IBM NAAN MUDALVAN

KINGS ENGINEERING COLLEGE-2108

Domine: Applied Data Science-Phase 3

Team Members

K MANIMARAN :manimaran31072004@gmail.com

D RISWIN :riswinfdo@gmail.com

SHARONJEBASTEEVE:sharonjebasteeve46@gmail.com

P RAMESH :pmramesh03@gmail.com

STOCK PRICE PREDICTION

Introduction:

such as cryptocurrency price predictionIn this article, we will study the applications of neural networks on time series forecasting to accomplish stock market price prediction. You can find the full code for this tutorial and run it on a free GPU from the ML Showcase. While you can directly run the entire notebook on your own, it is advisable that you understand and study each individual aspect and consider code blocks

individually to better understand these concepts. The methodology for this project can be extended to other similar applications s as well.

Table of Contents:

Preparing the data

The first step to complete this project on stock price prediction using deep learning with LSTMs is the collection of the data. We are going to consider a random dataset from Kaggle, which consists of Apple's historical stock data. We are going to read the CSV file using the Panda's library, and then view the first five elements of the data.

```
# Importing the Pandas Library for loading our dataset

# Grab The Data Here
-https://www.kaggle.com/datasets/prasoonkottarathil/microsoft-lifetime-stocks-dataset
```

```
import pandas as pd

df = pd.read_csv('HistoricalQuotes.csv')

df.head()
```

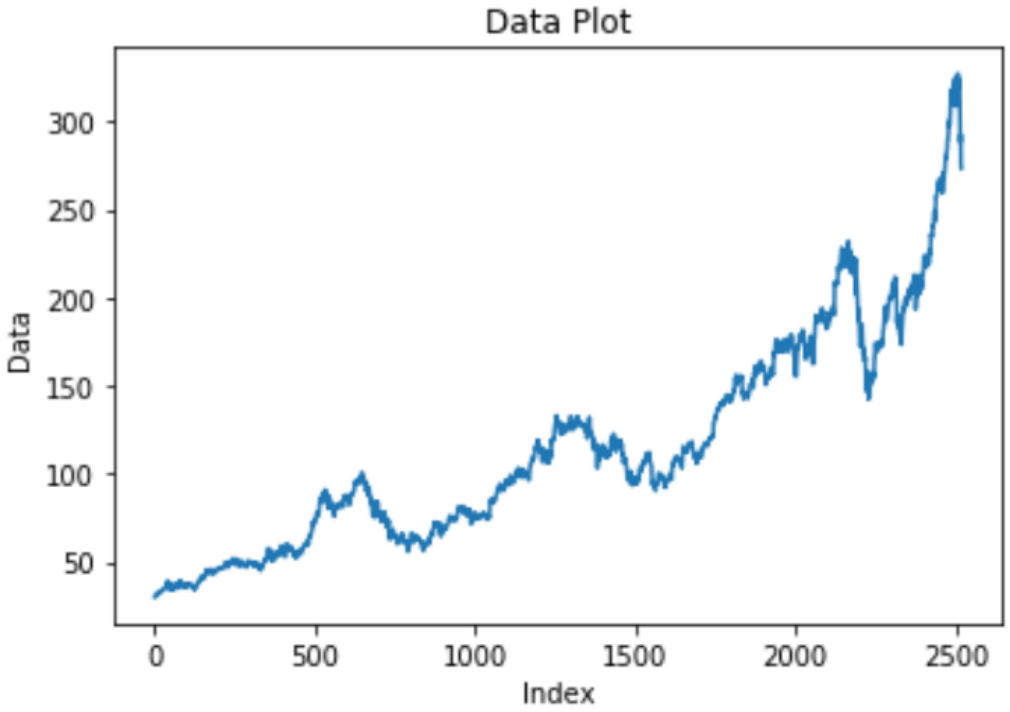
	Date	Close/Last	Volume	Open	High	Low
0	02/28/2020	$273.36	106721200	$257.26	$278.41	$256.37
1	02/27/2020	$273.52	80151380	$281.1	$286	$272.96
2	02/26/2020	$292.65	49678430	$286.53	$297.88	$286.5
3	02/25/2020	$288.08	57668360	$300.95	$302.53	$286.13
4	02/24/2020	$298.18	55548830	$297.26	$304.18	$289.23

Visualize Your Data

For any machine learning and deep learning problem, one of the most crucial steps is the visualization of your data. Once you visualize and pre-process the data, you can have a brief understanding of the type of model you are dealing with and the necessary steps and measures required to solve the task. One of the best visualization libraries in the Python programming language is the Matplotlib library. It will allow you to visualize the dataset accordingly. Let us plot the model with the data and their respective indexes. The data consists of the values of the stocks at their respective intervals.

program:

```
# Using the Matplotlib Library for visualizing our time-series data

import matplotlib.pyplot as plt

plt.title("Data Plot")

plt.xlabel("Index")

plt.ylabel("Data")

plt.plot(df1)
```



Analyzing The Data

Now that we have a brief understanding of what we are trying to achieve with the previous explanations and code blocks, let us try to gain further insight on this topic by actually analyzing

and looking at the first two sets of training data and their respective outputs that are stored in the output dataset.

program:

X_train_data[0]

Result:

array([0. , 0.00023069, 0.00089352, 0.0048523 , 0.00491451,

array([0. , 0.00023069, 0.00089352, 0.0048523 , 0.00491451,

0.00680747, 0.00768182, 0.00799894, 0.00852725, 0.00720127,

0.00749451, 0.00733578, 0.00759035, 0.00643756, 0.00763844,

0.00937268, 0.00985794, 0.00855146, 0.01059307, 0.01130902,

0.0129686 , 0.01256271, 0.0130288 , 0.01423944, 0.01474387,

0.01525301, 0.01494093, 0.0158247 , 0.01606481, 0.01613206,

0.01769849, 0.01925013, 0.01851971, 0.01836132, 0.01716985,

0.02419826, 0.0276812 , 0.02977593, 0.02913699, 0.02555317,

0.02534164, 0.02872369, 0.02509683, 0.02762369, 0.02393899,

0.02264428, 0.01796752, 0.012976 , 0.02168586, 0.0229012 ,

0.02557704, 0.0237853 , 0.02160414, 0.02179616, 0.02090264,

0.01897134, 0.01388869, 0.01607927, 0.01821234, 0.01747251,

0.01693882, 0.02137815, 0.02307405, 0.02497173, 0.02647056,

0.02607206, 0.02263453, 0.02022065, 0.01944719, 0.01650198,

0.02001383, 0.02145516, 0.02182508, 0.02442425, 0.02805616,

0.03027566, 0.03133429, 0.02945882, 0.03122668, 0.02984319,

0.02889688, 0.02779184, 0.02856059, 0.02273306, 0.02050381,

0.0190386 , 0.01829877, 0.0191109 , 0.02393159, 0.0236555 ,

0.02439062, 0.02326876, 0.0206326 , 0.02107886, 0.02046547,

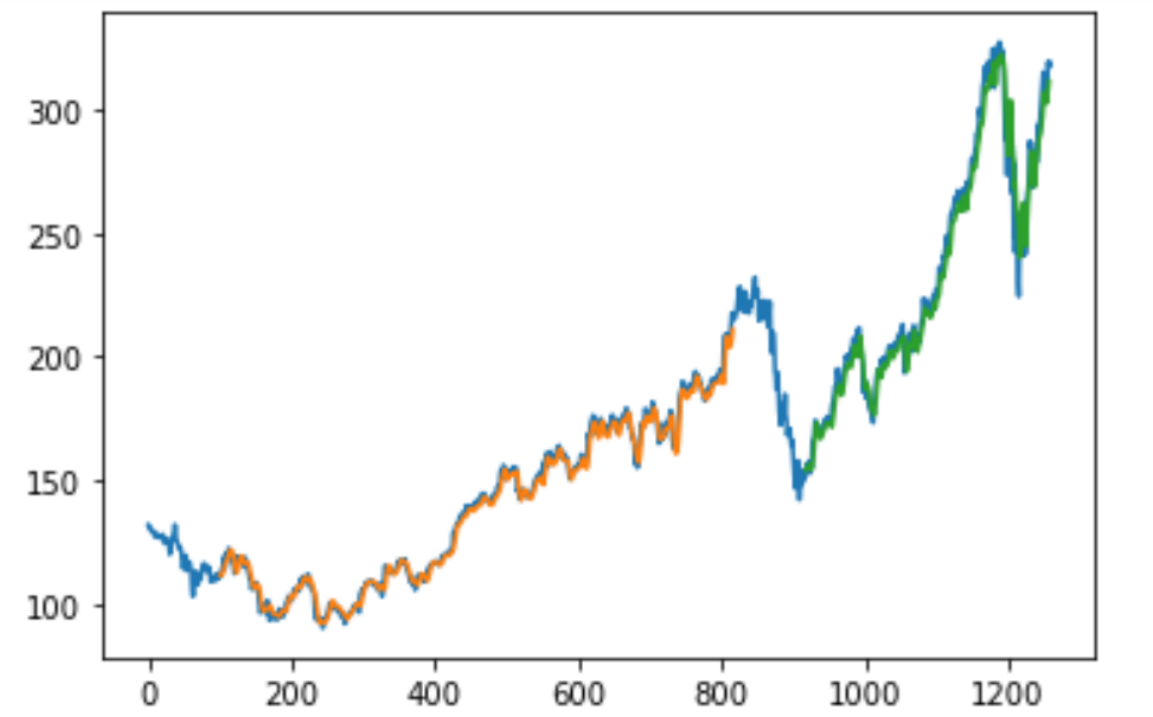
0.01972093, 0.01764536, 0.02067699, 0.02180591, 0.0241041])

Predictions and Evaluations

In the next few code blocks, we will use the predict function from the model to make the respective predictions on the training and testing data. It is essential to note that we had previously converted the range of values for the training and testing data from 0 to 1 for faster and efficient computation. Hence, it is now time to use the inverse transform attribute from the scaler function to return the desired values. By making use of this procedural method, we will achieve the required result that we need to attain since the values obtained are scaled back to their original numbers.

```
train_predict = model.predict(X_train_data)

test_predict = model.predict(X_test_data)

# Transform back to original form

train_predict = scaler.inverse_transform(train_predict)

test_predict = scaler.inverse_transform(test_predict)


# Calculate RMSE performance metrics for train and test data

import math

from sklearn.metrics import mean_squared_error

print("Train MSE = ",
math.sqrt(mean_squared_error(Y_train_data, train_predict)))

print("Test MSE = ",
```

math.sqrt(mean_squared_error(Y_test_data, test_predict)))



Conclusion:

We have succesfully finish the Stock market prediction has been a significant area of research in Machine Learning. Machine learning algorithms such as regression, classifier, and support vector machine (SVM) help predict the stock market. This article presents a simple implementation of analyzing and forecasting stock market prediction using machine learning.