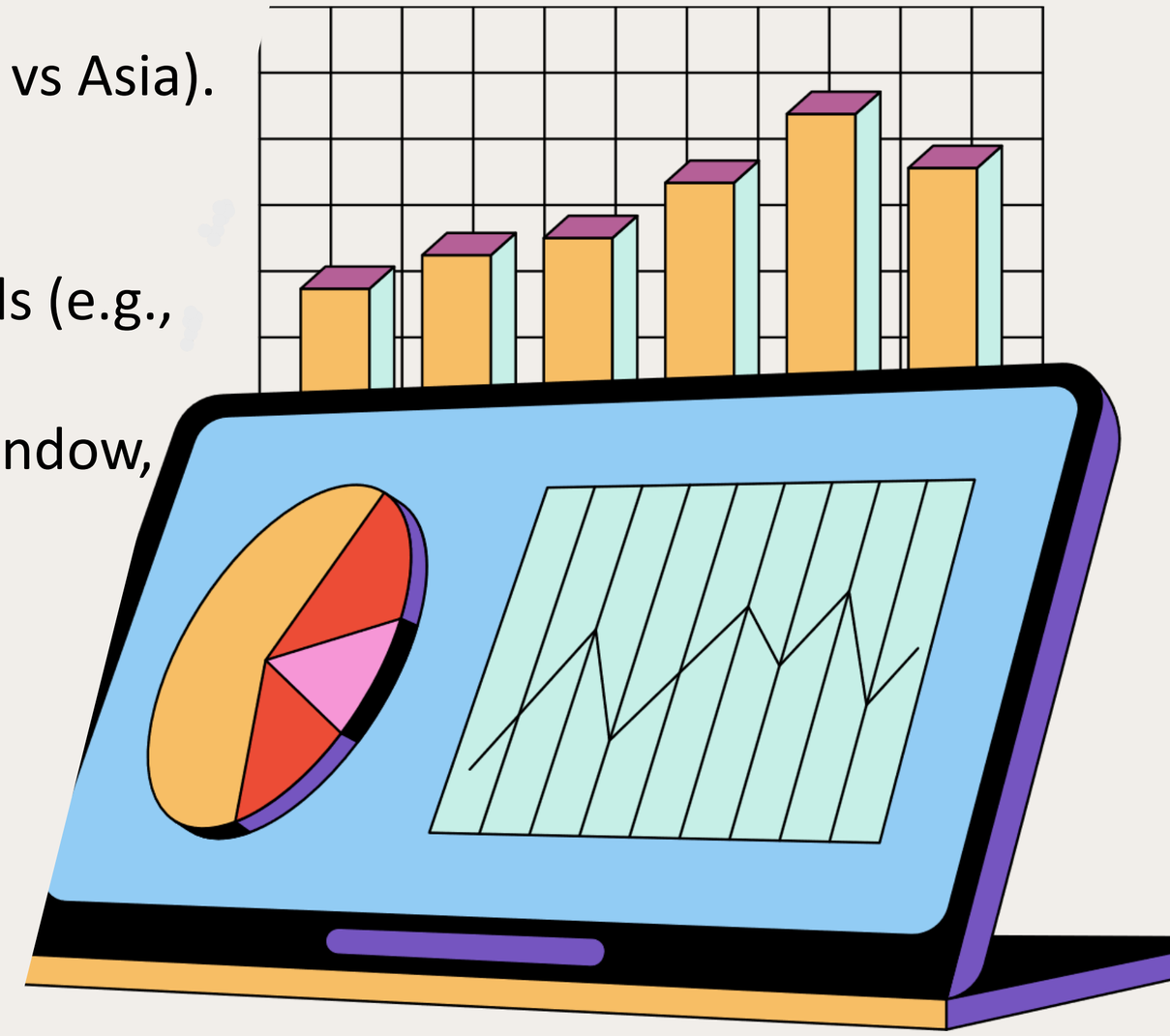
Airflow Orchestration

DAG: global-market-dag.py orchestrates:

- - ingest_data Runs (load_data.py)
- - run_dbt (executes dbt models)
- - dbt_test (Test dbt)
- - forecast (runs model_forecast.py)
- - refresh_dashboard (refreshes analytics layer / cache)

Dash Dashboard Features

- • Market index time series, returns, volatility.
- • Cross-market comparison (e.g., US vs Europe vs Asia).
- • Forecast plots vs actual [trends].
- • Risk metrics / volatility heatmaps.
- • Alerts or flags when indices exceed thresholds (e.g., large drops or spikes).
- • Support slicing by region, index type, time window, and volatility regimes.



Setup & Execution

1. Environment Setup
   - pip install -r requirements.txt
2. Database Setup
   - mysql -u admin -p < sql/create_tables.sql
   - python sql/load_data.py
3. Run dbt models
   - Cd dbt
   - source ~/venv/bin/activate
   - cd debug
   - dbt run
4. Run Forecasting
   - python forecasting/model_forecast.py
5. Launch Dashboard
   - python dash/app.py
6. Airflow DAG
   - airflow standalone