Stock Forecasting Project Documentation

Amal Thomas

January 28, 2025

Contents

1 Introduction **2**

2 Script Overview **2**

3 Code Explanation **2**
3.1 Importing Libraries 2
3.2 Data Loading 2
3.3 Exploratory Data Analysis (EDA) 2
3.4 ARIMA Model 3
3.5 Gradient Boosting Model 3
3.6 Model Evaluation 3

4 Findings and Conclusion **3**

1 Introduction

This project focuses on forecasting stock prices using various machine learning and statistical models. It includes data preprocessing, exploratory data analysis, feature engineering, and the application of ARIMA and Gradient Boosting models for prediction.

2 Script Overview

The script is structured into the following steps:

1. Data Loading and Preprocessing
2. Exploratory Data Analysis (EDA)
3. ARIMA Model Implementation
4. Gradient Boosting Model Implementation
5. Model Evaluation

3 Code Explanation

Below is a detailed explanation of the code components:

3.1 Importing Libraries

The script imports necessary Python libraries for data manipulation, visualization, and modeling.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from sklearn.ensemble import GradientBoostingRegressor
from sklearn.metrics import mean_squared_error
```

3.2 Data Loading

The dataset is loaded and parsed, with 'Date' set as the index.

```
df = pd.read_csv('AAPL_with_features.csv', parse_dates=['Date'],
    index_col='Date')
```

3.3 Exploratory Data Analysis (EDA)

Missing values and trends are analyzed. Key findings:

- Missing values were identified and resolved.
- Stationarity was checked using the ADF test.

3.4 ARIMA Model

The ARIMA model was used to forecast stock prices based on historical data.

```
model = ARIMA(train, order=(1, 1, 1))
model_fit = model.fit()
predictions = model_fit.forecast(steps=len(test))
```

3.5 Gradient Boosting Model

Gradient Boosting was applied using engineered features like moving averages and lagged returns.

```
gbr = GradientBoostingRegressor()
gbr.fit(X_train, y_train)
y_pred = gbr.predict(X_test)
```

3.6 Model Evaluation

The performance of both models was evaluated using metrics such as RMSE, MAE, and MAPE.

4 Findings and Conclusion

- The ARIMA model performed well in capturing time-series trends.
- Gradient Boosting demonstrated better performance with engineered features.

Recommendations for trading strategies and future improvements are discussed.