STOCK MARKET PREDICTION USING LSTM

## MINI PROJECT REPORT

***Submitted by,***

## PALURU PAVAN KUMAR REDDY (723920243037)

## PERUVAILU KAMAL (723920243039)

## R.PREM KUMAR (723920243040)

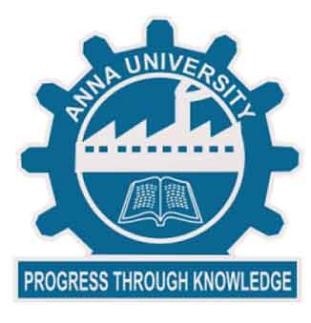
*In partial fulfilment for the award of the degree*

***of***

## BACHELOR OF TECHNOLOGY

***in***

## ARTIFICIAL INTELLIGENCE AND DATA SCIENCE



**ARJUN COLLEGE OF TECHNOLOGY**

**COIMBATORE- 642 120**

## ANNA UNIVERSITY: CHENNAI 600 025

## JANUARY 2023

# ANNA UNIVERSITY: CHENNAI 600 025

**BONAFIDE CERTIFICATE**

## Certified that this Report titled “STOCK MARKET PREDICTION USING LSTM” is the Bonafide work of PALURU PAVAN KUMAR REDDY (723920243037), PERUVAILU KAMAL (723920243039), R.PREM KUMAR (723920243040) who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported here in does not form part of any other project work on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

|  |  |
| --- | --- |
| **SIGNATURE**  **Ms.R.ROOPA M.E.,**  **SUPERVISOR**  Assistant Professor  Department of AI & DS  Arjun College of Technology  Coimbatore -642 120 | **SIGNATURE**  **Dr. J. THILAGAVATHI MCA., Ph.D.,**  **HEAD OF THE DEPARTMENT**  Associate Professor  Department of AI & DS  Arjun College of Technology  Coimbatore -642 120 |

Submitted for the university project viva-voce held on \_\_\_\_\_22-01-2023\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We owe our sincere and heartfelt thanks to our chairman **Thiru. R. SURIYANARAYANAN,** and also we extend our profound thanks to our Secretary **Dr. R. SURESH KUMAR M.E., MBA., Ph.D.,** for their exuberance in motivating young minds.

Our deepest gratitude and thanks to our motivator and Principal **Dr. D. KUMAR M.E., MBA., Ph.D.,** who always helping us whenever we approach him during the course of our project.

We would also like to express our profound thanks to our Project Coordinator **Dr. J. THILAGAVATHI MCA., Ph.D.,** Associate Professor & Head, Department of AI & DS, whose thoughtful words, advise and help to complete our project successfully.

Our sincere gratitude and unplumbed thanks to our beloved project guide **Ms.R.ROOPA M.E.,** Assistant Professor, Department of AI & DS, for her constant encouragement, Valuable Guidance and constructive criticism in making this project a successful one.

We express our sincere thanks to all **Faculty Members and Skilled Assistants** of Artificial Intelligence and Data Science Department and our lovable **friends** for their help and wishes for the successful completion of this project.

Finally, yet importantly, we would like to express our indebtedness to our beloved **parents** for their affectionate blessing co-operation at all stages of this academic venture and also our well-wishers.

ABSTRACT

The Disease Prediction System based on various prediction models that help to

predict the disease of the user on the basis of the symptoms that user enters as

an input to the system. Predictive models with the help of machine learning clas-

sification algorithms analyzes the symptoms provided by the user as input and

gives the name and probability of the disease as an output. Disease Prediction

is done by implementing the Naive Bayes Classifier, Decision tree and Random

Forest Algorithm. The Naive Bayes helps to calculate the probability of the dis-

ease which is predicted. Average prediction accuracy probability 87% is obtained.

The model uses a dataset with the count of 132 symptoms from which the user

can select their symptoms. The user does not need to have a medical report to

use this system as the prediction is based on the symptoms which will save the

money. The system also has a very easy to use user interface so all the users can

use it to predict the generic diseases.

ABSTRACT

The Disease Prediction System based on various prediction models that help to

predict the disease of the user on the basis of the symptoms that user enters as

an input to the system. Predictive models with the help of machine learning clas-

sification algorithms analyzes the symptoms provided by the user as input and

gives the name and probability of the disease as an output. Disease Prediction

is done by implementing the Naive Bayes Classifier, Decision tree and Random

Forest Algorithm. The Naive Bayes helps to calculate the probability of the dis-

ease which is predicted. Average prediction accuracy probability 87% is obtained.

The model uses a dataset with the count of 132 symptoms from which the user

can select their symptoms. The user does not need to have a medical report to

use this system as the prediction is based on the symptoms which will save the

money. The system also has a very easy to use user interface so all the users can

use it to predict the generic diseases.

ABSTRACT

The Disease Prediction System based on various prediction models that help to

predict the disease of the user on the basis of the symptoms that user enters as

an input to the system. Predictive models with the help of machine learning clas-

sification algorithms analyzes the symptoms provided by the user as input and

gives the name and probability of the disease as an output. Disease Prediction

is done by implementing the Naive Bayes Classifier, Decision tree and Random

Forest Algorithm. The Naive Bayes helps to calculate the probability of the dis-

ease which is predicted. Average prediction accuracy probability 87% is obtained.

The model uses a dataset with the count of 132 symptoms from which the user

can select their symptoms. The user does not need to have a medical report to

use this system as the prediction is based on the symptoms which will save the

money. The system also has a very easy to use user interface so all the users can

use it to predict the generic diseases.

## DECLARATION

We, **PALURU PAVAN KUMAR REDDY, PERUVAILU KAMAL, R.PREM KUMAR** of fifth semesterB.Tech, in the department of Artificial Intelligence and Data Science from Arjun College of Technology, Coimbatore, hereby declare that the project work entitled **STOCK MARKET PREDICTION USING LSTM** is carried outby us and submitted in partial fulfilment of the requirements for the award of **Bachelor of Technology in Artificial Intelligence and Data Science**, under Arjun College of Technology, Coimbatore, during the academic year 2020-2024 and has not been submitted to any other university for the award of any kind of degree.

Paluru Pavan Kumar Reddy - 723920243037

Peruvailu Kamal - 723920243039

R.Prem Kumar - 723920243040

## ABSTRACT

Stock market investment is one of the most complex and sophisticated way to do business. Stock market is very uncertain as the prices of stocks keep fluctuating because of several factors that makes prediction of stocks a difficult and extremely complicated task. Now a days investors need fast and accurate information to make effective decisions and highly interested in the research area with exponentially growing technological advances of stock price prediction. Understanding the pattern of stock price of a particular company by predicting their future development and financial growth will be highly beneficial. Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without being explicitly programmed. It consists of making computers performs tasks that normally requiring human intelligence is currently the dominant trend in the scientific research. Usually the stock prices are correlated within the nature of the market, hence it will be difficult to predict the costs. This project seeks to solve the problem of stock price prediction by deep learning model known as Long-Short Term Memory(LSTM)Neural Network algorithm, to predict the future value of the stocks. This can be considered as a Time series analysis is a specialized branch of statistics used extensively in fields such as Econometrics & Operation Research. This system will provide accurate outcomes in comparison to currently available stock price predictor algorithms. To achieve this goal, I trained traditional machine learning algorithms and trained multiple deep learning models taking into consideration the importance of the relevant news. This will provide more accurate results when compared to existing stock price prediction algorithms. Various experiments Ire conducted, the highest accuracy(99.01%) of which was achieved using LSTM for APPLEe Inc. (AAPL) stock dataset.

**CONTENTS**

Acknowledgement i

Abstract ii

1. **Introduction 1**
   1. Stock Price Prediction using LSTM
   2. Motivation
   3. Objectives
   4. Chapter wise Summary
2. **Analysis and Design 3**
   1. Architecture and Techniques
   2. Deep recurrent neural networks
   3. Tools
   4. Flow and sequence Diagram
   5. Sequence Diagram
   6. Activity Diagram
   7. Component Diagram
3. **Implementation** . **16**
   1. Overview of the project
   2. Data Preparation
   3. Visualizing the data
   4. Heat Map
   5. Training and Testing
4. **Result analysis 28**
   1. Model Testing
   2. Evaluation of the model
   3. Verification
5. **Conclusion 36**
6. **References 37**

**List of diagrams Page No**

* + 1. General Architecture of LSTM 3
    2. LSTM Cell 4
    3. The internal structure of an LSTM 4
    4. The neural network model without and with dropout 5
    5. Deep recurrent neural networks based on LSTM 6
    6. A structural model of associated net 7
    7. The data processing flow chart of associated net 8
    8. Structured Diagram 9

2.5.1 Flow chart for Stock Prediction 11

* + 1. Detail Flow chart for Stock Prediction 12
  1. Sequence Diagram 13
  2. Activity Diagram 14
  3. Component Diagram 15
     1. Flow chart 17
     2. Pre-Processing of data 18
     3. Dataset APPLE 18
     4. Dataset MSFT 18
     5. Mean value of MSFT 19
     6. Mean value of APPLE 20
     7. Volume of the Stocks 21
     8. Closing price of APPLE 22
     9. Closing price of MSFT 22
     10. High price stocks of APPLE 23
     11. High price stocks of MSFT 23
     12. Bar chart of Adj close value 24
     13. Heat Map of APPLE dataset 25
     14. Heat Map of MSFT dataset 25
     15. LSTM model 27
     16. Model summary 27
     17. Prediction of MSFT (Actual vs Predicted) 28
     18. 30 days forecasting of APPLE dataset 29
     19. Forecasting output 29
     20. Evaluation of APPLE dataset 31
     21. Evaluation of MSFT dataset 31

# 1.INTRODUCTION

* 1. **Stock Market Prediction Using LSTM**

Stock price predictions are very important among many business people and the public. People can make a lot of money or lose their financial income from a stock market job. Algorithm predictions and models can be used to make future predictions Applied to historical data. Predicting the future has been a difficult task, one that many have found difficult to understand. This type of prediction is even more appealing when it involves money and risks such as Stock Market speculation. Researchers have tried a variety of methods to predict the market, including different strategies and algorithms and the combination of indicators. The attribute that makes a prediction model depends on factors on which market performance can depend.

Models capable of predicting future values based on previously observed values are known as time-series forecasting models. For non-stationary data, time- series forecasting is widely used. Non-stationary data are those whose statistical features, such as mean and standard deviation, do not remain constant throughout time but instead change. The non-stationary input data used as input to these models is typically called time-series. Temperature values over time, stock prices over time, and the price of a property over time are all instances of time series. As a result, the input is a signal characterized by observations recorded in a specific order throughout time.

Neural networks are widely used in financial time series problem and are considered suitable due to the high dimensionality of the data. Specifically, LSTMs are most appropriate for stock price prediction due to their capability of temporal memory. This mini-project is about analyzing the historical data of the stocks & predict the daily closing prices of a stock. The forecast horizon is for 30 days, the historical data used for the training is volumes for a period of five years and one year for the testing. It includes an pre-processing of the data as it is a fundamental step before experimenting on model architectures and methods. The proposed study utilizes the carefully selected features from fundamental, macroeconomic, and technical data to build the model. After that, the collected data has been normalized using the min–max normalization technique. Then input sequence for the LSTM model is created using a specific time step. The hyperparameters such as number of neurons, epochs, learning rate, batch size, and time step have been incorporated in the model. The [regularization](https://www.sciencedirect.com/topics/computer-science/regularization) techniques have been utilized to overcome the over- fitting problems. Once the hyperparameters are tuned, the input data is fed into the LSTM model to predict the closing price of the stock market index. The quality of the proposed model is assessed through RMSE and R2-Score.

* 1. **Motivation**

Motivation of this project is to make the stock prediction model using Machine Learning algorithm. Stocks are very volatile; this complex nature of stock prices is a significant attraction for researcher and statisticians to find a way to predict them. Despite the numerous amount of research publications in this field, there are still many that claim that stock markets cannot be predicted. This is primarily because of the number of factors that affect stocks prices and those factors themselves depend on some other, potentially unknown factors. This is the motivation to work on this topic.

* 1. **Objective**

The objective of this project is to predict the future price of the stocks using special kind of Recurrent Neural Network known as LSTM. Based on the features like stocks high, low and adjusted close value of a particular company I can predict the stock values and it is evaluated by the metrics such as R2-Score.

# 2.ANALYSIS AND DESIGN

* 1. **Architecture and Techniques**

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell remembers values over arbitrary time intervals and the three gates regulate the flow of information into and out of the cell. The general architecture of LSTM is shown in Figure.2.1.1

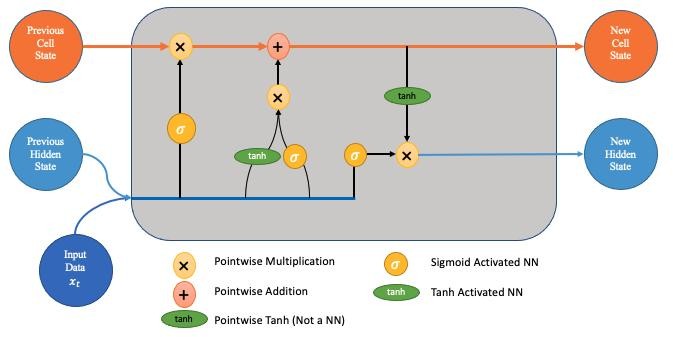


Figure 2.1.1 General Architecture of LSTM

Long Short-Term Memory (LSTM) is one of many types of Recurrent Neural Network RNN, it’s also capable of catching data from past stages and use it for future predictions . In general, an Artificial Neural Network (ANN) consists of three layers:

* + 1. Input layer,
    2. Hidden layers,
    3. Output layer.

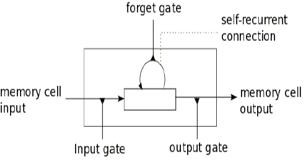


Figure 2.1.2 LSTM Cell

In a NN that only contains one hidden layer the number of nodes in the input layer always depend on the dimension of the data, the nodes of the input layer connect to the hidden layer via links called ‘synapses’. The relation between every two nodes from (input to the hidden layer), has a coefficient called Weight, which is the decision maker for signals. The process of learning is naturally a continues adjustment of weights, after completing the process of learning, the Artificial NN will have optimal weights for each synapses. The hidden layer nodes Apply a sigmoid or tangent hyperbolic (tanh) function on the sum of weights coming from the input layer which is called the activation function, this transformation will generate values, with a minimized error rate between the train and test data using the adam function.

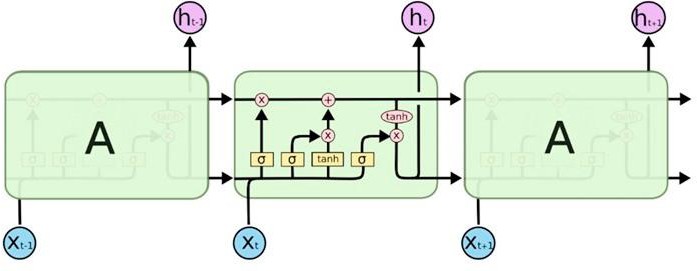
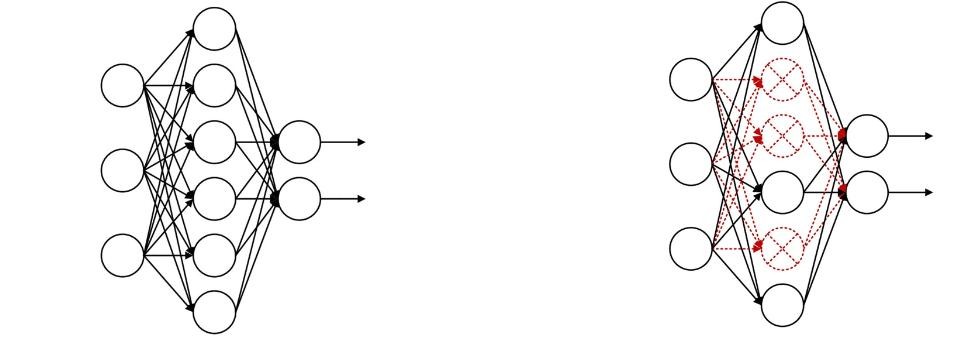


Figure 2.1.3 The internal structure of an LSTM

**2.2 Deep recurrent neural network**



A LSTM based deep recurrent neural network (DRNN) is a variant of the recurrent neural network. To enhance the expressive power of the model, the loop body at each moment can be repeated many times, the structure diagram of deep recurrent neural network is given. Deep recurrent neural network is composed of LSTM, so its operation mechanism is same as LSTM. During the process of constructing the task model, the dropout method was used. Dropout refers to the temporary discarding of the neural network unit from the network according to a certain probability during the training of the deep learning network, which is a means to prevent over-fitting. The principle of dropout operation is that the neurons in each layer are randomly deleted with probability P in a training iteration, and the data in this iteration are trained with the network composed of the remaining (1 − p)\*N neurons, thus alleviating the over- fitting problem.

Figure 2.2.1 The neural network model without and with dropout

The values obtained after this transformation constitute the output layer of our NN, these value may not be the best output, in this case a back propagation process will be Applied to target the optimal value of error, the back propagation process connect the output layer to the hidden layer, sending a signal conforming the best Light with the optimal error for the number of epochs decided. This process will be repeated trying to improve our predictions and minimize the prediction error.

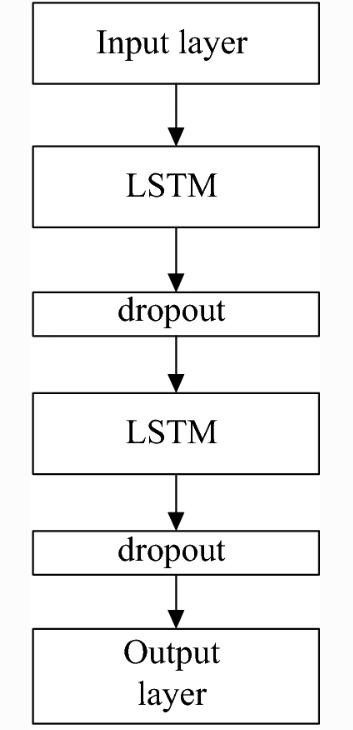


Figure 2.2.2 Deep recurrent neural network based on LSTM

Deep recurrent neural network is composed of LSTM, so its operation mechanism is

same as LSTM. During the process of constructing the task model, the dropout

method was used. Dropout refers to the temporary discarding of the neural network

unit from the network according to a certain probability during the training of the

deep learning network, which is a means to prevent over-fitting. The principle of

dropout operation is that the neurons in each layer are randomly deleted with

probability P in a training iteration, and the data in this iteration are trained with the

network composed of the remaining (1 − p)\*N neurons, thus alleviating the over-

fitting problem. The neural network model without dropout is shown in Figure 2.2.3 is the

neural network model with dropout. The LSTM based deep recurrent neural network

model with dropout layer was used as the contrast model to verify the feasibility and

Applicability of the proposed associated neural network model.

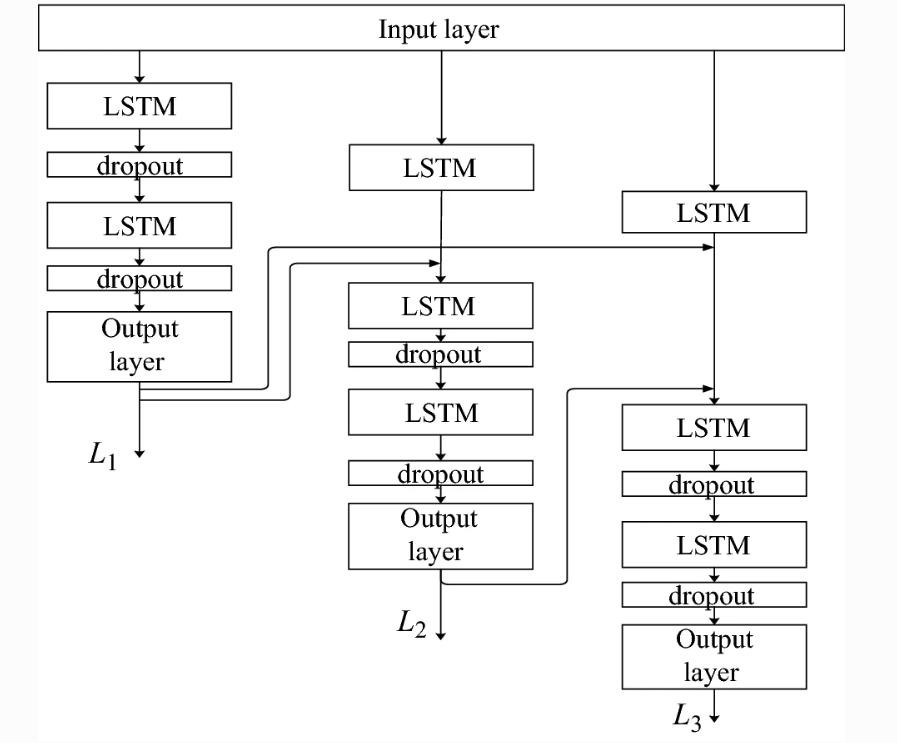


Figure 2.2.3 A structural model of associated net

Figure 6: A structural model of associated net

The specific data processing flow of the multi-value associated neural iiiiiiisonetwork

model. Data through the input layer to all three branches simultaneously. Three is a

branches predict the opening price, the lowest price and the highest price respectively.

In the Chinese stock market, the maximum fluctuation of stock price is only 10%.

Therefor the model recombines the output of the left branch (opening price) and the

output of the LSTM network of the second branch as the input parameter data of the

predicted lowest price, and the highest price is subject to the opening price of the day,

the impact of the lowest price, so the output of the left branch (opening price) and the

output of the intermediate branch (lowest price) and the output of the LSTM network

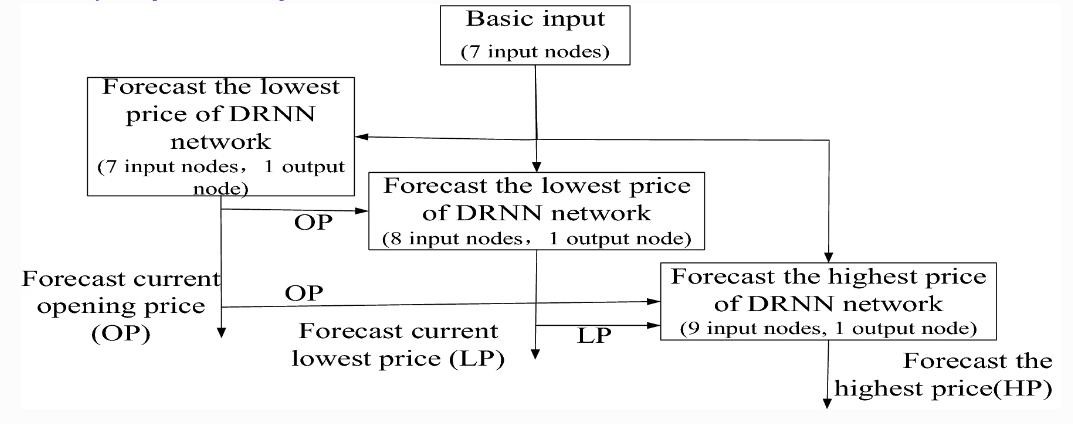


Figure 2.2.4 The data processing flow chart of associated net

**2.3 Structure Chart:**

A structure chart in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name. Structure for this prediction is given in Figure 2.3

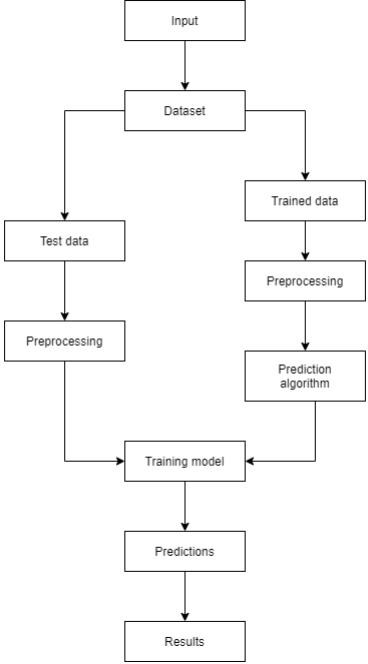


Figure 2.3 Structured Diagram

**2.4 Tools**

To construct a classification algorithm I need the machine learning tools like scikit- learn, tensorflow etc. Tools are defined as algorithmic uses of artificial intelligence, that enable the system to understand and improve without being clearly programmed because these tools are capable of performing complex processing tasks such as image awareness, speech text, generating natural languages, and so on. Let us see all the tools that have been used in this overall algorithm.

* **Pandas** - Pandas is an open-source setup for a python programming language and a python library licensed by which offers high-performance data analysis tools and easy-to-use data structures for the Python programming language.
* **NumPy** - Numerical Python (NumPy) is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* **Matplotlib** - Matplotlib is easy to use and an amazing visualizing library in Python. It is built on NumPy arrays and designed to work with the broader SciPy stack and consists of several plots like line, bar, scatter, histogram, etc.
* **Scikit-learn** - (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statis–tical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python. This library, which is largely written in Python, is built upon NumPy, SciPy and Matplotlib.

## 2.5 Flow and Sequence Diagram

**Flow Chart**

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task. The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. Flow chart is given in Figure 2.5.1

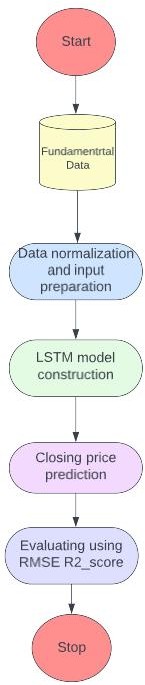


Figure 2.5.1 Flow Chart for Stock Prediction

As outlined in the diagram, the proposed study utilizes the carefully selected features from fundamental data to build the model. After that, the collected data has been normalized using the min–max normalization technique. Then input sequence for the LSTM model is created using a specific time step. The hyperparameters such as number of neurons, epochs, learning rate, batch size, and time step have been incorporated in the model. The [regularization](https://www.sciencedirect.com/topics/computer-science/regularization) techniques have been utilized to overcome the over-fitting problems. Once the hyperparameters are tuned, the input data is fed into the LSTM model to predict the closing price of the stock market index. The quality of the proposed model is assessed through RMSE, MAPE, and R2\_Score

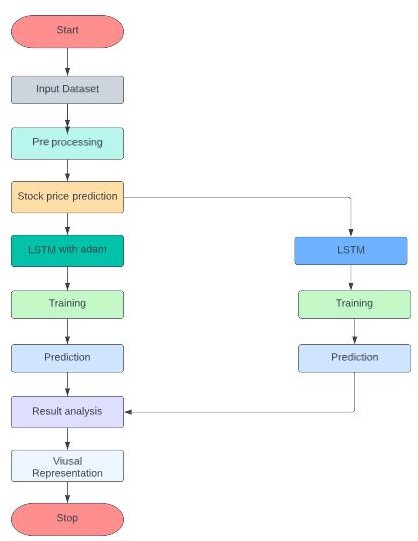


Figure 2.5.2 Detailed Flow Chart Stock Prediction

**2.6 Sequence Diagram**

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

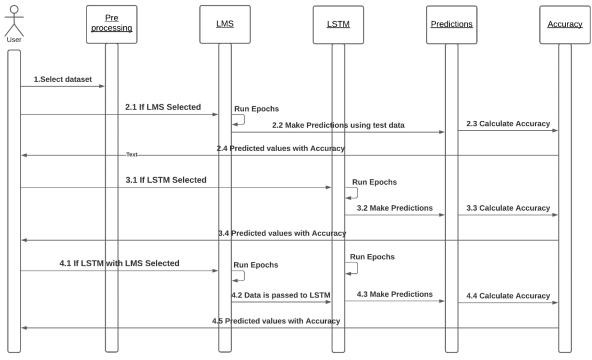


Figure 2.6 Sequence Diagram

**2.7 Activity Diagram**

An activity diagram is a behavioral diagram it depicts the behavior of a system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

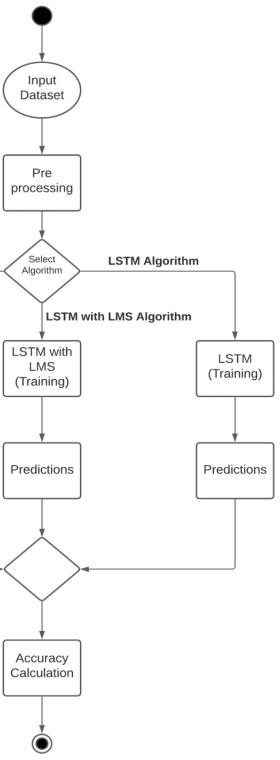


Figure. 2.7 Execution based on algorithm selection

**2.8 Component Diagram**

Component diagram is a special kind of diagram in UML. The purpose is also different from all other diagrams discussed so far. It does not describe the functionality of the system but it describes the components used to make those functionalities. Component diagrams are used in modeling the physical aspects of object-oriented systems that are used for visualizing, specifying, and documenting component-based systems and also for constructing executable systems through forward and reverse engineering. Component diagrams are essentially class diagrams that focus on a system's components that often used to model the static implementation view of a system.

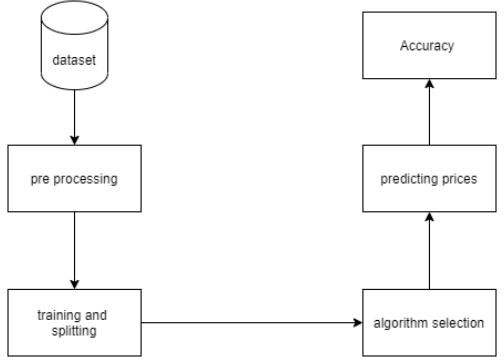


Figure 2.8 Components present in the system

# 3. IMPLEMENTATION

## Overview of the project

This mini-project is implemented using python on Google Collab environment with required python modules like

* + - Pandas
    - Numpy
    - Tensorflow
    - Matplotlib
    - TimeSeriesSplit
    - Keras
    - Minmaxscaler

Microsoft Corporation stock dataset is used for analyzing and predict the value,mnist.csv is the data file used for this report. First the python libraries and the dataset file are imported in the project .Using the Pandas, I will upload the stock data from the local system as a Comma Separated Value file and save it to a pandas Data Frame.

The sci-kit-learn library’s Time Series Split class will be used for this. I set the number of splits to 10, indicating that 10% of the data will be used as the test set and 90% of the data would be used to train the LSTM model. Once the training and test sets are finalized, I will input the data into the LSTM model and then I use the fit function to train the LSTM model created above on the training data. Finally, I can use it to forecast the Adjacent Close Value of the Microsoft stock by using a model trained using the LSTM network on the test set. Comparing the actual predicted stock value is the conclusion of the project.



Figure 3.1.1 Flow Chart

## Data Preparation

The dataset which is used for the prediction is MSFT dataset and APPLE dataset. Both the datasets contains columns associated with time series like the date and the different variables like close, high, low and volume. I will use opening and closing values for our experimentation of time series with LSTM.

## 3.2.1 Pre-processing of the input dataset

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. In this project a real-world dataset in collected for prediction. So, a real-world dataset contains noises, null values, and some unwanted data’s that cannot be used in machine learning algorithm. So, in order to remove the noises and unwanted data I use Data Pre-processing to clean the data. Then the missing values are replaced by specific values. Only required data columns and rows are taken, the rest are removed .Then the dataset is splitted into training set and testing set for the further process.



Figure 3.2.1 Pre-Processing of data

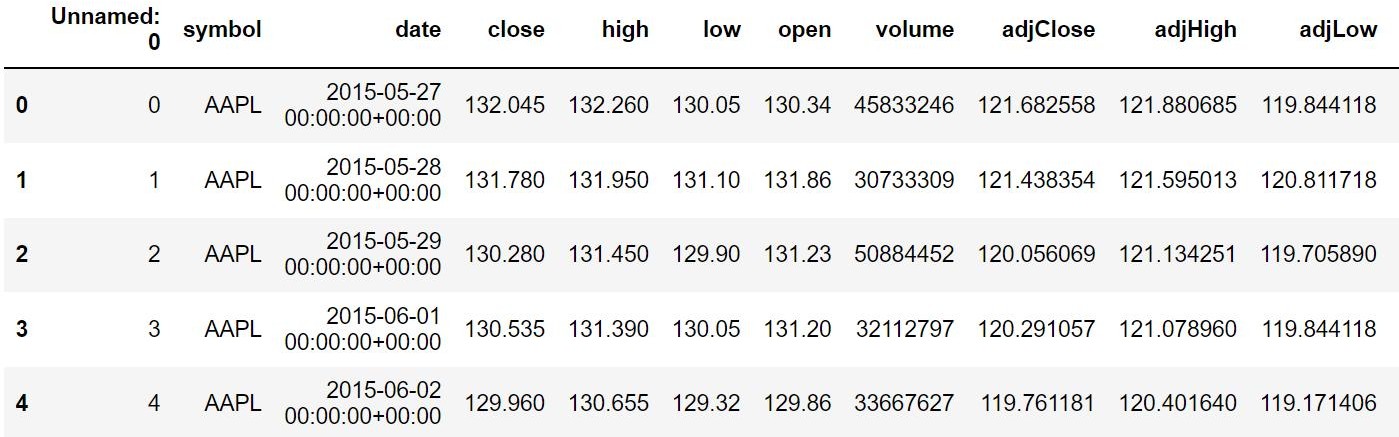


Figure 3.2.2 Dataset APPLE

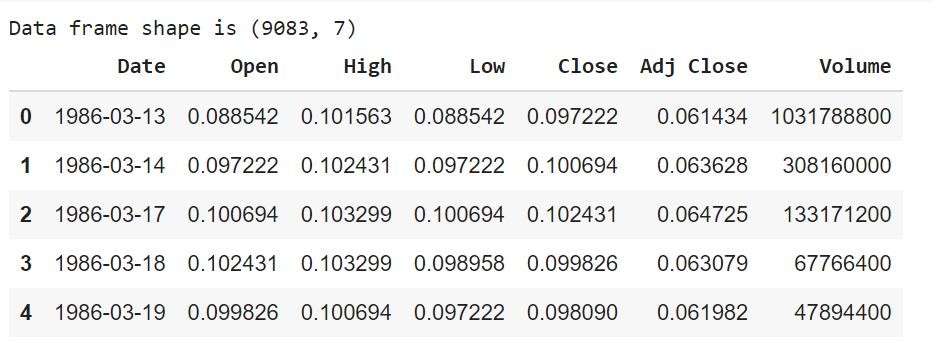


Figure 3.2.3 Dataset MSFT

However I are not going to use all the features for training the model. The actual features that are selected for training the model is open, high, low and closing value of the stocks and the target variable is adjusted close value. Next I have to preprocess the data. Data preprocessing in Machine Learning refers to the technique of preparing cleaning and organizing the raw data to make it suitable for a building and training Machine Learning models.

**3.2.4 Mean Value:**

The Mean value of a dataset is the average value i.e. a number around which a whole data is spread out. All values used in calculating the average are weighted equally when defining the Mean. Mean value of the MSFT and Apple dataset are given below:

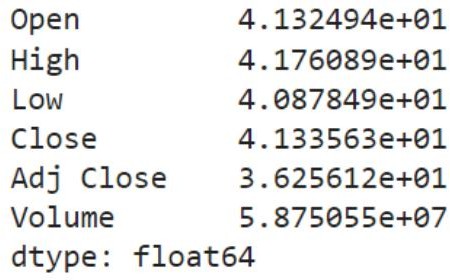


Figure 3.2.4 Mean value of MSFT

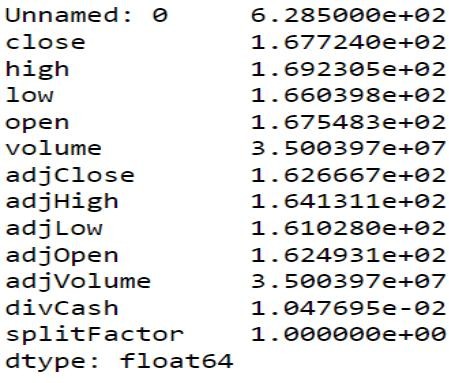


Figure 3.2.5 Mean value of Apple

* 1. **Visualizing data**

**3.3.1 Why visualization is important?**

Data visualization is the presentation of data in a pictorial or graphical format. The data

visualization is useful for data cleaning, exploring data structure, detecting outliers and unusual groups, identifying trends and clusters, spotting local patterns, evaluating modeling output, and presenting results. Data Visualization can be used to interact with the users in a way that textual table-based data cannot. After all, data visualization tells a story from a particular angle to the users, and that by default makes it interactive. Users can also focus on aspects of the data visualization that they find particularly interesting and then they can learn more about that aspect of the data. The data visualization also allows the user to obtain a holistic view of the data using different types of charts and ample usage of colors, shapes, etc. This interactivity also allows the viewers to understand the data visualization at a single glance which is rather difficult for

This is the visualizing of volume traded every day or sales volume of both Apple and MSFT dataset. It is shown in the Figure.3.3

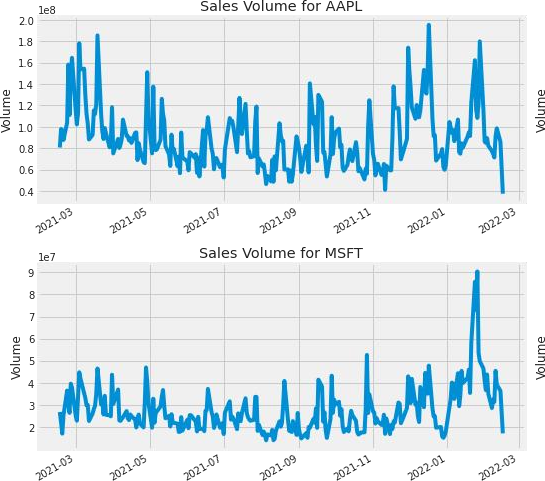


Figure 3.3.1 Volume of the stocks

From the extracted data I am now visualizing the closing value on the graph. This is the actual trend of the closing prices of the stock for Apple data set. It is shown in the Figure.3.3.2 and 3.3.3

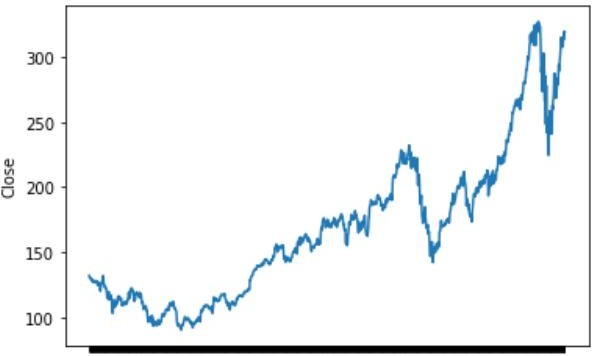


Figure 3.3.2 Closing price of APPLEE

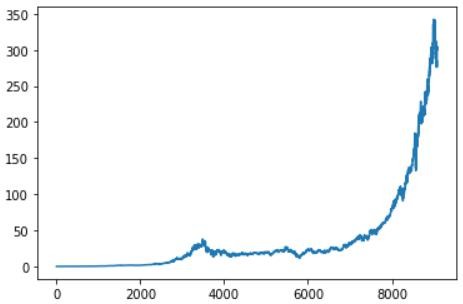
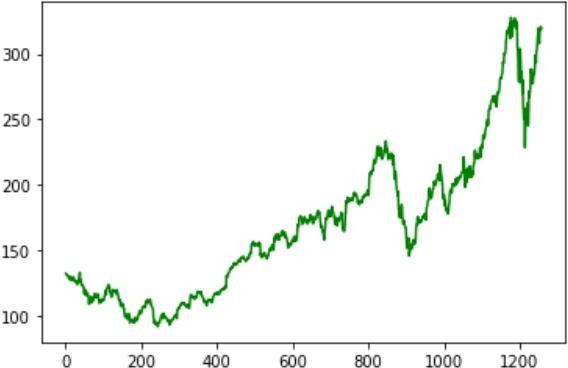


Figure 3.3.3 Closing price of MSFT

Next I have to visualize the stocks high value on the particular dates of both Apple and MSFT dataset.



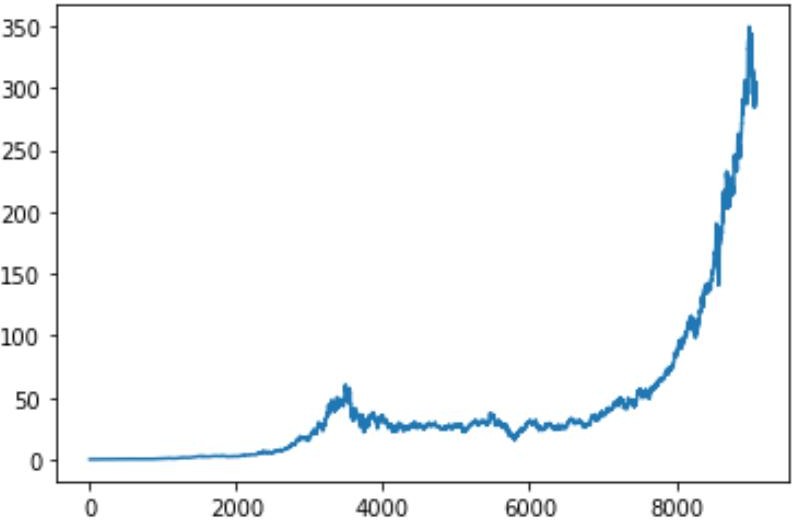
Figure 3.3.4 High price stocks of Apple

Figure 3.3.5 High price stocks of MSFT

This is some of the sample adjusted close value on previous dates using the Apple dataset .Adj close value is the target variable for the LSTM model shown in Figure.3.3.6

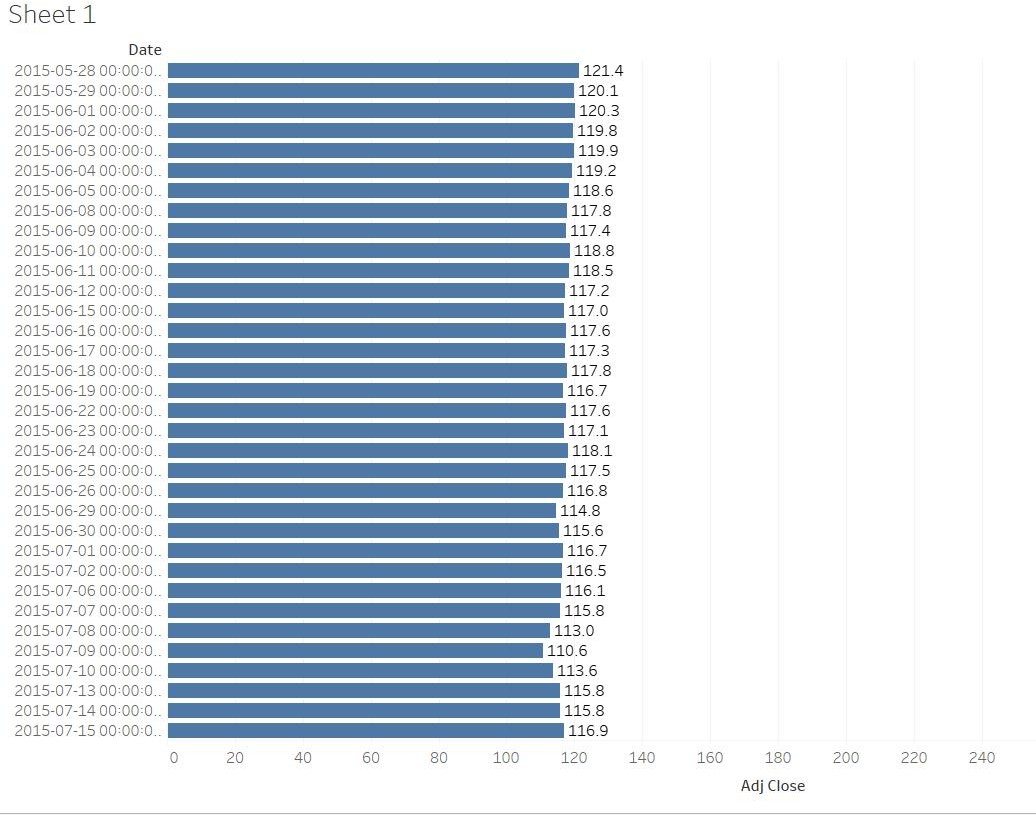


Figure 3.3.6 Bar chart of Adj close value

**3.4 Heatmap**

A heat map is a two-dimensional representation of data in which values are represented by colors. A simple heat map provides an immediate visual summary of information. More elaborate heat maps allow the viewer to understand complex data sets. It provides at a glance visual representation of market data. View the performance of different market slices, sectors, asset classes, or individual stocks.

Compares the stock market of a specific country to that of another nation.



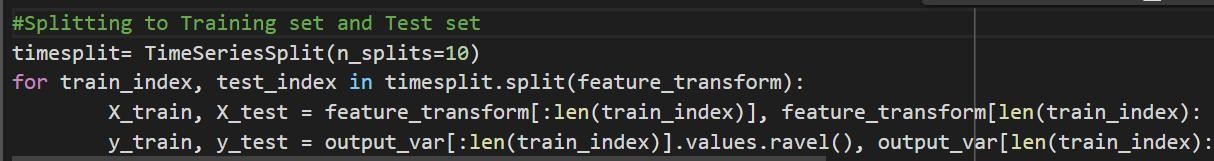
Figure 3.4.1 Heatmap of APPLEe dataset

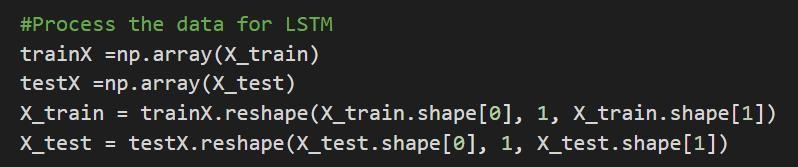


Figure 3.4.2 Heatmap of MSFT dataset

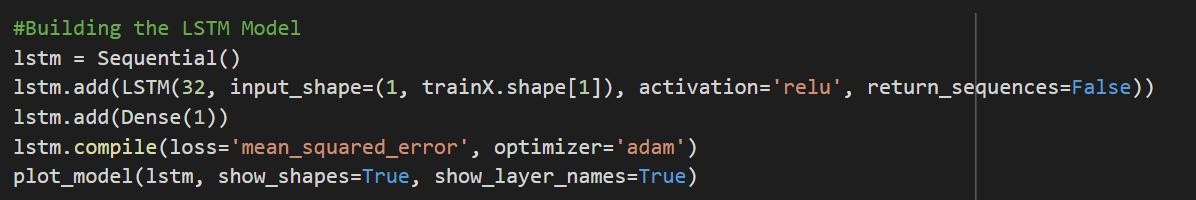
## Training and Testing

Since the data is time series, I will take the first 90% of data as training and the last 10% as test data. Using the Time Series Split I am splitting it into two divisions. A function is created so that I can create the sequence for training and testing.



In the next step, I create our LSTM model. In this article, I will use the Sequential model imported from Keras and required libraries are imported.

I use two LSTM layers in our model and implement drop out in between for regularization. The number of units assigned in the LSTM parameter is fifty. with a dropout of 10 %. Mean squared error is the loss function for optimizing the problem with adam optimizer. Mean absolute error is the metric used in our LSTM network as it is associated with time-series data



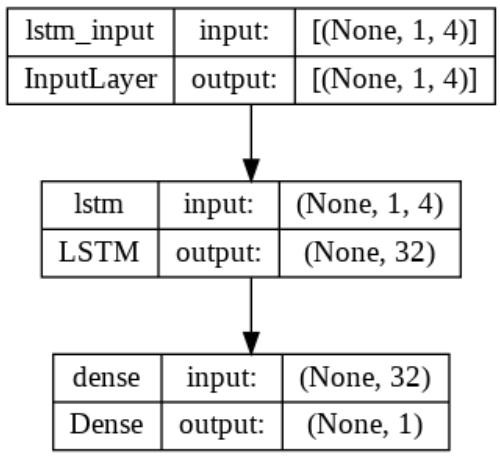


Figure 3.5.1 LSTM Model

Dropout layers are used to avoid overfitting issues, and the option “return sequences” is used to decide if the layer will return a sequence compatible with an LSTM.

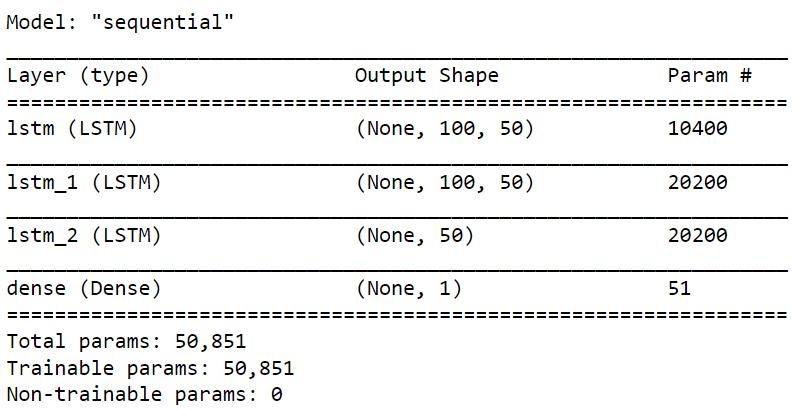
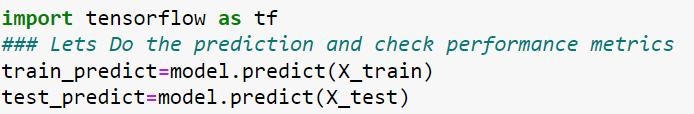


Figure 3.5.2 Model Summary

# 4.RESULT ANALYSIS

* 1. **Model Testing**

After fitting the data with our model I use it for prediction. I have used inverse transformation to get back the original value with the transformed function. Now I can use this data to visualize the prediction. However, before plotting our predictions, I must first APPLE an inverse transform () to the predictions array, because I use the Scale to generate predictions, and hence our predictions are between 0 and 1.



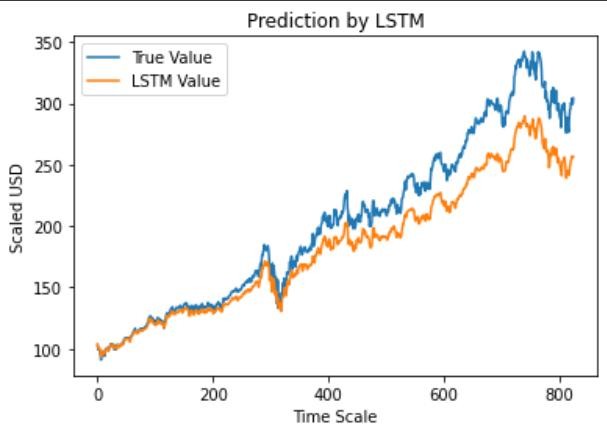


Figure 4.1.1 Prediction of MSFT(Actual vs Predicted)

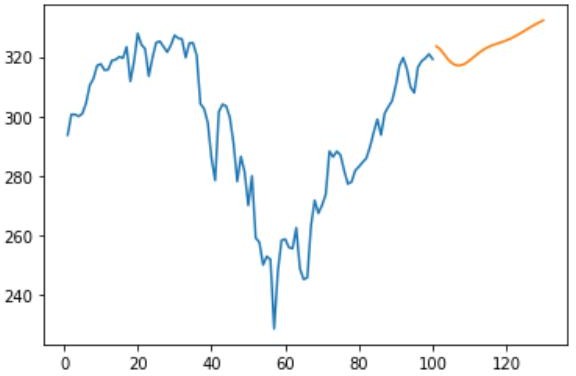


Figure 4.1.2 30 days forecasting of APPLE dataset

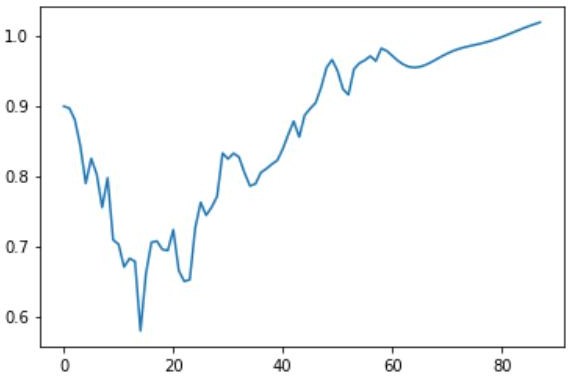


Figure 4.1.3 Forecasting output

## Evaluation of the model

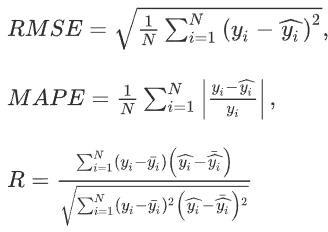
We implement single layer and multilayer LSTM architecture to predict the closing price. Within each of these models, several options are considered with different number of neurons. Prediction accuracy and reliability of these models are assessed by calculating three different performance metrics —RMSE, MAPE, and R. The analytical form of these metrics are defined as follows:

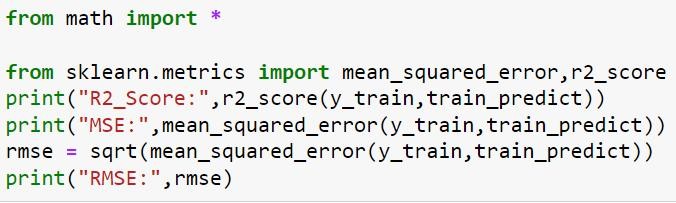
where,

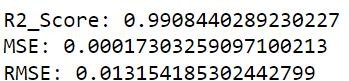
yi:Original time series,

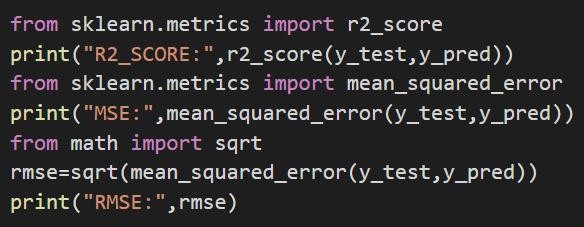
yi: The average value of the original time series, yiˆ: Predicted time series computed from the model, yiˆ̄: Average value of the predicted time series,

N: Number of observations







 Figure 4.2.1 Evaluation of APPLEe dataset

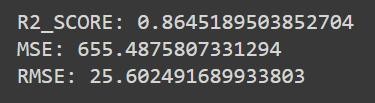


Figure 4.2.2 Evaluation of MSFT dataset

## 4.3 Verification

The model is tested and verified successfully. The forecasted Apple dataset gets an 99% of accuracy when compared with the actual value and the MSFT dataset gets 86% accuracy.

**SOURCE CODE**

import pandas as pd

df=pd.read\_csv('AAPL.csv')

df.head()

corr=df.corr()

corr.style.background\_gradient(cmap ='summer')

df1=df.reset\_index()['high']

df1

import matplotlib.pyplot as plt

plt.plot(df1,color='g')#,marker='o',markerfacecolor='R')

df2=df['close']

plt.plot(df2)

from sklearn.preprocessing import MinMaxScaler

import numpy as np

scaler=MinMaxScaler(feature\_range=(0,1))

df1=scaler.fit\_transform(np.array(df1).reshape(-1,1))

import numpy as np

df1

training\_size=int(len(df1)\*0.65)

test\_size=len(df1)-training\_size

train\_data,test\_data=df1[0:training\_size,:],df1[training\_size:len(df1),:1]

training\_size,test\_size

##splitting dataset into train and test split

training\_size=int(len(df1)\*0.65)

test\_size=len(df1)-training\_size

train\_data,test\_data=df1[0:training\_size,:],df1[training\_size:len(df1),:1]

import numpy

# convert an array of values into a dataset matrix

def create\_dataset(dataset, time\_step=1):

dataX, dataY = [], []

for i in range(len(dataset)-time\_step-1):

a = dataset[i:(i+time\_step), 0] ###i=0, X=0,1,2,3-----99 Y=100

dataX.append(a)

dataY.append(dataset[i + time\_step, 0])

return numpy.array(dataX), numpy.array(dataY)

time\_step = 100

X\_train, y\_train = create\_dataset(train\_data, time\_step)

X\_test, ytest = create\_dataset(test\_data, time\_step)

print(X\_train.shape), print(y\_train.shape)

# reshape input to be [samples, time steps, features] which is required for LSTM

X\_train =X\_train.reshape(X\_train.shape[0],X\_train.shape[1] , 1)

X\_test = X\_test.reshape(X\_test.shape[0],X\_test.shape[1] , 1)

### Create the Stacked LSTM model

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from tensorflow.keras.layers import LSTM

model=Sequential()

model.add(LSTM(50,return\_sequences=True,input\_shape=(100,1)))

model.add(LSTM(50,return\_sequences=True))

model.add(LSTM(50))

model.add(Dense(1))

model.compile(loss='mean\_squared\_error',optimizer='adam')

import tensorflow as tf

### Lets Do the prediction and check performance metrics

train\_predict=model.predict(X\_train)

test\_predict=model.predict(X\_test)

from math import \*

from sklearn.metrics import mean\_squared\_error,r2\_score

print("R2\_Score:",r2\_score(y\_train,train\_predict))

print("MSE:",mean\_squared\_error(y\_train,train\_predict))

rmse = sqrt(mean\_squared\_error(y\_train,train\_predict))

print("RMSE:",rmse)

### Plotting

# shift train predictions for plotting

look\_back=100

trainPredictPlot = numpy.empty\_like(df1)

trainPredictPlot[:, :] = np.nan

trainPredictPlot[look\_back:len(train\_predict)+look\_back, :] = train\_predict

# shift test predictions for plotting

testPredictPlot = numpy.empty\_like(df1)

testPredictPlot[:, :] = numpy.nan

testPredictPlot[len(train\_predict)+(look\_back\*2)+1:len(df1)-1, :] = test\_predict

# plot baseline and predictions

plt.plot(scaler.inverse\_transform(df1))

plt.plot(trainPredictPlot)

plt.plot(testPredictPlot)

plt.show()

# demonstrate prediction for next 30 days

from numpy import array

lst\_output=[]

n\_steps=100

i=0

while(i<30):

if(len(temp\_input)>100):

#print(temp\_input)

x\_input=np.array(temp\_input[1:])

print("{} day input {}".format(i,x\_input))

x\_input=x\_input.reshape(1,-1)

x\_input = x\_input.reshape((1, n\_steps, 1))

#print(x\_input)

yhat = model.predict(x\_input, verbose=0)

print("{} day output {}".format(i,yhat))

temp\_input.extend(yhat[0].tolist())

temp\_input=temp\_input[1:]

#print(temp\_input)

lst\_output.extend(yhat.tolist())

i=i+1

else:

x\_input = x\_input.reshape((1, n\_steps,1))

yhat = model.predict(x\_input, verbose=0)

print(yhat[0])

temp\_input.extend(yhat[0].tolist())

print(len(temp\_input))

lst\_output.extend(yhat.tolist())

i=i+1

print(lst\_output)

## 5.CONCLUSION

With the introduction of Machine Learning and its strong algorithms, the most recent market research and Stock Market Prediction advancements have begun to include such approaches in analyzing stock market data. This project proposes RNN based on LSTM built to forecast future values for both MSFT and APPLE assets, the result of our model has shown some promising result. The testing result conform that our model is capable of tracing the evolution of opening prices for both assets. For our future work we will try to find the best sets for bout data length and number of training epochs that better suit our assets and maximize our predictions accuracy. The machine learning models which is implemented and has a greater accuracy is taken for the user defined data prediction. This will give us nearly accurate result which will be helpful in prediction of the stock values.

## 6.REFERENCES

[1] Batres-Estrada, B. (2015). Deep learning for multivariate financial time series.

[2 ]Emerson, S., Kennedy, R., O'Shea, L., & O'Brien, J. (2019, May). Trends and APPLEications of Machine Learning in Quantitative Finance. In 8th International Conference on Economics and Finance Research (ICEFR 2019).

[3] Heaton, J. B., Polson, N. G., & Witte, J. H. (2017). Deep learning for finance: deep portfolios. APPLEied Stochastic Models in Business and Industry, 33(1), 3-12.

[4] Moritz, B., & Zimmermann, T. (2016). Tree-based conditional portfolio sorts: The relation between past and future stock returns. Available at SSRN 2740751.

[5] Olah, C. (2015). Understanding lstm networks–colah’s blog. Colah. github. io.

[6] Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2018). Decision-Making for Financial Trading: A Fusion Approach of Machine Learning and Portfolio Selection. Expert Systems with APPLEications.

[7] Patterson J., 2017. Deep Learning: A Practitioner’s Approach, O’Reilly Media.

[8] Siami-Namini, S., & Namin, A. S. (2018). Forecasting economics and financial time series: Arima vs. lstm. arXiv preprint arXiv:1803.