COMP 5313: ARTIFICIAL INTELLIGENCE

PROJECT 1

Stock Price Prediction using AI and Machine Learning

Name: Supprethaa Shankar

Student ID: 1229712

Task:

Choose a model (or a set of models) like using a neural network to perform a regression function for example. The next logical steps will be building and training your model to predict the stock prices. You need to demonstrate how effective your model(s) in predicting stock prices for the selected data.

Dataset:

The dataset consists of the historical stock data of Netflix (past 5 years) and has the following columns:

- Date
- Open
- Close
- Adjusted Close
- Volume

We are interested in predicting the future closing prices based on the historical trends. Our columns of interest are ‘Date’ and ‘Close’ for this purpose.

AI_StockPricePrediction_XGBoost.ipynb:

In this file, the Stock Price Prediction is implemented using **XGBoost** model, which is a popular machine learning algorithm that belongs to the family of ensemble learning methods. XGBoost is primarily based on decision trees, often referred to as "boosted trees." These trees are shallow, meaning they have a limited depth, which helps prevent overfitting. It is capable of capturing temporal patterns in time-series data. Its ability to build sequential models and handle temporal dependencies makes it suitable for time-series forecasting tasks.

Steps:

1. The dataset is pre-processed and only the Date and Close price columns are retained.
2. Creating lag features for the 'Close' prices in the dataset. Lag features involve shifting the values of a particular column (in this case, the 'Close' column) by a specified number of time steps. These lagged values can capture temporal patterns and dependencies in time-series data, making them useful for predictive modeling.
3. Splitting the dataset into Train and Test (80:20 ratio).
4. Building the XGBoost model and fitting the Training data
5. Making predictions on Test data and evaluation of model performance.

AI_StockPricePrediction_LSTM.ipynb:

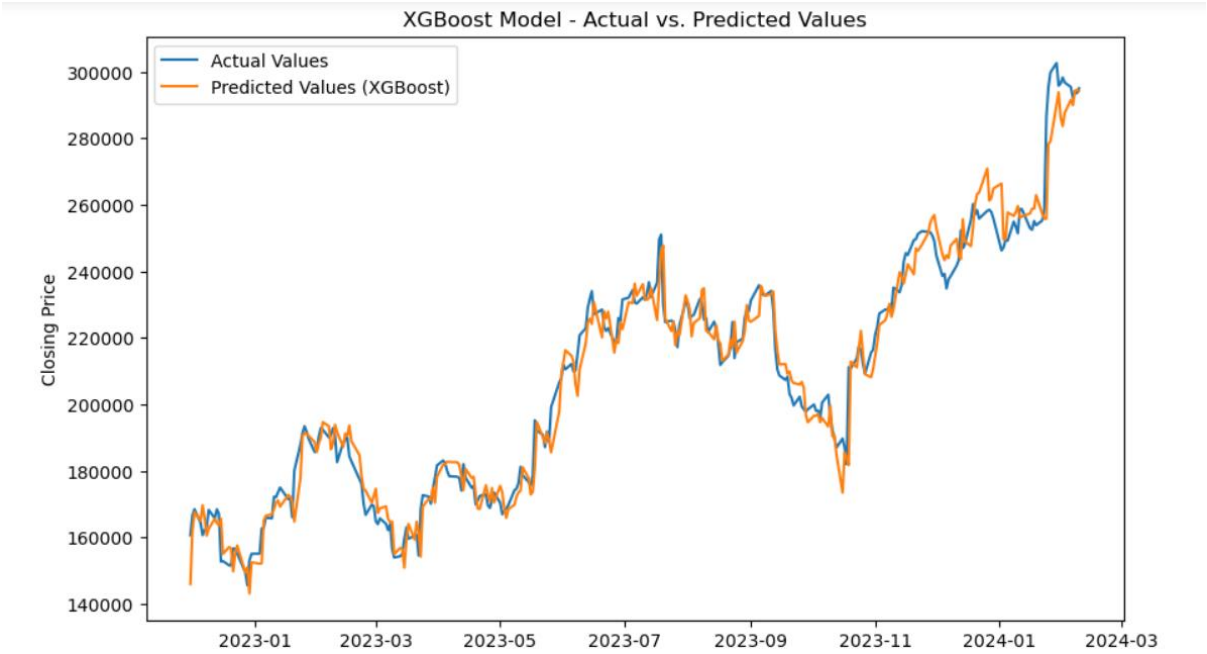
In this file, the Stock Price Prediction is performed using LSTM Recurrent Neural Network model. Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) architecture designed to address the vanishing gradient problem, which can occur in traditional RNNs when training on long sequences of data.

LSTMs can adapt to different patterns and trends in the time-series data, making them versatile for various types of temporal information.

Steps:

1. The dataset is pre-processed and only the Date and Close price columns are retained.
2. Basic visualizations are performed to view the distribution of the data
3. Data is scaled using MinMax layer
4. Creating historical sequences for the LSTM model to capture previous trends
5. Building the LSTM model with 50 units and one Dense layer
6. Splitting the dataset into Train and Test (80:20 ratio)
7. Training the LSTM model using the Training data
8. Making predictions on Test data and evaluating model performance.
9. Making future predictions (next 10 days from Test data) based on captured trends

XGBoost Performance:



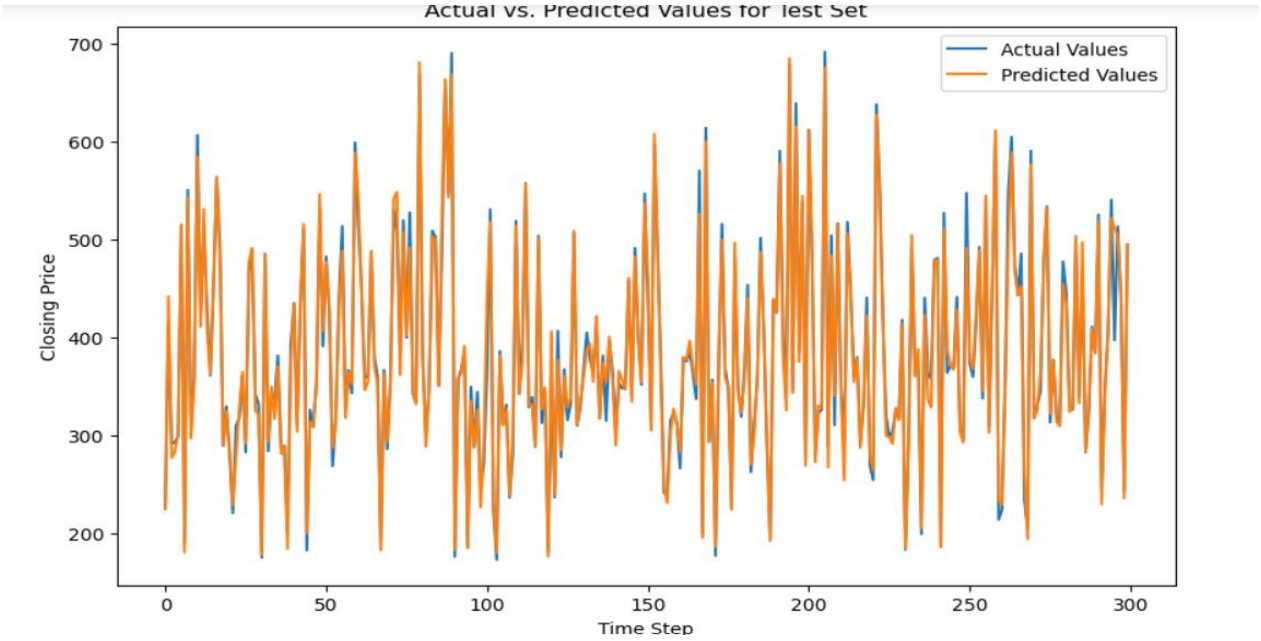
Root Mean Squared Error

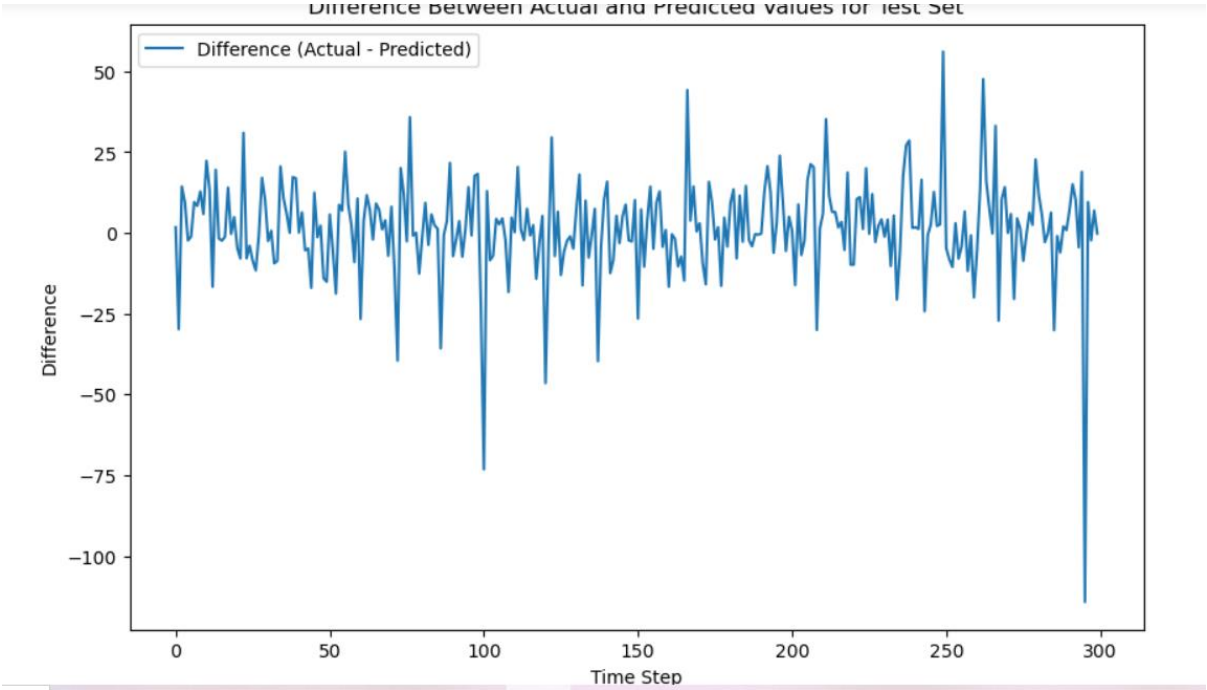
```
RMSE: 6332.039270048227
```

Mean Absolute Error

```
Mean Absolute Error (MAE): 4672.530891760598
```

LSTM Performance:






```
Mean Squared Error (MSE): 494.01069499197484
Root Mean Squared Error (RMSE): 22.226351364809627
Mean Absolute Error (MAE): 14.380532517649742
```

Comparison of Model Performances:

The reported performance metrics for the LSTM and XGBoost models indicate that the LSTM model is performing better based on the Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE).

Possible reasons can be:

- LSTM models, being a type of neural network, have the ability to capture complex non-linear relationships in the data. They can learn intricate patterns and dependencies, especially in time-series data; while XGBoost, while powerful, is based on decision trees and might struggle to capture long-term dependencies present in time-series data.

- LSTM models can automatically learn hierarchical features from the sequential data. XGBoost relies on manually engineered features, and the effectiveness of these features could impact its performance.

- LSTM models are well-suited for sequential and time-dependent data, which might be the case for stock price data. They can capture temporal patterns and trends effectively, whereas XGBoost might struggle with capturing sequential dependencies, as it doesn't inherently consider the temporal order of the data.

Future predictions by LSTM:

	Future Predictions
2024-02-09	572.002197
2024-02-10	572.118469
2024-02-11	573.436035
2024-02-12	575.569824
2024-02-13	578.149414
2024-02-14	580.881104
2024-02-15	583.602234
2024-02-16	586.239075
2024-02-17	588.816345
2024-02-18	591.357971