Amazon Stock Price Forecasting

1. Project Objective

Forecasting the next 5 business-day closing prices (one trading week) of Amazon stock, using time-series modeling techniques, and identify the most effective predictive approach.

2. Project Summary

This project explores multiple forecasting models including ARIMA, Facebook Prophet, LSTM, and XGBoost to predict Amazon's next-week closing stock prices. The dataset is enriched with time-series features such as lag values, rolling statistics, and technical indicators (RSI, MACD, CMF). Among the evaluated models, XGBoost delivered the most accurate results, with a forecasting error of no more than 1.08%, and was therefore selected for deployment in the final web application.

3. Dataset Overview

Based on historical Amazon stock data (1997-2021)

4. Tools and Techniques Used

- Languages: Python
- Libraries: pandas, NumPy, scikit-learn, XGBoost, matplotlib, seaborn, statsmodels, prophet, keras, joblib
- Environment: Jupyter Notebook
- Deployment: Flask API for real-time customer input and forcasting

5. Feature Engineering

- Lag features (lag_1 to lag_7)
- Rolling statistics (mean, std, min, max)
- Technical indicators: RSI, MACD, CMF

6. Models Compared

- ARIMA
- Facebook Prophet
- XGBoost
- LSTM

7. Evaluation Metrics

Model	Mean Absolute Error	Root Mean Squared Error
ARIMA	3.85%	4.46%
Prophet	2.07%	2.39%
XGBoost	1.08%	1.44%
LSTM	1.44%	1.77%

LSTM and XGBoost performed best overall. ARIMA and Prophet were useful as a baseline models but had lower performance.

8. Key Takeaways

- XGBoost outperforms deep learning and classical models significantly because it leverages: rich technical indicators, engineered lag/rolling features, simpler training and less data-requirement compared to LSTM
- LSTM Underperformed, but with careful tuning and more improvement could come closer to XGBoost
- Flask app successfully deployed for real-time customer input and forcasting

9. Resources

GitHub Repo

Kaggle Notebook

Flask App Demo