Model Selection Documentation-Task 4

1. Comparison of Different Modeling Approaches

To predict stock prices, we experimented with the following models:

a) Long Short-Term Memory (LSTM) Network

- Deep learning model specifically designed for time series forecasting.
- Captures long-term dependencies in sequential data.
- Requires careful tuning of hyperparameters.
- Computationally expensive and requires more training time.

b) XGBoost (Extreme Gradient Boosting)

- A powerful ensemble learning model.
- Works well with tabular data.
- Requires feature engineering to extract useful trends.
- Does not naturally capture sequential dependencies like LSTMs.

c) ARIMA (AutoRegressive Integrated Moving Average)

- Traditional statistical model for time series forecasting.
- Works well when time series exhibits a clear trend and seasonality.
- Limited ability to handle complex patterns and requires assumptions about stationarity.

d) Random Forest Regressor

- Tree-based ensemble model.
- Can handle nonlinear relationships in the data.
- Does not inherently consider temporal dependencies.

2. Explanation of Evaluation Metrics

To compare model performance, we used the following evaluation metrics:

- **Root Mean Squared Error (RMSE):** Measures the average error magnitude. Lower values indicate better performance.
- **Mean Absolute Error (MAE):** Measures the average absolute difference between actual and predicted values.

- **Directional Accuracy:** Measures how often the model correctly predicts whether the price will go up or down.

3. Justification for Final Model Choice

Selected Model: LSTM

- LSTMs demonstrated superior performance in capturing trends and long-term dependencies in stock price movements.
- Achieved the lowest RMSE and higher directional accuracy compared to other models.
- While XGBoost and ARIMA performed well in short-term predictions, they struggled with long-term dependencies.
- Random Forest was less effective due to the lack of sequential learning.

4. Model Limitations and Potential Improvements

Limitations:

- **Data Sensitivity:** LSTM requires a large amount of historical data for effective training.
- **Computational Cost:** Training deep learning models requires more resources compared to traditional models.
- **Feature Dependence:** Performance can be improved with additional technical indicators and external data (e.g., market sentiment, economic indicators).
- **Overfitting:** Without proper regularization, LSTM can overfit to historical trends.

Potential Improvements:

- **Hybrid Models:** Combining LSTM with XGBoost or ARIMA for enhanced predictive performance.
- **Feature Engineering:** Introducing additional features like moving averages, Bollinger Bands, and trading volume.
- **Hyperparameter Tuning:** Fine-tuning LSTM layers, batch size, and dropout rates to optimize performance.
- **Alternative Deep Learning Models:** Exploring GRU (Gated Recurrent Units) and Transformer-based models for improved predictions.

Conclusion

LSTM was selected as the final model due to its ability to capture long-term dependencies and provide accurate forecasts. Future improvements could involve hybrid modeling and more advanced deep learning architectures to enhance performance further.