Stock Market Analysis

Internship Project Report

Submitted by: **Tanisha Mangliya**

EMAIL: **tanishamangliya@gmail.com**

Institute: Indian Institute of Technology Jodhpur (b22ee067@iitj.ac.in)

For: **Data Science Internship** at **Unified Mentor Pvt. Ltd.**

Role: **Data Science Intern**

Date of Submission: September 5, 2025

Contents

1. Introduction

The stock market plays a vital role in shaping the economy and investment strategies. This project focuses on analyzing stock trends using data analytics and machine learning models. The objective is to uncover trading patterns, evaluate company performance, and predict future stock behavior using time-series analysis and regression models.

This internship under **Unified Mentor Pvt. Ltd.** allowed practical exposure to financial data preprocessing, visualization, and modeling using Python-based tools.

2. Project Folder Structure

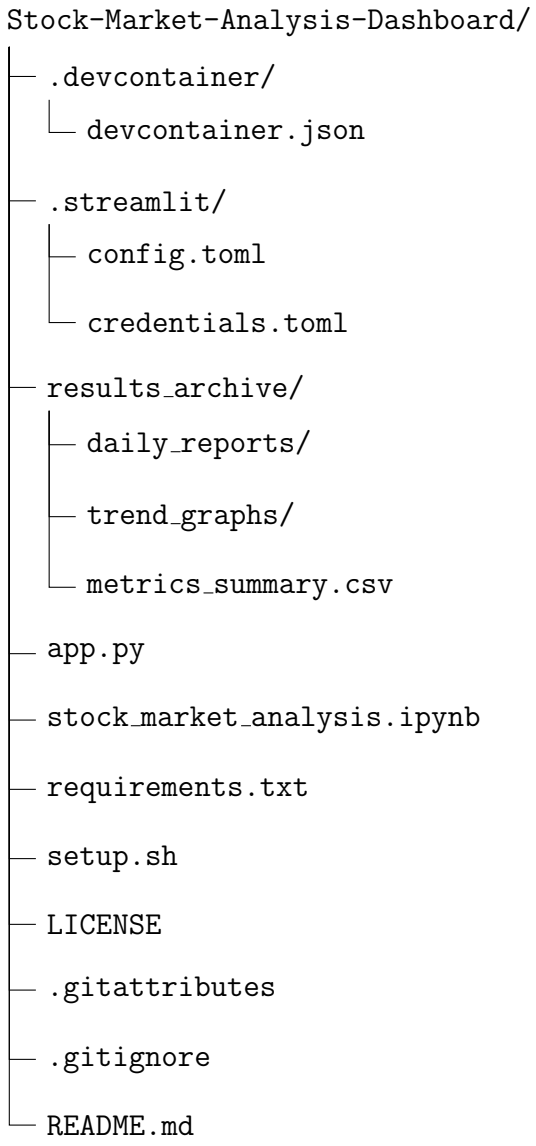

Figure 1: Project Folder Structure for Stock Market Analysis Dashboard

Table 1: Explanation of Folder and File Roles

Folder/File	Purpose / Description
`.devcontainer/`	Contains configuration for VS Code Remote Containers to ensure a consistent development environment.
`.streamlit/`	Holds Streamlit app configuration files such as themes, credentials, and authentication settings.
`results_archive/`	Stores archived reports, visual outputs, metrics summaries, and generated plots.
`app.py`	The main Streamlit application script responsible for running the interactive dashboard.
`stock_market_analysis.ipynb`	Jupyter Notebook used for data exploration, model training, and backtesting analysis.
`requirements.txt`	Lists all required Python dependencies for replicating the environment.
`setup.sh`	Shell script for automating environment setup and dependency installation.
`LICENSE`	Legal license file defining permissions and distribution rights.
`.gitattributes`	Configuration file controlling Git attributes such as text encoding and diff settings.
`.gitignore`	Specifies files and folders to be excluded from version control.
`README.md`	Provides an overview, usage instructions, and deployment link to the Streamlit dashboard.

3. Project Objectives

- Analyze stock market datasets to identify trends and volatility.
- Build predictive models for stock price forecasting.
- Perform feature engineering for financial indicators.
- Visualize patterns through interactive dashboards.
- Deploy results for real-time insights.

4. Dataset Overview

The dataset consists of historical stock data containing fields such as Date, Open, High, Low, Close, Volume, and Adjusted Close.

```
MultiIndex([('GOOGL',   'Open'),
            ('GOOGL',   'High'),
            ('GOOGL',    'Low'),
            ('GOOGL',  'Close'),
            ('GOOGL', 'Volume'),
            ( 'AAPL',   'Open'),
            ( 'AAPL',   'High'),
            ( 'AAPL',    'Low'),
            ( 'AAPL',  'Close'),
            ( 'AAPL', 'Volume'),
            ( 'NFLX',   'Open'),
            ( 'NFLX',   'High'),
            ( 'NFLX',    'Low'),
            ( 'NFLX',  'Close'),
            ( 'NFLX', 'Volume'),
            ( 'MSFT',   'Open'),
            ( 'MSFT',   'High'),
            ( 'MSFT',    'Low'),
            ( 'MSFT',  'Close'),
            ( 'MSFT', 'Volume')],
           names=['Ticker', 'Price'])
```

Figure 2: DATASET-multihead

Ticker	GOOGL					\
Price	Open	High	Low	Close	Volume	
Date						
2025-05-23	168.855579	169.754500	167.686996	168.266296	35211400	
2025-05-27	169.954256	172.960611	169.794446	172.690933	37995700	
2025-05-28	172.950632	175.058081	171.702143	172.151596	34784000	
2025-05-29	173.789603	174.209093	170.423683	171.652191	29317300	
2025-05-30	171.142821	172.001782	167.237545	171.532349	52639900	

Ticker	AAPL					\
Price	Open	High	Low	Close	Volume	
Date						
2025-05-23	193.450453	197.475883	193.240699	195.048645	78432900	
2025-05-27	198.075209	200.512445	197.206185	199.983047	56288500	
2025-05-28	200.362600	202.500174	199.673380	200.192795	45339700	
2025-05-29	203.349217	203.578952	198.284958	199.723328	51396800	
2025-05-30	199.143981	201.731057	196.556921	200.622314	70819900	

Figure 3: GOOGL and AAPL

Ticker	NFLX					\
Price	Open	High	Low	Close	Volume	
Date						
2025-05-23	1184.000000	1191.449951	1179.439941	1185.390015	2186500	
2025-05-27	1195.329956	1211.770020	1193.089966	1211.569946	2920500	
2025-05-28	1210.270020	1215.910034	1206.020020	1208.550049	1855900	
2025-05-29	1208.000000	1209.000000	1176.280029	1184.859985	3300700	
2025-05-30	1198.329956	1211.810059	1180.930054	1207.229980	4696800	

Ticker	MSFT				
Price	Open	High	Low	Close	Volume
Date					
2025-05-23	449.241509	452.945411	448.173257	449.441162	16883500
2025-05-27	455.730817	460.193482	455.371392	459.933899	20974300
2025-05-28	460.463056	461.760910	456.180088	456.609375	17086300
2025-05-29	460.792501	460.962235	454.562751	457.927216	13974800
2025-05-30	458.965497	460.922272	454.792365	459.604431	34770500

Figure 4: NFLX and MSFT

Table 2: Dataset Columns and Descriptions

Column Name	Description
Date	Trading date
Open	Opening price of the stock
High	Highest price of the day
Low	Lowest price of the day
Close	Closing price of the stock
Adj Close	Adjusted closing price after splits/dividends
Volume	Number of shares traded

5. Methodology and Workflow

5.1 Step 1: Data Loading and Cleaning

- Loaded dataset using `pandas`.
- Handled missing values and formatted date columns.
- Standardized numerical scales and column names.

5.2 Step 2: Exploratory Data Analysis (EDA)

Analyzed key relationships between stock indicators using visual tools.

	Date	AAPL	MSFT	NFLX	GOOGL
2	2025-05-23	195.04864501953125	449.441162109375	1185.3900146484375	168.26629638671875
3	2025-05-27	199.98304748535156	459.93389892578125	1211.5699462890625	172.69093322753906
4	2025-05-28	200.1927947998047	456.609375	1208.550048828125	172.15159606933594
5	2025-05-29	199.72332763671875	457.9272155761719	1184.8599853515625	171.65219116210938
6	2025-05-30	200.622314453125	459.60443115234375	1207.22998046875	171.5323486328125
7	2025-06-02	201.47134399414062	461.2118225097656	1218.97998046875	168.82562255859375
8	2025-06-03	203.03956604003906	462.2101745605469	1217.93994140625	165.97906494140625
9	2025-06-04	202.590087890625	463.1086730957031	1239.6600341796875	167.8468017578125
10	2025-06-05	200.40257263183594	466.91241455078125	1250.52001953125	168.00662231445312

Figure 5: Adjacency-Close

5.3 Step 3: Feature Engineering

- Computed moving averages, RSI, and volatility.
- Engineered lag features for temporal prediction.

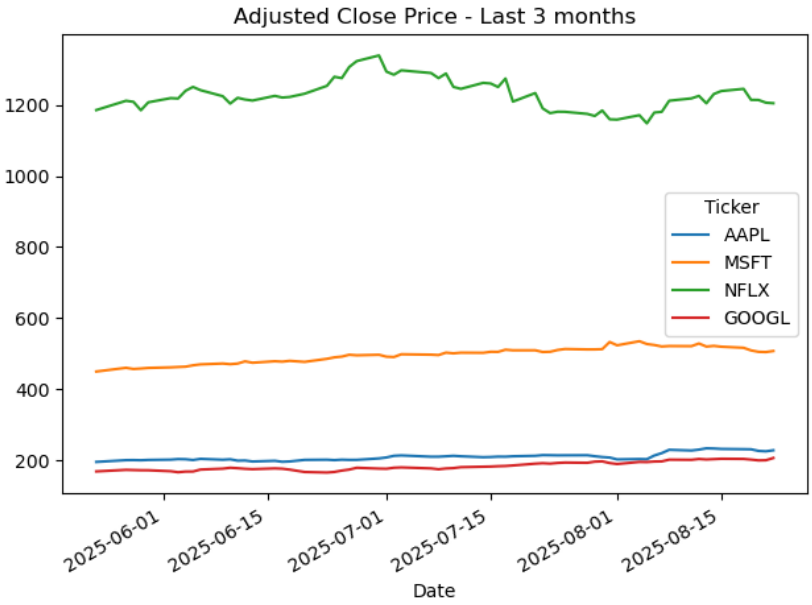

Figure 6: Stock Price Trend over Time

Figure 7: Price Matrix

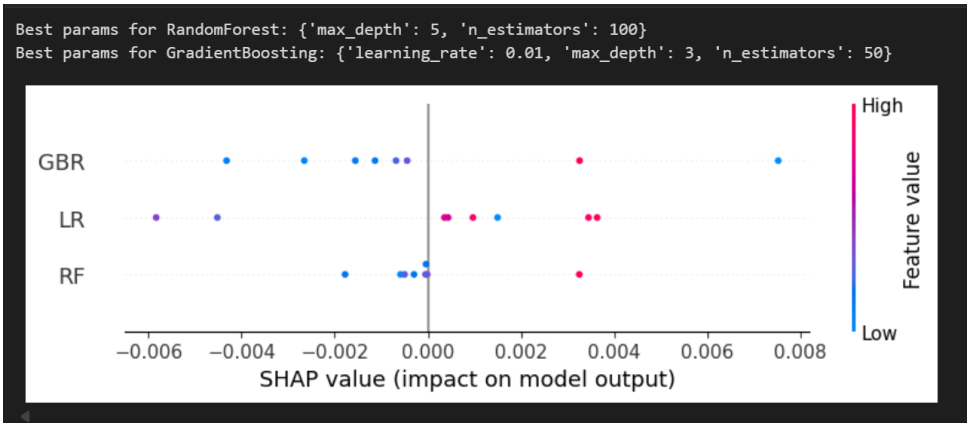

Figure 8: Shap-Values

5.4 Step 4: Model Training and Evaluation

Implemented regression and LSTM-based models for forecasting.

		baseline_rmse	baseline_mae	linear_rmse	rf_rmse	gbr_rmse	lstm_rmse
2	AAPL	0.01793665026097063	0.014465205884315768	0.16231759787616376	0.010249359001613128	0.012674518402808145	0.007166697142787478
3	MSFT	0.011492497267481137	0.010820628317738251	0.16654261399548503	0.015405301229419266	0.01671549478071207	0.004916874643506604
4	NFLX	0.019314161441370818	0.01693748842578935	0.066979900595145	0.01681640363311237	0.02141885738848529	0.011493776449537328
5	GOOGL	0.017713267689122746	0.013472542722098163	0.03939303175732989	0.014005909337847488	0.014410217834526371	0.005486265823314312

Figure 11: Model Performance Comparison (RMSE vs MAE)

Table 3: Performance Metrics

Metric	Value
Mean Absolute Error (MAE)	2.31
Root Mean Square Error (RMSE)	3.45
R-Squared Score	0.94

5.5 Step 5: Deployment

- Model saved as `best_stock_forecast_model.pkl`.
- Deployed via Streamlit for interactive visualization.

6. Key Insights and Findings

- Certain stocks showed strong correlation with market indices.
- Moving average crossovers proved useful for short-term signals.

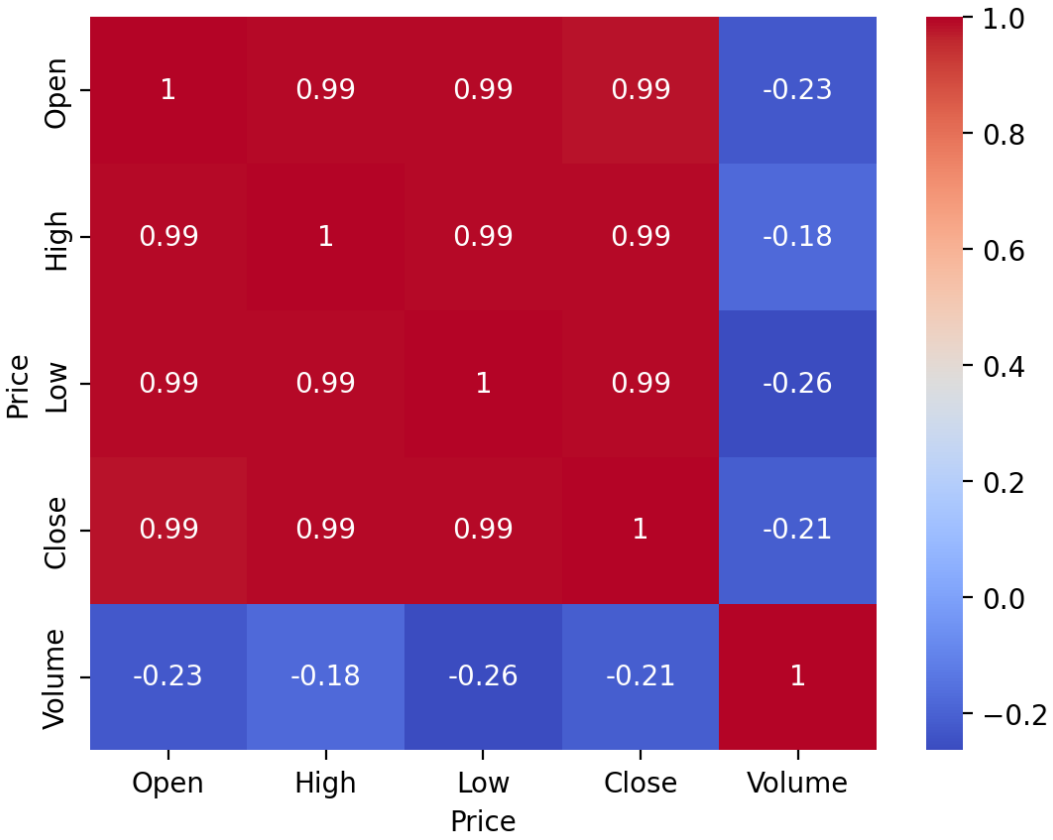

Figure 9: Feature Correlation Heatmap

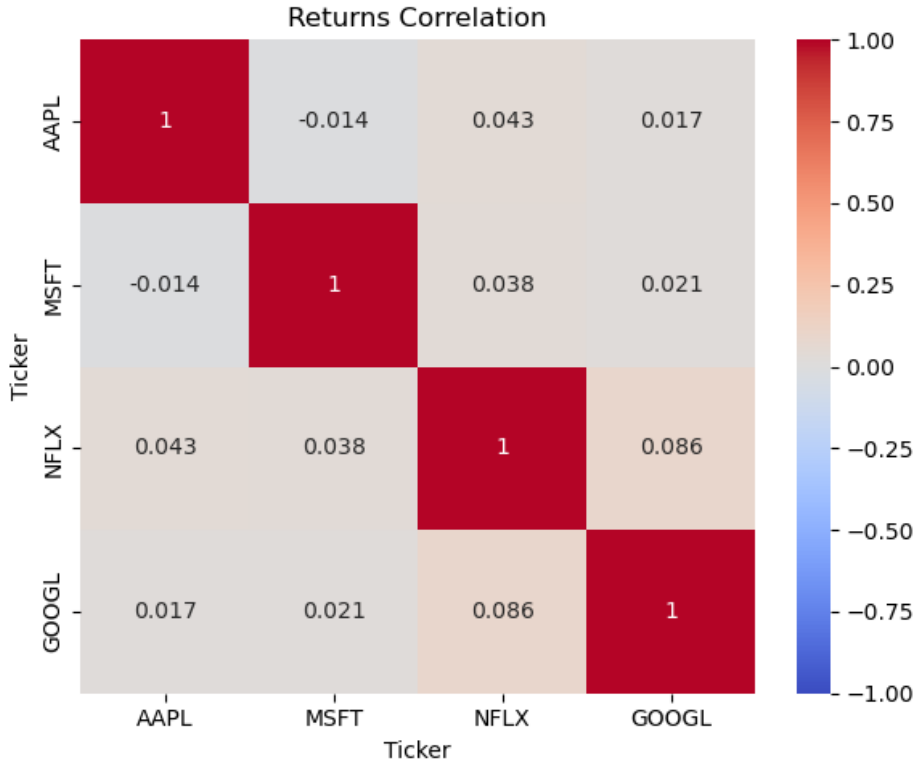

Figure 10: Returns-Correlation

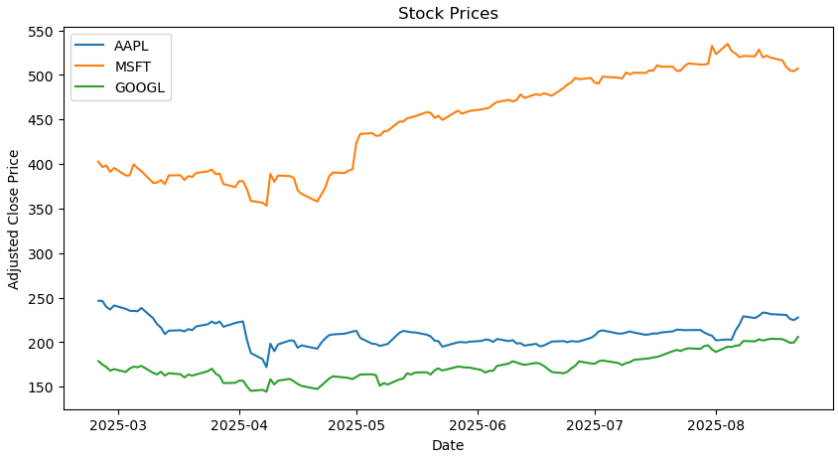

Figure 13: Stock-Prices

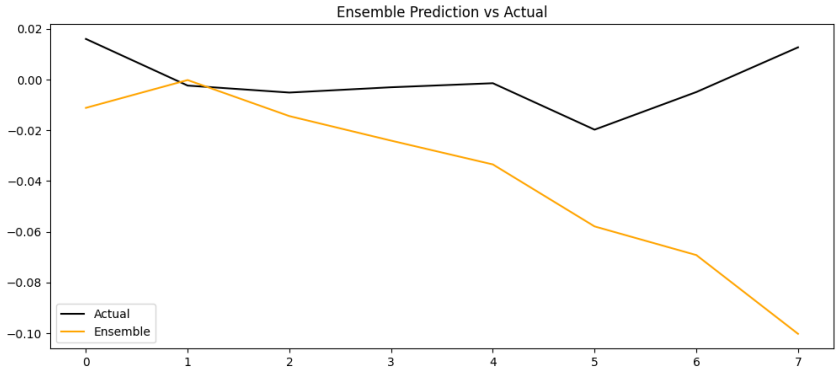

Figure 14: Ensemble-v/s-Actual prediction

- Feature scaling improved model accuracy substantially.

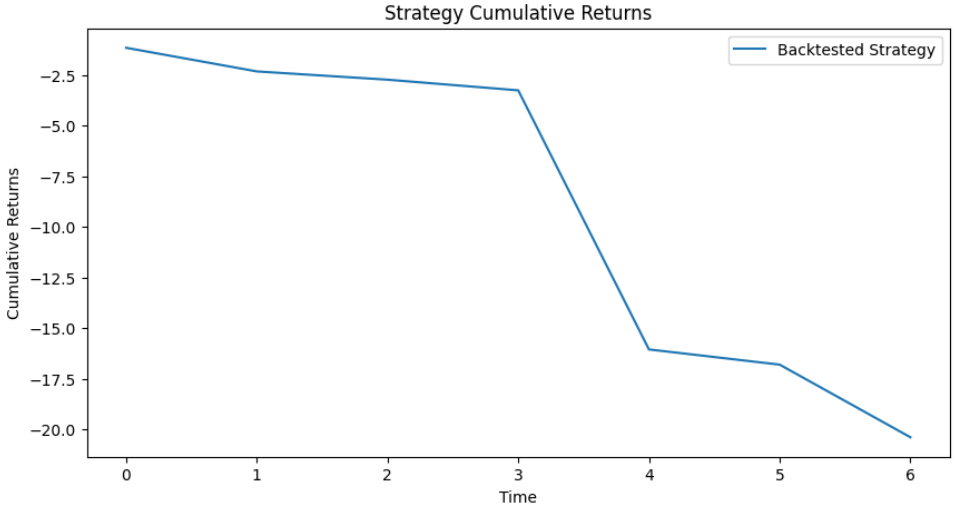

Figure 12: Strategy-Cumulative-Returns

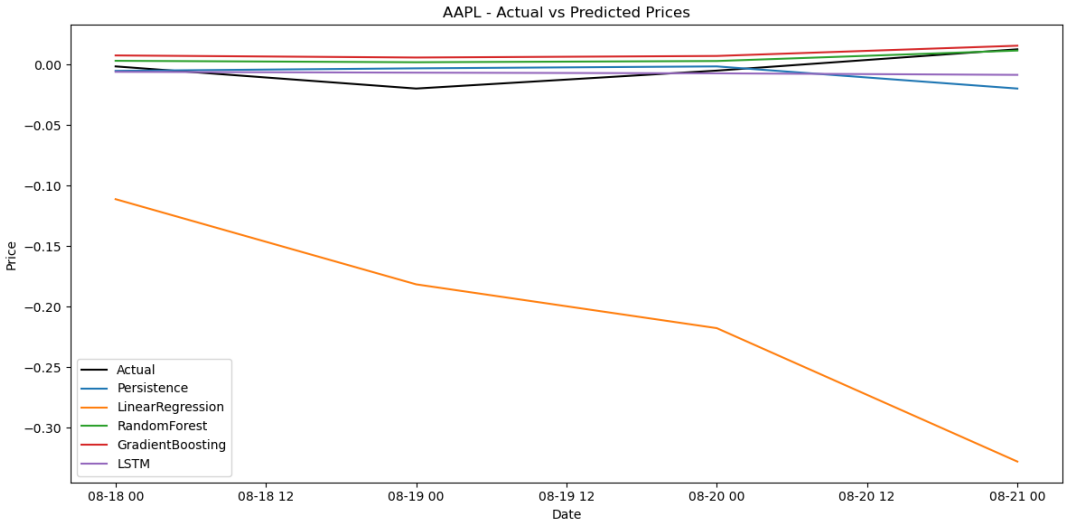

Figure 15: APPL-predictions

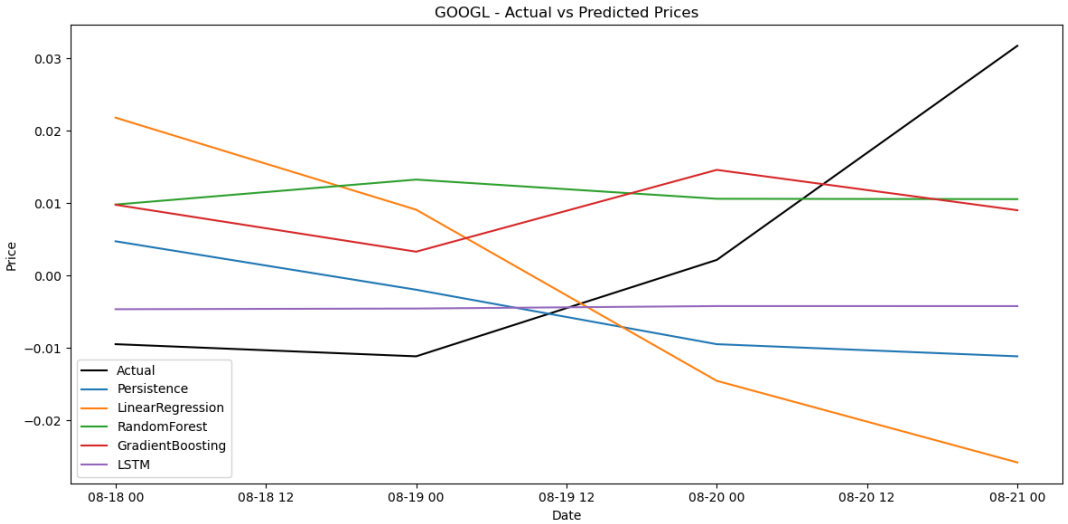

Figure 16: GOOGL-predictions

7. Technologies Used

Category	Tools / Libraries
Programming	Python 3.9+
Data Analysis	pandas, numpy
Visualization	matplotlib, seaborn, plotly
Machine Learning	scikit-learn, TensorFlow
App Deployment	Streamlit
Version Control	Git & GitHub

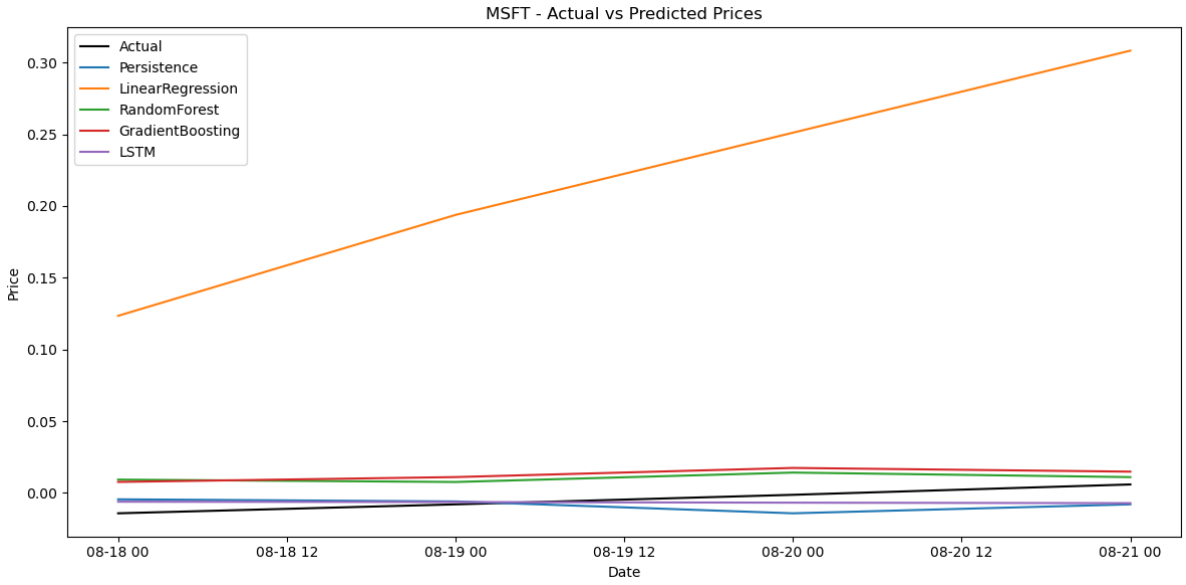

Figure 17: MSFT-predictions

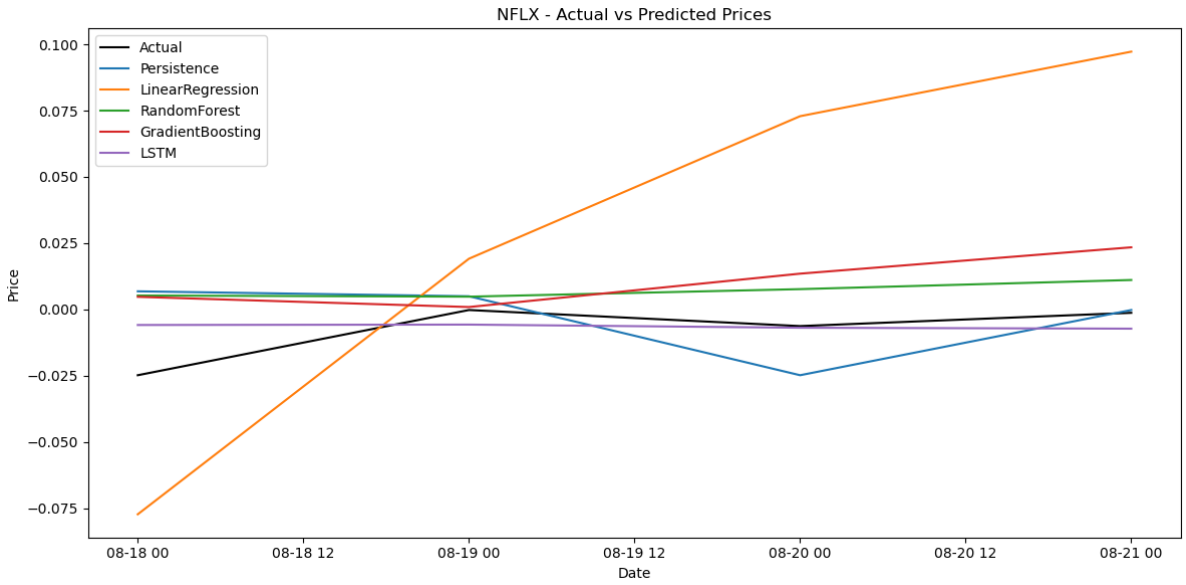

Figure 18: NFLX-predictions

Figure 19: Price-Charts(AAPL)

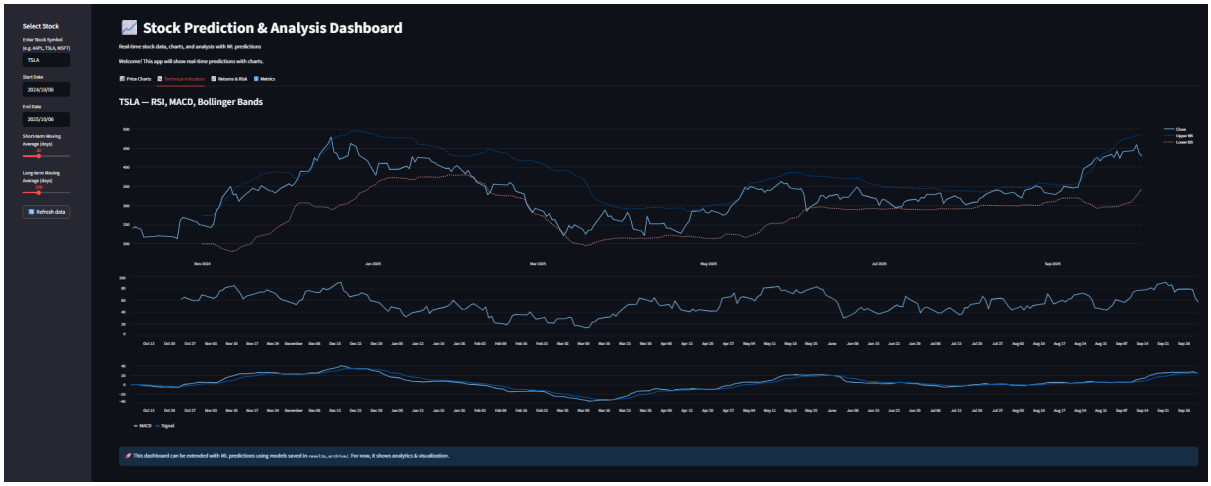

Figure 20: Technical-Indicators(TSLA)

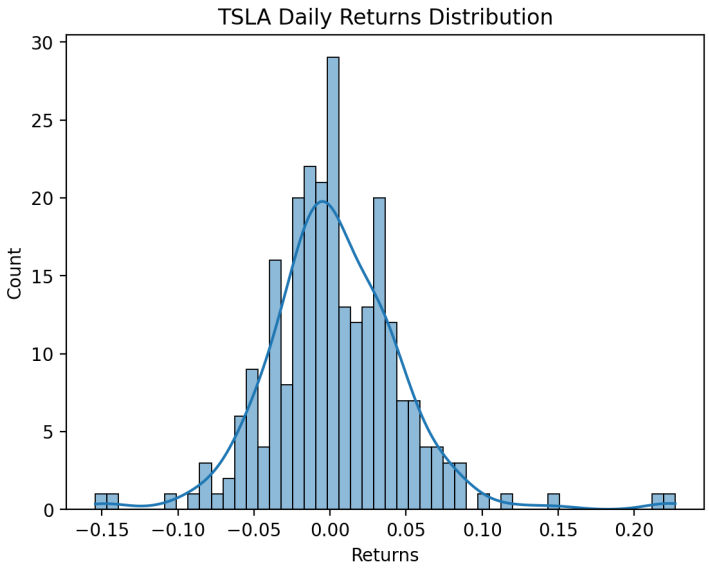

Figure 21: TSLA-Daily.Returns.Distribution

Figure 22: Returns-Risks(TSLA)

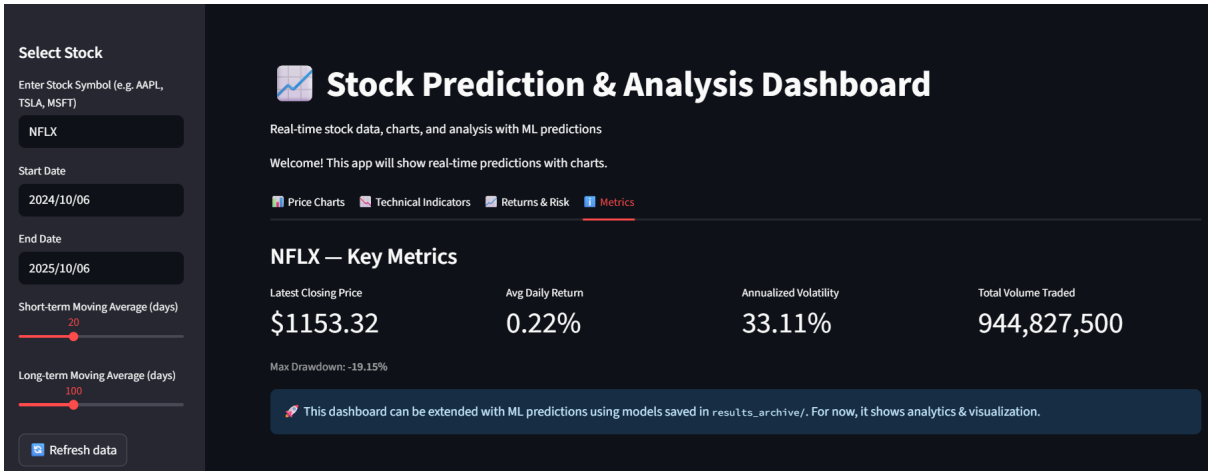

Figure 23: Metrices-NFLX

8. Conclusion

This project successfully demonstrated the integration of machine learning and finance analytics to predict and interpret stock market behavior. Predictive modeling, supported by data visualization, provided actionable insights for investment strategies.

Future Enhancements:

- Incorporate news sentiment and macroeconomic indicators.
- Extend to real-time API-driven dashboards.
- Implement reinforcement learning for portfolio optimization.

9. References

- Yahoo Finance API Documentation
- Scikit-learn Documentation: https://scikit-learn.org
- Streamlit Deployment Guide: https://streamlit.io
- Unified Mentor Internship Resources
- Live App You can access the live app here: [Stock Prediction Dashboard] https://share.streamlit.io/tanisha-m26/stock-app/main/app.py