Predicting Tesla stock price using Support Vector Machines, Gradient Boosting, Random Forest, and ARIMA

I. Model Selection and Methodology

The code uses the **yfinance** library to download the historical stock price data forTesla. It selects relevant columns ('Open', 'Close', 'High', 'Low') which are commonly used in stock market analysis.

This code includes a target variable **'Signal'**, which indicates whether the next day's closing price is higher than the current day's closing price. It then proceeds with feature normalization, model training, and predictions. **Mean Squared Error** (MSE) is calculated to evaluate the performance of each model. MSE measures the average squared difference between the predicted and actual prices. Lower MSE values also indicate better accuracy Finally, it plots the actual vs predicted close prices by each model over time for visual comparison.

The data is normalized using **MinMaxScaler** to scale all features to a range between 0 and 1. Normalization is essential for ensuring that all features contribute equally to the model training process, especially when using algorithms sensitive to feature scales, like SVM and Gradient boosting. The **create_dataset** function is defined to create input-output pairs suitable for supervised learning from the time series data. By choosing an appropriate time step (100 in this case), the model can learn from historical patterns and make predictions.

In this code I have used ***ARIMA*** (Autoregressive Integrated Moving Average), which is a classical time series forecasting method. It's chosen here because it is a linear regression model that learns parameters from the historical data and uses them for predictions on the stock market prices [1]. Another algorithm used here is ***Random Forest***. This is an ensemble learning method that builds multiple decision trees and combines their predictions [2]. It's chosen for its ability to handle non-linear relationships in the data, which can be useful for capturing the complex behaviour of stock prices. ***Gradient Boosting*** is another ensemble learning technique that builds trees sequentially, where each tree corrects the errors of the previous one. It's selected for its strong predictive power and capability to handle complex interactions between features [3]. Another powerful supervised learning algorithm is ***Support Vector Machine*** (SVM), used for classification and regression tasks [4]. It's chosen in this project for its ability to capture complex relationships in high-dimensional spaces and its flexibility in handling different types of data.

II. Advantages and Disadvantages:

ARIMA:

Advantages: Suitable for time series data, captures trend and seasonality, interpretable results.

Disadvantages: Assumes linear relationships, may not capture complex patterns, sensitive to outliers.

Random Forest:

Advantages: Handles non-linear relationships well, robust to overfitting, works well with large datasets.

Disadvantages: Black-box model, difficult to interpret, training time can be high for large datasets.

Gradient Boosting:

Advantages: High predictive accuracy handles complex interactions well, less prone to overfitting compared to Random Forest.

Disadvantages: Sensitive to noisy data, longer training time compared to simpler models like linear regression.

Support Vector Machine (SVM):

Advantages: Effective in high-dimensional spaces, versatile with different kernel functions, effective in capturing complex relationships.

Disadvantages: Computationally intensive, sensitive to choose of kernel and regularization parameters.

III. Analysis of Results:

Mean squared error (MSE) values:

ARIMA: MSE = 20206.33

Random Forest: MSE = 1614.01

Gradient Boosting: MSE = 2191.65

Support Vector Machine (SVM): MSE = 6107.26

(A) ***ARIMA*** performed reasonably well but had the highest MSE among the selected algorithms. The relatively high MSE suggests that ARIMA might struggle to capture the complexities and non-linearities present in the stock price data. ARIMA is known to perform better when the underlying data exhibits clear trends and seasonal patterns [1], which might not always be the case in stock market data. In this scenario, ARIMA may not have been able to adequately capture the dynamics of stock price movements.

(B) ***Random Forest*** achieved the lowest MSE among the selected algorithms, indicating superior performance in predicting the stock prices. Random Forest's ability to handle non-linear relationships and interactions between features likely contributed to its strong performance. The ensemble nature of Random Forest, where multiple decision trees are combined, allows it to capture complex patterns in the data effectively [2]. Random Forest is generally robust and less sensitive to outliers compared to other models like ARIMA.

(C) ***Gradient Boosting*** performed relatively well but had a higher MSE compared to Random Forest. Gradient Boosting builds trees sequentially, with each tree correcting the errors of the previous ones [3]. This approach typically results in high predictive accuracy. However, Gradient Boosting might be more sensitive to noisy data compared to Random Forest, which could have contributed to its slightly higher MSE in this case.

(D) ***Support Vector Machine (SVM)*** had the highest MSE among the selected algorithms, indicating poorer performance compared to Random Forest and Gradient Boosting. While SVM is powerful in capturing complex relationships [4], it might not have been the most suitable choice for this dataset. SVM's performance might have been affected by the choice of kernel and regularization parameters, which require careful tuning. Additionally, SVM is computationally intensive and might not scale well to larger datasets compared to ensemble methods like Random Forest and Gradient Boosting.

IV. Overall Analysis:

Random Forest outperformed the other algorithms in terms of predictive accuracy for the given dataset, achieving the lowest MSE. Gradient Boosting also performed reasonably well but had a slightly higher MSE compared to Random Forest. ARIMA and SVM had higher MSE values, suggesting that they might not have been the most suitable choices for this particular stock price prediction task.

References:

[1] P. Cowpertwait, A. Metcalfe. Introductory time series with R. 2009. https://link.springer.com/book/10.1007/978-0-387-88698-5

[2] Machine Learning Mastery. https://machinelearningmastery.com/random-forest-for-time-series-forecasting/

[3] Gradient Boosting. https://statlect.com/machine-learning/gradient-boosting

[4] M. Bazrkar, S. Hosseini. Predicting Stock Prices using Supervised Learning Algorithms. *Compute Econ* **62**, 165–186 (2023). https://link.springer.com/article/10.1007/s10614-022-10273-3