Stock Market analysis and prediction using LSTM
EE769 Project

Piyush Mourya
Roll no. - 200260037
Engineering Physics
Email: 200260037@iitb.ac.in

Lokesh Mishra
Roll no. - 200260026
Engineering Physics
Email: 200260026@iitb.ac.in

Nirmal Solanki
Roll no. - 200260034
Engineering Physics
Email: 200260034@iitb.ac.in

***Abstract*—This project uses neural networks to analyze and predict the closing price of Microsoft stock. The objective is to develop a model that can capture underlying patterns and trends in historical stock data to make accurate predictions. The project follows a systematic methodology that includes data collection, preprocessing, feature selection, neural network model development, training, validation, evaluation, and prediction.**

The project begins with collecting historical stock data for Microsoft, including closing prices and other relevant features.

A neural network architecture suitable for time series forecasting is designed and implemented. Recurrent neural networks (RNNs) or long short-term memory (LSTM) networks are considered to effectively capture temporal dependencies. The model is trained using a training dataset while monitoring its performance on a validation set to prevent overfitting.

The trained model's performance is evaluated using evaluation metrics such as mean squared error (MSE), mean absolute error (MAE), or root mean squared error (RMSE). A comparison is made between the model's predictions and the actual closing prices to assess its accuracy.

Finally, the developed model is applied to make predictions on unseen data, such as future time periods. The accuracy of the predictions is evaluated, and the predicted closing prices are visualized alongside the actual prices to access the model's performance in real-world scenarios.

Overall, this project provides insights into the process of analyzing and predicting stock prices using neural networks, specifically focusing on Microsoft stock. The findings can be beneficial for investors, financial analysts, and researchers interested in leveraging neural networks for stock price forecasting

1. Introduction

This project aims to analyze historical data of Microsoft stock and develop a neural network model to predict the stock's closing price. By leveraging the power of neural networks, we hope to uncover hidden patterns and trends in the stock data that can help us make better investment decisions. Stock price prediction is a challenging task that involves many factors, such as market conditions, company performance, news events, investor sentiment, etc. Traditional methods, such as statistical or technical analysis, may not be able to capture the complex and nonlinear relationships among these factors. Neural networks, on the other hand, are capable of learning from large amounts of data and adapting to changing environments. They can also handle noisy and incomplete data, which are common in financial markets.

In this project, we focus on Microsoft stock because it is one of the most widely traded and influential companies in the world. Microsoft is a leader in software development, cloud computing, artificial intelligence, gaming, and hardware. Its products and services are used by millions of people and businesses around the globe. Microsoft's stock price reflects its performance and growth, as well as the expectations and opinions of investors and analysts. Predicting Microsoft's stock price can help us understand its future prospects and potential risks.

2. Methodology

The project follows the following methodology:

2.1. Data collection:

We collect historical stock data for Microsoft from an online source, such as Yahoo Finance or Google Finance. We obtain daily closing prices and other relevant features, such as volume, open, high, low, etc., for a certain time period.

2.2. Data preparation:

The Microsoft stock dataset ('MSFT.csv') is loaded into a pandas DataFrame, focusing on the 'Date' and 'Close' columns. The 'Close' column represents the closing price of the stock. The data is visualized using line plots to understand the trends and patterns.

2.3. Data preprocessing:

To prepare the data for LSTM modeling, the 'Date' column is converted from string to datetime format. The conversion enables better manipulation and visualization of the data. Additional visualizations are created to examine the volume and moving averages of the stock.

2.4. Model Building :

The LSTM model is implemented using the Keras library with TensorFlow as the backend. The model architecture consists of an input layer, an LSTM layer with 64 units, and multiple dense layers with ReLU activation. The model is compiled with the mean squared error (MSE) loss function and the Adam optimizer.

2.5. Neural network model development:

We design and implement a neural network architecture that is suitable for time series forecasting. We consider different types of neural networks, such as recurrent neural networks (RNNs) or long short-term memory (LSTM) networks, that can effectively capture temporal dependencies in the stock data.

2.6. Training and validation:

The dataset is split into training, validation, and testing sets. The training set comprises 80% of the data, while the validation and testing sets each contain 10% of the data. The model is trained on the training set and validated on the validation set for 100 epochs. The mean absolute error (MAE) is used as the evaluation metric.

2.7. Evaluation:

We evaluate the performance of the trained model using appropriate evaluation metrics here we used accuracy. We also apply the trained model to make predictions on test data and compare them with the actual closing prices.

2.8. Prediction:

Apply the trained model to make predictions on test data and evaluate its accuracy.

2.9. Result and analysis :

The training and testing results are visualized by plotting the predicted closing prices against the actual prices. The training predictions show a good fit with the training observations, indicating that the model has learned the patterns in the training data. The testing predictions also provide reasonable estimates of the actual prices, suggesting that the model generalizes well to unseen data.

3. Conclusion

The project successfully developed a neural network model that could predict the closing price of Microsoft stock. The methodology used in the project can be replicated for other stock price forecasting tasks, and the insights gained from the project can be valuable for various stakeholders. The project also demonstrated the effectiveness of using deep learning techniques for time series analysis, and the challenges and limitations of applying them to real-world data. The project provided a comprehensive evaluation of the model performance and suggested some possible directions for future improvement and research. Despite all the 'quite nicely' matching train and test values of neural networks one has to see this from a real-life point of view. If you try to match it with the real-life application, you see there are lots of problems you are highly probable to go through. Some of them are listed below:
1) We only used previous closing prices to predict the next ones but in real life, it actually depends on various parameters.
2) In modeling the neural networks we use and fit train data which is the biggest problem of neural networks. Now in a case like the stock market where features like price can vary too much in comparison to train data, it will lead to bad prediction.
3) Using prices as features and target is a problem because generally our features and target need to be in some bounded region but we know stock prices are not bounded so it is very much probable to bad prediction.

4. Contributions

Lokesh Mishra : Analysis of stock data
Piyush Mourya : Making neural network and Prediction
Nirmal Solanki : Data collection, Background research and Report making

References

https://www.kaggle.com/code/bryanb/stock-prices-forecasting-with-lstmKaggle source
For various doubts used ChatGPT and youtube.