AutoStockML – Automated Stock Prediction with Machine Learning

By
Hitanshi Patil - 16010122283
Harikrishnan Gopal - 16010122284
Aditya Raut - 16010122288
Suhrud Korgaokar – 16010122334

Guide
Prof. Swapnil Pawar

Somaiya Vidyavihar University
Vidyavihar, Mumabi - 400 077
Academic Year 2025-2026

Introduction

In the era of algorithmic trading and financial automation, the ability to predict stock market movements has become one of the most challenging yet rewarding problems in data science.
However, stock data is dynamic, volatile, and time-sensitive, which means any static machine learning model rapidly loses accuracy as market behavior changes.

AutoStockML was conceptualized to overcome this limitation by creating a fully automated, self-improving stock prediction system. The project integrates the principles of Machine Learning (ML) and DevOps (MLOps) to enable a daily retraining and redeployment pipeline, ensuring that the prediction model always stays up to date with the latest market data.

This system operates autonomously, fetching fresh intraday data, retraining the predictive model, evaluating its performance, version-controlling the outputs, and redeploying it to production all handled by an intelligent GitHub Actions CI/CD pipeline.

Through this, AutoStockML demonstrates the concept of "self-sustaining AI", where models continuously evolve and redeploy themselves with minimal human oversight.

Objectives

The main objective of AutoStockML is to automate the complete lifecycle of a stock price prediction model.
Specific goals include:

1. Data Automation: Automatically fetch and store daily intraday market data after market close.
2. Model Retraining: Retrain the machine learning model using the newly fetched data.
3. Continuous Evaluation: Evaluate model performance metrics to ensure consistent improvement.
4. Version Control: Save and manage model versions and corresponding datasets for traceability.
5. Automated Deployment: Automatically deploy the latest model version as a REST API using Railway.
6. CI/CD Integration: Establish a GitHub Actions pipeline that orchestrates data ingestion, training, evaluation, and deployment without manual execution.

By combining these elements, the system ensures high accuracy, reliability, and operational efficiency for real-world financial prediction tasks.

System Overview

The AutoStockML architecture consists of multiple automated components working in synchronization.
The workflow begins with a GitHub Actions trigger that fetches new stock market data after market close. This data is cleaned and appended to the historical dataset.
Next, a machine learning model (LSTM) is trained on the updated data and evaluated for performance.

Once trained, the new model is saved, versioned, and committed back to the repository.
This triggers Railway, which automatically redeploys the updated Flask API, making the new model instantly available for live predictions.

The system thus forms a closed-loop automation cycle:

Data Fetch → Preprocess → Train → Evaluate → Commit → Deploy → Predict

This pipeline embodies the essence of Continuous Integration (CI), Continuous Delivery (CD), and Continuous Training (CT) in modern MLOps systems.

Workflow Architecture

Automated Trigger (GitHub Actions)

- Workflow scheduled daily at 11:30 UTC via CRON job or manual trigger

Data Fetching

- Connects to stock data APIs (Alpha Vantage).
- Retrieves the latest intraday OHLCV (Open, High, Low, Close, Volume) data.
- Cleans, validates, and saves data as CSV files.

Data Source (Stock API)

- Acts as the external data provider feeding live market information to the system.

Model Training

- Merges new intraday data with historical records.
- Retrains the ML model (LSTM).
- Evaluates accuracy using RMSE, MAE, R^2 Score.
- Saves updated model files for versioning.

GitHub Repository (Main Branch)

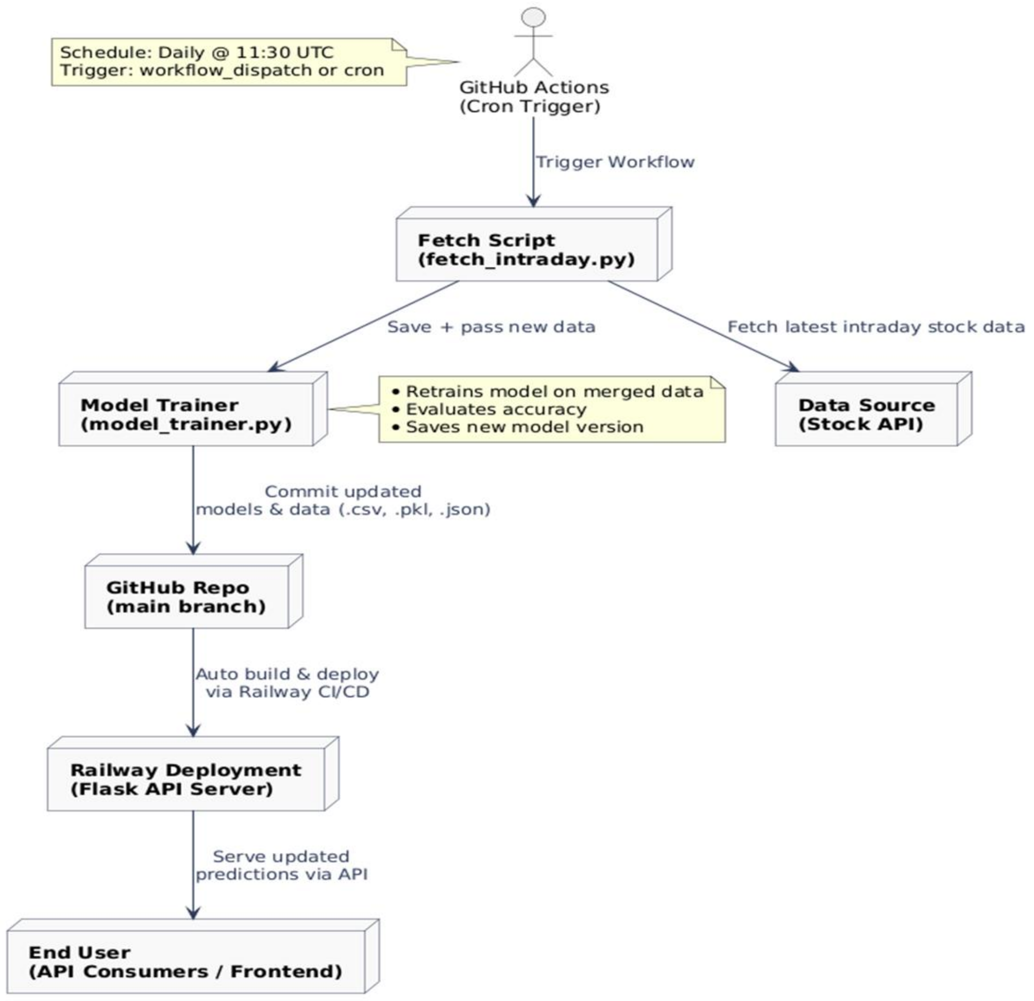
- Automatically commits and pushes updated datasets and model files.
- Maintains full version control and traceability for each retrain cycle.

Deployment via Railway CI/CD

- Railway detects the new commit and rebuilds the container.
- Flask API is redeployed automatically with the new model.
- Ensures zero-downtime updates and continuous availability.

End User

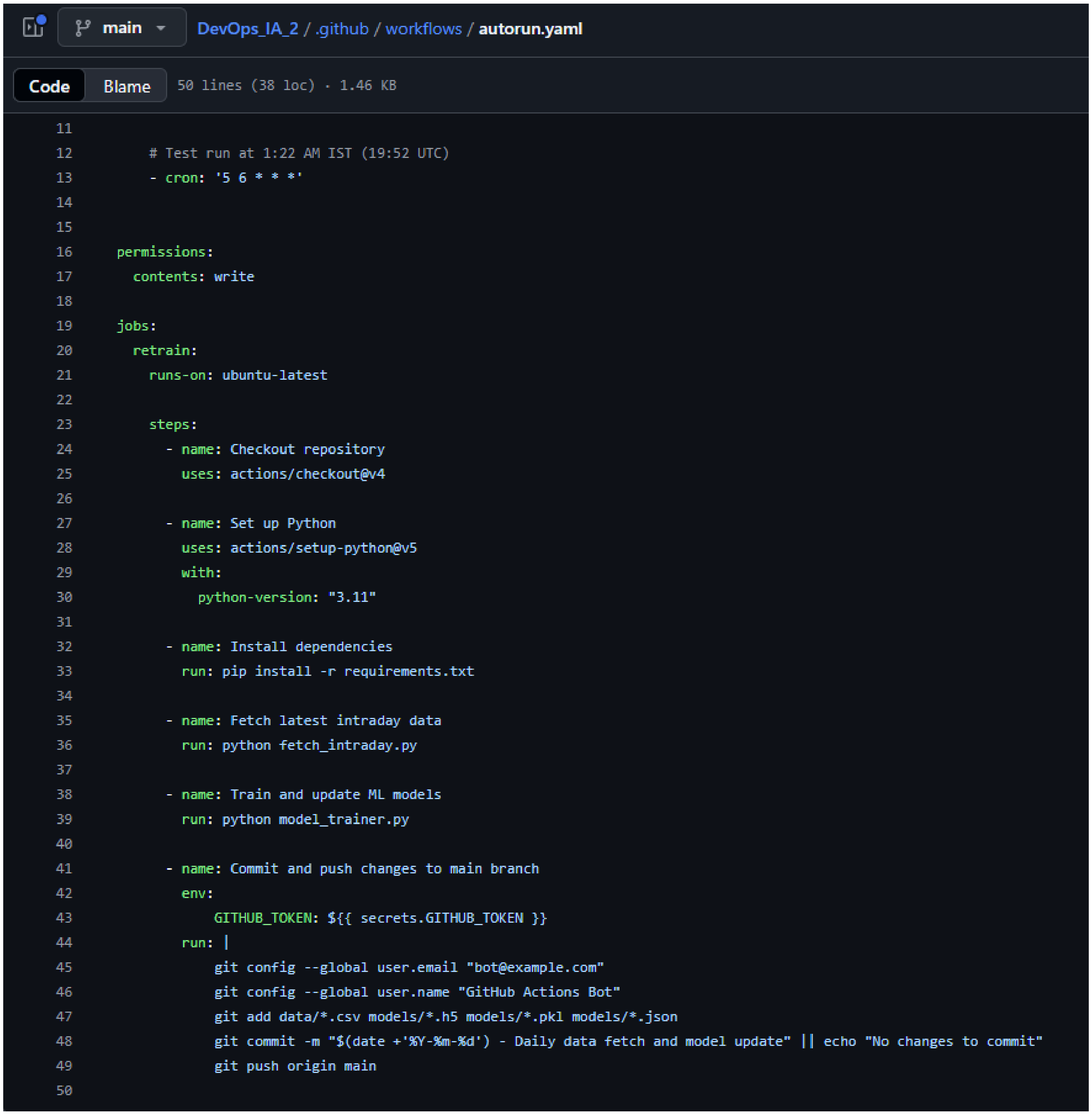
- Users (dashboards or frontend apps) query the live API endpoint for predictions.
- Always receive latest, retrained model outputs.



Implementation

Github: https://github.com/amNobodyyy/DevOps_IA_2

The workflow file:

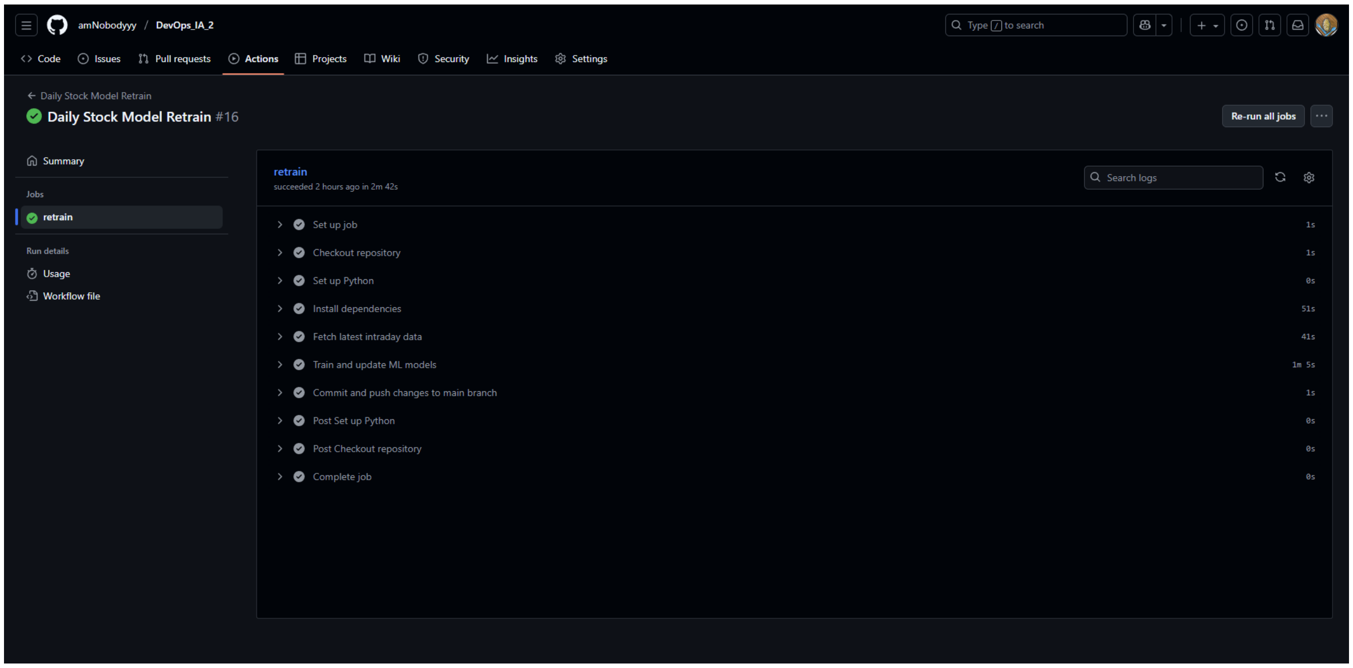


main DevOps_IA_2 / .github / workflows / autorun.yaml

Code Blame 50 lines (38 loc) · 1.46 KB

```yaml

    # Test run at 1:22 AM IST (19:52 UTC)
    - cron: '5 6 * * *'


permissions:
  contents: write

jobs:
  retrain:
    runs-on: ubuntu-latest

    steps:
      - name: Checkout repository
        uses: actions/checkout@v4

      - name: Set up Python
        uses: actions/setup-python@v5
        with:
          python-version: "3.11"

      - name: Install dependencies
        run: pip install -r requirements.txt

      - name: Fetch latest intraday data
        run: python fetch_intraday.py

      - name: Train and update ML models
        run: python model_trainer.py

      - name: Commit and push changes to main branch
        env:
            GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
        run: |
            git config --global user.email "bot@example.com"
            git config --global user.name "GitHub Actions Bot"
            git add data/*.csv models/*.h5 models/*.pkl models/*.json
            git commit -m "$(date +'%Y-%m-%d') - Daily data fetch and model update" || echo "No changes to commit"
            git push origin main

```

Workflow execution:



amNobodyyy / DevOps_IA_2

Code | Issues | Pull requests | Actions | Projects | Wiki | Security | Insights | Settings

← Daily Stock Model Retrain

Daily Stock Model Retrain #16

Re-run all jobs

Summary

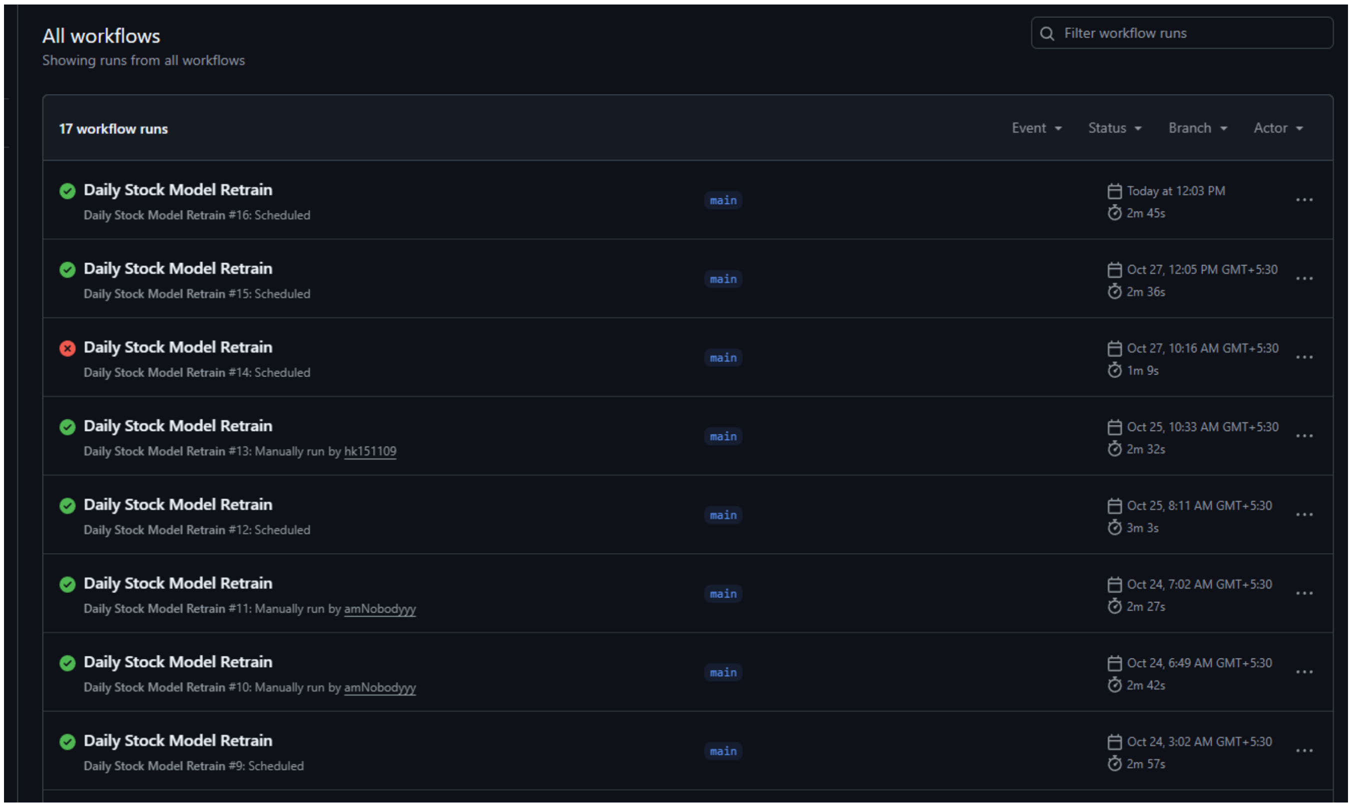
Jobs
- retrain

Run details
- Usage
- Workflow file

retrain
succeeded 2 hours ago in 2m 42s

```
Set up job                                   1s
Checkout repository                          1s
Set up Python                                0s
Install dependencies                        51s
Fetch latest intraday data                  41s
Train and update ML models               1m 5s
Commit and push changes to main branch       1s
Post Set up Python                           0s
Post Checkout repository                     0s
Complete job                                 0s
```

All workflow list:



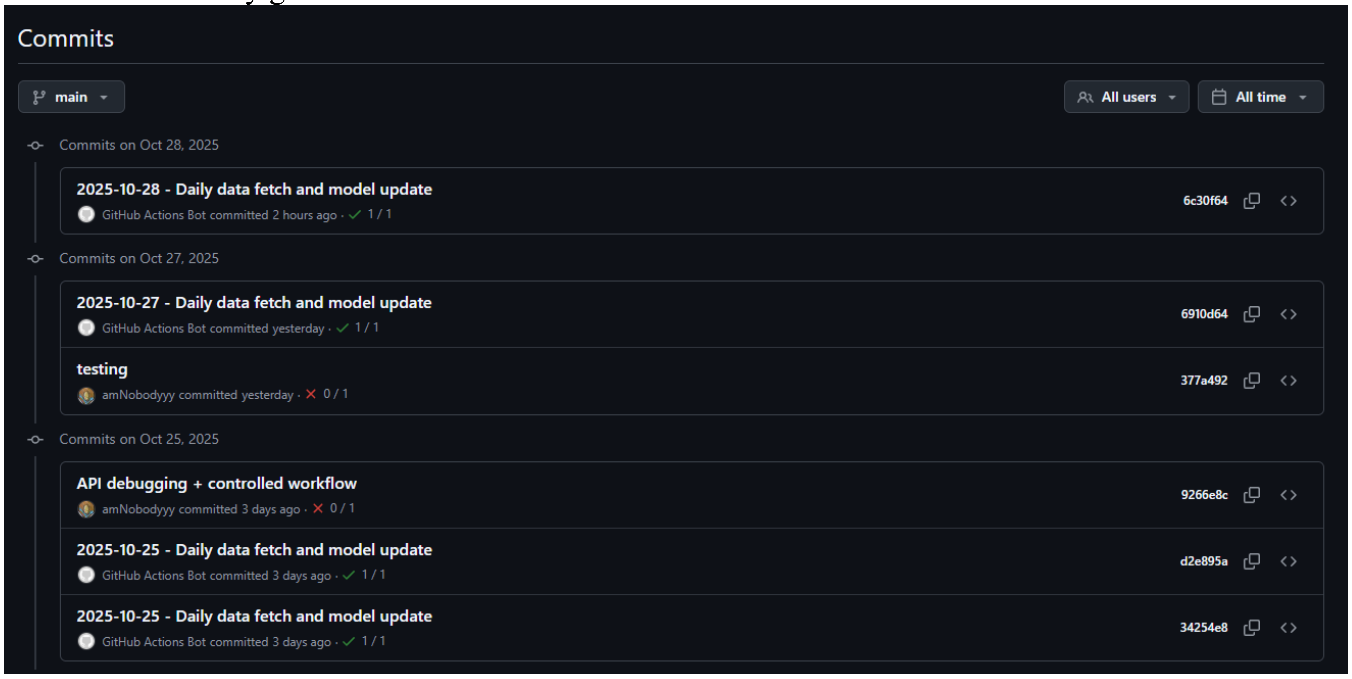
All workflows
Showing runs from all workflows

Filter workflow runs

17 workflow runs — Event | Status | Branch | Actor

Run	Branch	Date / Duration
✓ Daily Stock Model Retrain Daily Stock Model Retrain #16: Scheduled	main	Today at 12:03 PM 2m 45s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #15: Scheduled	main	Oct 27, 12:05 PM GMT+5:30 2m 36s
✗ Daily Stock Model Retrain Daily Stock Model Retrain #14: Scheduled	main	Oct 27, 10:16 AM GMT+5:30 1m 9s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #13: Manually run by hk151109	main	Oct 25, 10:33 AM GMT+5:30 2m 32s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #12: Scheduled	main	Oct 25, 8:11 AM GMT+5:30 3m 3s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #11: Manually run by amNobodyyy	main	Oct 24, 7:02 AM GMT+5:30 2m 27s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #10: Manually run by amNobodyyy	main	Oct 24, 6:49 AM GMT+5:30 2m 42s
✓ Daily Stock Model Retrain Daily Stock Model Retrain #9: Scheduled	main	Oct 24, 3:02 AM GMT+5:30 2m 57s

Commits created by github actions:



Commits

main · All users · All time

Commits on Oct 28, 2025

2025-10-28 - Daily data fetch and model update — 6c30f64
GitHub Actions Bot committed 2 hours ago · ✓ 1 / 1

Commits on Oct 27, 2025

2025-10-27 - Daily data fetch and model update — 6910d64
GitHub Actions Bot committed yesterday · ✓ 1 / 1

testing — 377a492
amNobodyyy committed yesterday · ✗ 0 / 1

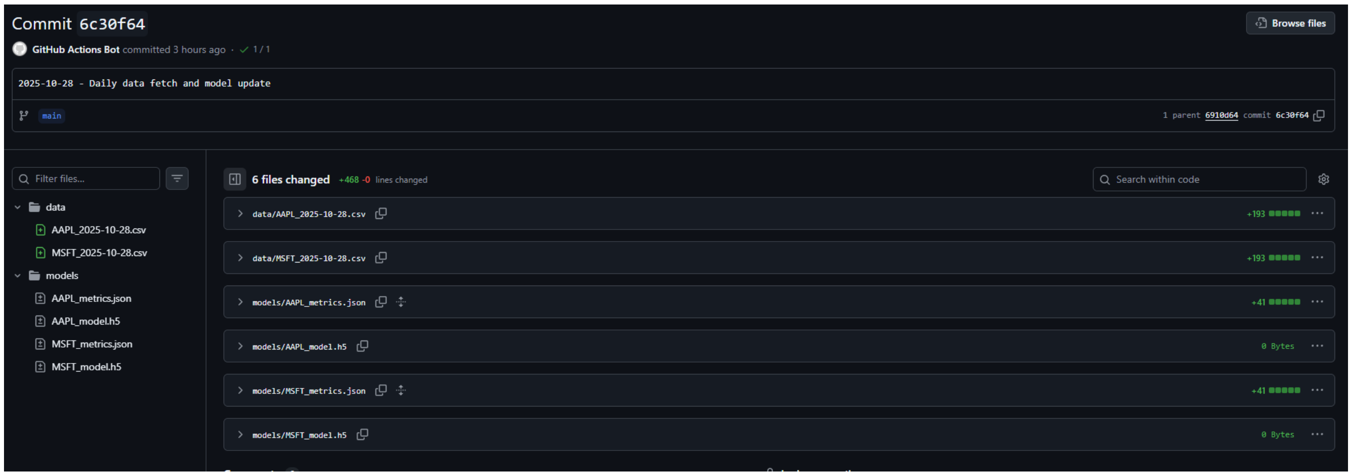
Commits on Oct 25, 2025

API debugging + controlled workflow — 9266e8c
amNobodyyy committed 3 days ago · ✗ 0 / 1

2025-10-25 - Daily data fetch and model update — d2e895a
GitHub Actions Bot committed 3 days ago · ✓ 1 / 1

2025-10-25 - Daily data fetch and model update — 34254e8
GitHub Actions Bot committed 3 days ago · ✓ 1 / 1

Commit created:



Commit 6c30f64 — Browse files

GitHub Actions Bot committed 3 hours ago · ✓ 1 / 1

```
2025-10-28 - Daily data fetch and model update
```

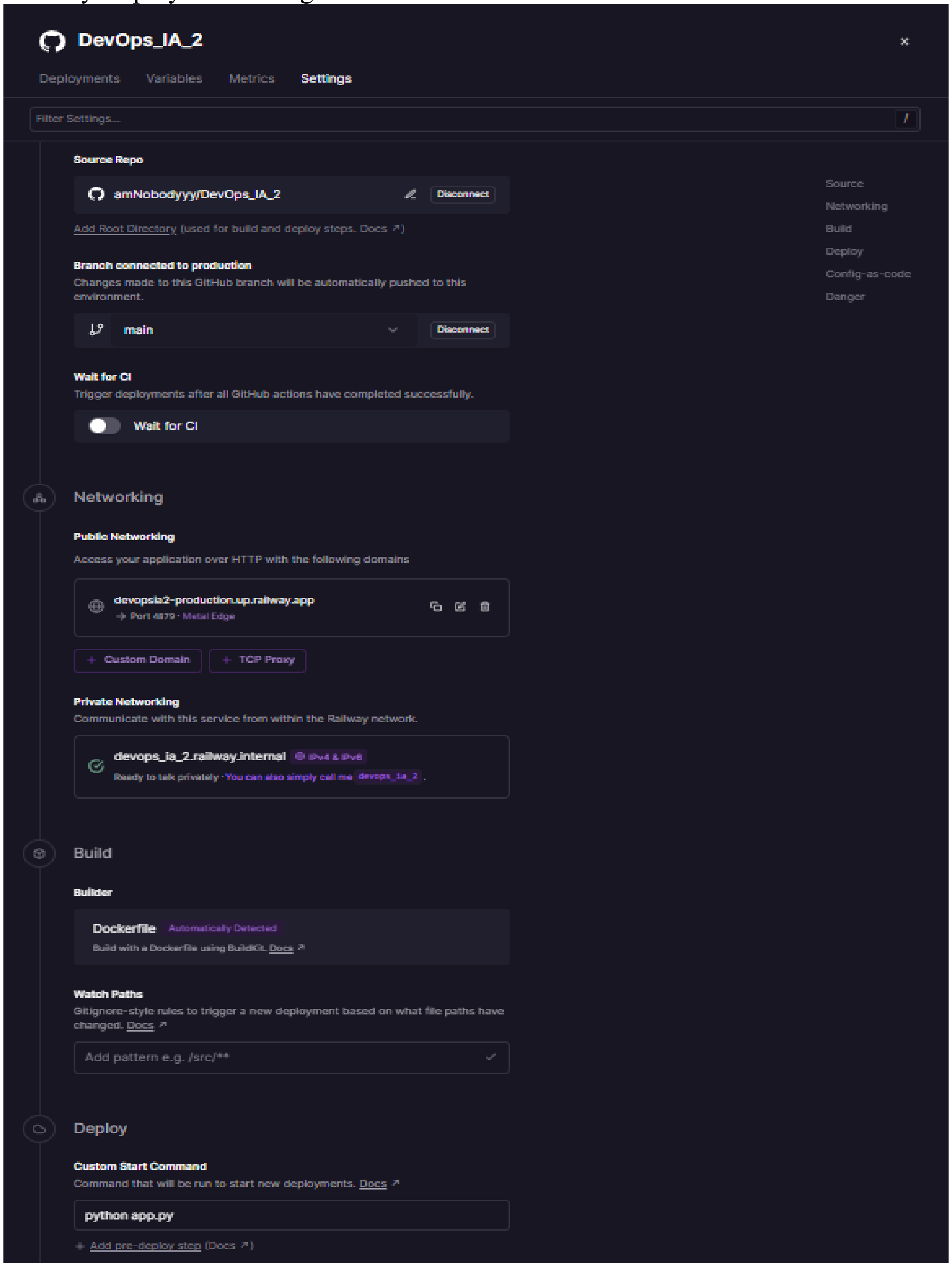
main — 1 parent 6910d64 commit 6c30f64

6 files changed +468 -0 lines changed

- data
  - AAPL_2025-10-28.csv
  - MSFT_2025-10-28.csv
- models
  - AAPL_metrics.json
  - AAPL_model.h5
  - MSFT_metrics.json
  - MSFT_model.h5

data/AAPL_2025-10-28.csv	+193
data/MSFT_2025-10-28.csv	+193
models/AAPL_metrics.json	+41
models/AAPL_model.h5	0 Bytes
models/MSFT_metrics.json	+41
models/MSFT_model.h5	0 Bytes

Railway Deployment configuration:



Source Repo

amNobodyyy/DevOps_IA_2 — Disconnect

Add Root Directory (used for build and deploy steps. Docs ↗)

Branch connected to production

Changes made to this GitHub branch will be automatically pushed to this environment.

main — Disconnect

Wait for CI

Trigger deployments after all GitHub actions have completed successfully.

Wait for CI

Networking

Public Networking

Access your application over HTTP with the following domains

devopsia2-production.up.railway.app
→ Port 4879 · Metal Edge

+ Custom Domain   + TCP Proxy

Private Networking

Communicate with this service from within the Railway network.

devops_ia_2.railway.internal   IPv4 & IPv6
Ready to talk privately · You can also simply call me devops_ia_2 .

Build

Builder

Dockerfile   Automatically Detected
Build with a Dockerfile using BuildKit. Docs ↗

Watch Paths

Gitignore-style rules to trigger a new deployment based on what file paths have changed. Docs ↗

Add pattern e.g. /src/**

Deploy

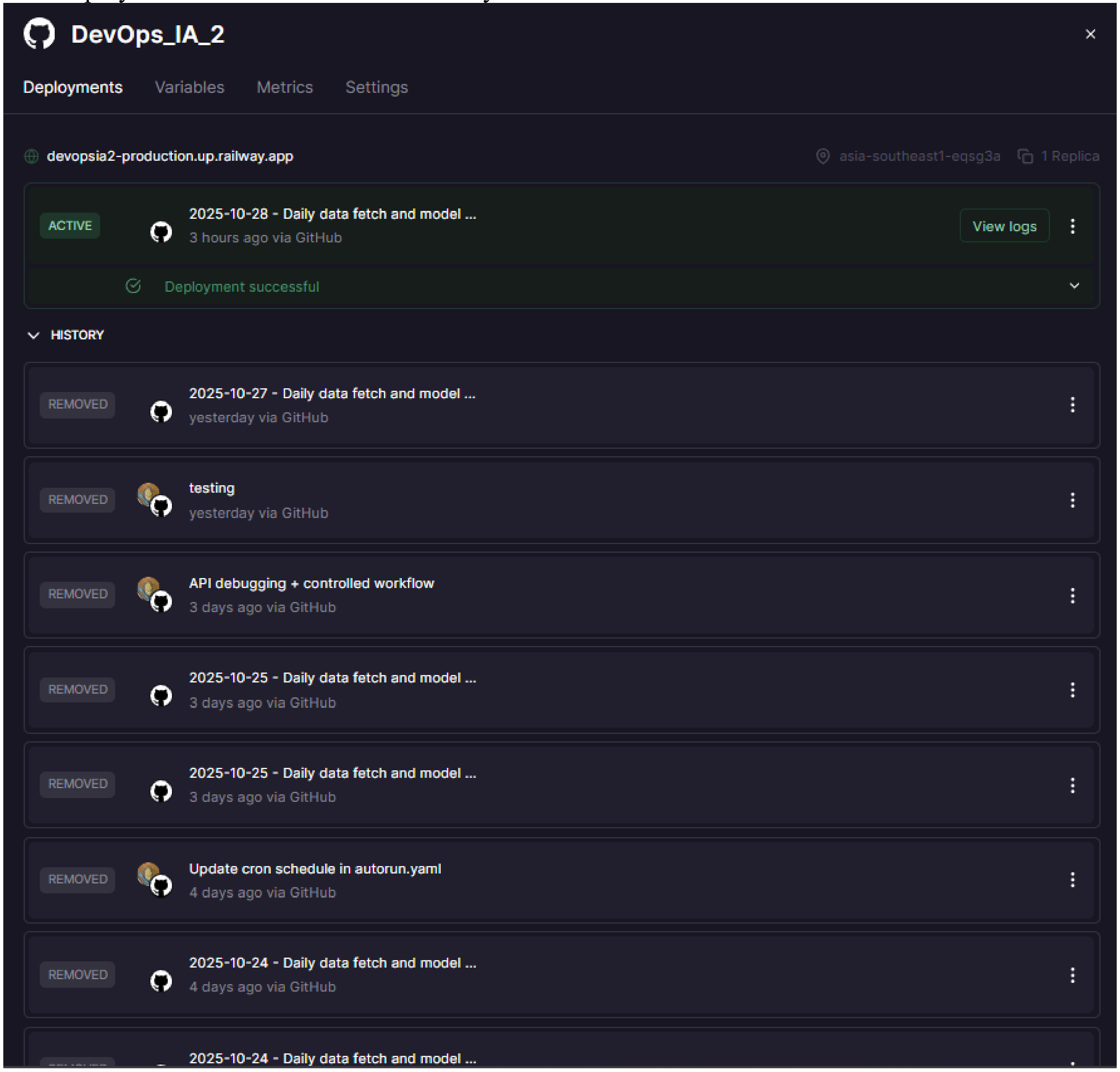
Custom Start Command

Command that will be run to start new deployments. Docs ↗

```
python app.py
```

+ Add pre-deploy step (Docs ↗)

New deployment after each commit in railway:



DevOps_IA_2

devopsia2-production.up.railway.app

asia-southeast1-eqsg3a   1 Replica

ACTIVE   2025-10-28 - Daily data fetch and model ...
3 hours ago via GitHub   View logs

Deployment successful

HISTORY

REMOVED   2025-10-27 - Daily data fetch and model ...
yesterday via GitHub

REMOVED   testing
yesterday via GitHub

REMOVED   API debugging + controlled workflow
3 days ago via GitHub

REMOVED   2025-10-25 - Daily data fetch and model ...
3 days ago via GitHub

REMOVED   2025-10-25 - Daily data fetch and model ...
3 days ago via GitHub

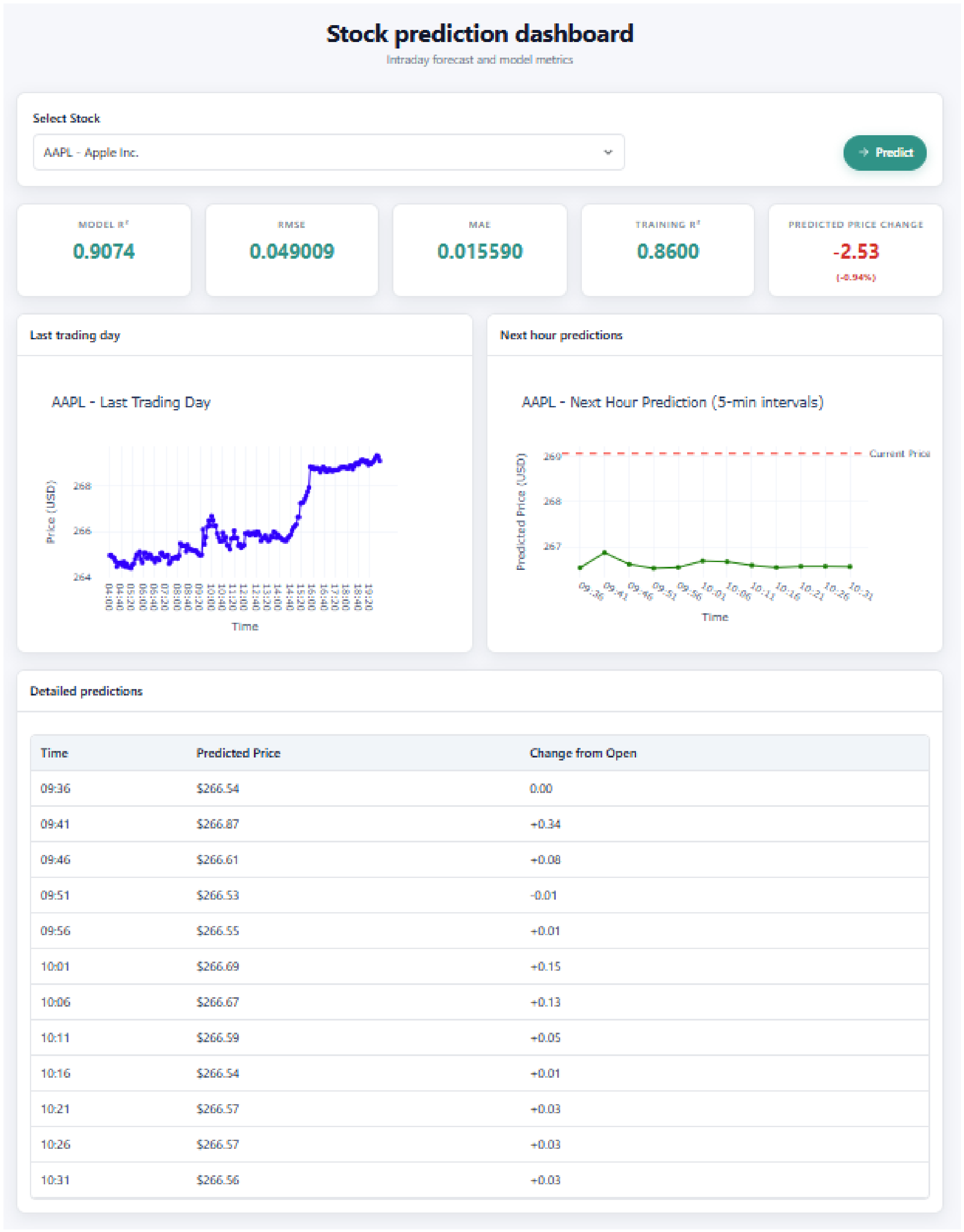
REMOVED   Update cron schedule in autorun.yaml
4 days ago via GitHub

REMOVED   2025-10-24 - Daily data fetch and model ...
4 days ago via GitHub

2025-10-24 - Daily data fetch and model ...

Live deployment:

https://devopsia2-production.up.railway.app/



Stock prediction dashboard

Intraday forecast and model metrics

Select Stock: AAPL - Apple Inc.

Predict

MODEL R²	RMSE	MAE	TRAINING R²	PREDICTED PRICE CHANGE
0.9074	0.049009	0.015590	0.8600	-2.53 (-0.94%)

Last trading day

Next hour predictions

Detailed predictions

Time	Predicted Price	Change from Open
09:36	$266.54	0.00
09:41	$266.87	+0.34
09:46	$266.61	+0.08
09:51	$266.53	-0.01
09:56	$266.55	+0.01
10:01	$266.69	+0.15
10:06	$266.67	+0.13
10:11	$266.59	+0.05
10:16	$266.54	+0.01
10:21	$266.57	+0.03
10:26	$266.57	+0.03
10:31	$266.56	+0.03

Results and Observations

- Daily Automated Retraining:
  The model retrains daily using fresh market data, staying synchronized with current market conditions.
- Adaptive Accuracy:
  Each retrain cycle improves the model's ability to capture new trading behaviors, leading to progressively better predictions.
- Zero Manual Effort:
  The entire system runs autonomously data fetching, training, versioning, and deployment require no human input.
- Seamless Integration:
  The CI/CD pipeline (GitHub Actions + Railway) ensures end-to-end automation, from ingestion to live API updates.
- Reliability:
  The system maintains consistent uptime, minimal latency, and fully traceable model versions — essential traits for real-world MLOps applications.

Future Enhancements

- Incorporate real-time streaming data for continuous intraday updates.
- Implement Docker & Kubernetes for horizontal scalability.
- Integrate MLflow or Weights & Biases (W&B) for experiment tracking and model comparisons. Add notification systems (Slack/email) to report retraining status and performance metrics.
- Extend predictions to multi-stock portfolios and sector-wise forecasting.
- Implement Reinforcement Learning models to simulate decision-making and trading strategies.

Conclusion

AutoStockML successfully delivers a self-evolving MLOps pipeline capable of autonomously managing the complete machine learning lifecycle.
By combining data automation, continuous training, and automatic deployment, the system achieves both operational efficiency and technical robustness.

This project proves that Machine Learning and DevOps can work synergistically to build intelligent, production-ready systems that learn, adapt, and deploy themselves with no human intervention.

In essence, AutoStockML is a scalable blueprint for future financial AI platforms, bridging the gap between data science automation and real-time market intelligence.