

DEEP LEARNING

(Stock Price Prediction)

Summer Internship Report Submitted in partial fulfillment
of the requirement for undergraduate degree of

Bachelor of Technology
in
Computer Science Engineering
By

Nalla Mallika Reddy

221710313039

Under the Guidance of



Department of Computer Science and Engineering

GITAM School of Technology

GITAM(Deemed to be University)

Hyderabad-502329

July 2020

DECLARATION

I submit this industrial training work entitled “STOCK PRICE PREDICTION” to GITAM(Deemed To Be University), Hyderabad in partial fulfillment of the requirements for the award of the degree of “Bachelor of Technology ” in “ Computer Science Engineering”. I declare that it was carried out independently by me under the guidance of _____, _____, GITAM(Deemed to be University),Hyderabad,India.

The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

Place: HYDERABAD

Nalla Mallika Reddy

Date:

221710313039



GITAM (DEEMED TO BE UNIVERSITY)

Hyderabad-502329, India

Dated:

CERTIFICATE

This is to certify that the Industrial Training Report entitled “**Stock Price Prediction**” is being submitted by **Nalla Mallika Reddy(221710313039)** in partial fulfillment of the requirement for the award of **Bachelor of Technology in Computer Science Engineering** at GITAM (Deemed to Be University), Hyderabad during the academic year 2020-21

It is faithful record work carried out by her at the **Computer Science Engineering Department**, GITAM University Hyderabad campus under my guidance and supervision.

Mr.
Assistant Professor
Department of CSE

Dr.
Professor and HOD
Department of CSE

ACKNOWLEDGEMENT

Apart from my effort, the success of this internship largely depends on the encouragement and guidance of many others. I take this opportunity to express my gratitude to the people who have helped me in the successful competition of this internship.

I would like to thank Dr. N. Siva Prasad, Pro Vice Chancellor, GITAM Hyderabad and Prof. N. Seetha Ramaiah, Principal, GITAM Hyderabad.

I would like to thank respected Prof. S. Phani Kumar, Head of the Department of Computer Science Engineering for giving me such a wonderful opportunity to expand my knowledge for my own branch and giving me guidelines to present the internship report. It helped me a lot to realize what we study for.

I would like to thank the respected faculties _____ who helped me to make this internship a successful accomplishment. I would also like to thank my friends who helped me to make my work more organized and well-stacked till the end.

Nalla Mallika Reddy

221710313039

ABSTRACT

Stock market has received widespread attention from investors. It has always been a hot spot for investors and investment companies to grasp the change regularity of the stock market and predict its trend. Currently, there are many methods for stock price prediction. The prediction methods can be roughly divided into two categories: statistical methods and artificial intelligence methods. Statistical methods include logistic regression model, ARCH model, etc. Artificial intelligence methods include multi-layer perceptron, convolutional neural network, naive Bayes network, back propagation network, single-layer LSTM, support vector machine, recurrent neural network, etc. But these studies predict only one single value. In order to predict multiple values in one model, it need to design a model which can handle multiple inputs and produces multiple associated output values at the same time. For this purpose, it is proposed an associated deep recurrent neural network model with multiple inputs and multiple outputs based on long short-term memory network. The associated network model can predict the opening price, the lowest price and the highest price of a stock simultaneously. The associated network model was compared with LSTM network model and deep recurrent neural network model. The experiments show that the accuracy of the associated model is superior to the other two models in predicting multiple values at the same time, and its prediction accuracy is over 95%

TABLE OF CONTENTS

CHAPTER 1.....	1
MACHINE LEARNING.....	1
1.1 INFORMATION ABOUT MACHINE LEARNING :.....	1
1.2 USES OF MACHINE LEARNING:.....	2
1.3 TYPES OF LEARNING ALGORITHMS:.....	3
1.3.1 Supervised Learning:.....	3
1.3.2 Unsupervised Learning:.....	3
1.3.3 Semi Supervised Learning:.....	4
CHAPTER 2.....	6
DEEP LEARNING.....	6
2.1 INFORMATION ABOUT DEEP LEARNING:.....	6
2.2 USES OF DEEP LEARNING:.....	6
2.3 TYPES OF LEARNING ALGORITHMS:.....	7
2.3.1 Convolutional Neural Network:.....	7
2.3.2 Recurrent Neural Network(RNN):.....	8
2.3.3 Long short-term memory (LSTM):.....	9
CHAPTER 3.....	10
PYTHON.....	10
3.1 INTRODUCTION TO PYTHON:.....	10
3.2 HISTORY OF PYTHON:.....	10
3.3 FEATURES OF PYTHON:.....	11
3.4 HOW TO SETUP PYTHON:.....	11
3.4.1 Installation(using python IDLE):.....	12
3.4.2 Installation(using Anaconda):.....	13
3.5 PYTHON VARIABLE TYPES:.....	14
3.6 PYTHON FUNCTION:.....	17
3.6.1 Defining a Function:.....	17

3.6.2 Calling a Function:.....	17
3.7 PYTHON USING OOPs CONCEPTS:.....	18
3.7.1 Class:.....	19
3.7.2 __init__ method in Class:.....	19
CHAPTER 4.....	20
CASE STUDY.....	20
4.1 PROBLEM STATEMENT:.....	20
4.2 DATA SET:.....	20
4.3 OBJECTIVE OF THE CASE STUDY:.....	21
CHAPTER 5.....	22
DATA PREPROCESSING.....	22
5.1 READING THE DATASET:.....	22
5.2 IMPORTING THE LIBRARIES:.....	22
5.3 VERSIONS OF PACKAGES:.....	23
5.4 IMPORTING THE DATA-SET:.....	23
5.5 HANDLING MISSING VALUES:.....	24
CHAPTER 6.....	25
FEATURE SELECTION.....	25
6.1 SELECTING A PARTICULAR COMPANY:.....	25
6.1.1 ABOUT COMPANY:.....	25
6.1.2 CHECK RAW DATA:.....	26
6.2 SELCTING RELEVANT FEATURE FOR ANALYSIS:.....	27
6.3 SPLITTING THE DATA INTO TRAIN AND TEST:.....	28
6.4 SCALING THE DATA:.....	30
CHAPTER 7.....	31
MODEL BUILDING.....	31
7.1 INTRODUCTION TO LSTM:.....	31
7.1.1 STRUCTURE Of LSTM:.....	31
7.2 BUILDING LSTM MODEL:.....	34

7.3 COMPILE THE MODEL:.....	35
7.4 EVALUATE THE MODEL:.....	37
7.5 COMPARE THE RESULT:.....	39
CONCLUSION:.....	40
REFERENCES:.....	40

LIST OF FIGURES

Figure 1.1.1	Process flow.....	2
Figure 1.3.2.1	Unsupervised Learning.....	4
Figure 1.3.3.1	Semi Supervised Learning.....	5
Figure 2.3.1.1	CNN.....	6
Figure 2.3.2.1	RNN.....	7
Figure 3.4.1.	Python download.....	12
Figure 3.4.2.1	Anoconda download.....	13
Figure 3.4.2.2	Jupyter notebook.....	19
Figure 3.7.1.1	Defining a class.....	20
Figure 5.2.1	Importing the required packages.....	22
Figure 5.3.1	Versions of packages.....	23
Figure 5.4.1	Reading the dataset.....	23
Figure 5.5.1	Handling missing values.....	24
Figure 6.1.1	Selecting a particular company.....	25
Figure 6.1.2	Checking stock price for unknown values.....	26
Figure 6.2.1	Selecting relevant features.....	27
Figure 6.3.1	Train Test Split.....	29
Figure 6.4.1	Scaling the data.....	30
Figure 7.1.1.1	Structure of LSTM.....	31
Figure 7.1.1.2	Forget gate.....	32
Figure 7.1.1.3	Input gate.....	32
Figure 7.1.1.4	Output gate.....	33
Figure 7.2.1	Building Lstm model.....	34

Figure 7.3.1	Compiling the model.....	35
Figure 7.3.1.1	Fitting the data.....	36
Figure 7.3.1.2	Visualizing the loss values.....	37
Figure 7.4.1	Evaluating the model.....	38
Figure 7.5.1	Predicted and actual values.....	39

CHAPTER 1

MACHINE LEARNING

1. INFORMATION ABOUT MACHINE LEARNING :

Machine Learning(ML) is the scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of Artificial Intelligence(AI).

1.1 IMPORTANCE OF MACHINE LEARNING:

Consider some of the instances where machine learning is applied: the self-driving Google car, cyber fraud detection, online recommendation engines—like friend suggestions on Facebook, Netflix showcasing the movies and shows you might like, and “more items to consider” and “get yourself a little something” on Amazon—are all examples of applied machine learning. All these examples echo the vital role machine learning has begun to take in today’s data-rich world.

Machines can aid in filtering useful pieces of information that help in major advancements, and we are already seeing how this technology is being implemented in a wide variety of industries.

With the constant evolution of the field, there has been a subsequent rise in the uses, demands, and importance of machine learning. Big data has become quite a buzzword in the last few years; that’s in part due to increased sophistication of machine learning, which helps analyze those big chunks of big data. Machine learning has also changed the way data extraction, and interpretation is done by involving automatic sets of generic methods that have replaced traditional statistical techniques.

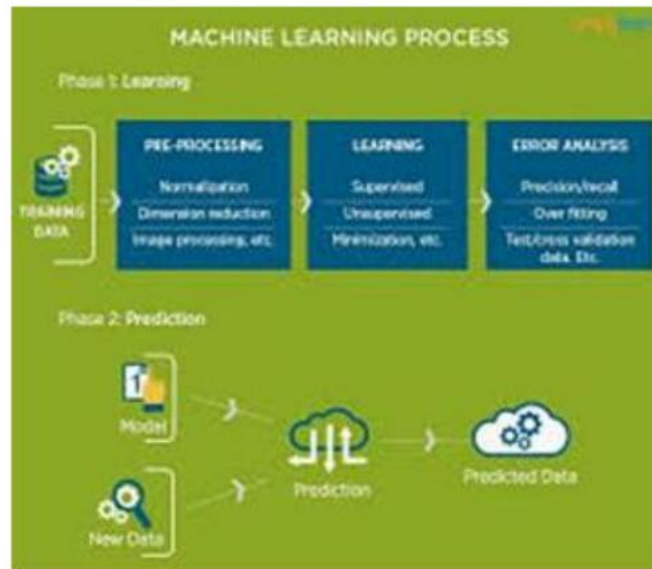


Figure 1.1.1: Process flow

1.2 USES OF MACHINE LEARNING:

Earlier in this article, we mentioned some applications of machine learning. To understand the concept of machine learning better, let's consider some more examples: web search results, real-time ads on web pages and mobile devices, email spam filtering, network intrusion detection, and pattern and image recognition. All these are by-products of applying machine learning to analyze huge volumes of data.

Traditionally, data analysis was always being characterized by trial and error, an approach that becomes impossible when data sets are large and heterogeneous. Machine learning comes as the solution to all this chaos by proposing clever alternatives to analyzing huge volumes of data.

By developing fast and efficient algorithms and data-driven models for real-time processing of data, machine learning can produce accurate results and analysis.

1.3 TYPES OF LEARNING ALGORITHMS:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

1.3.1 Supervised Learning :

When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of supervised learning.

Supervised machine learning algorithms uncover insights, patterns, and relationships from a labeled training data set – that is, a data set that already contains a known value for the target variable for each record. Because you provide the machine learning algorithm with the correct answers for a problem during training, it is able to “learn” how the rest of the features relate to the target, enabling you to uncover insights and make predictions about future outcomes based on historical data.

Examples of Supervised Machine Learning Techniques are Regression, in which the algorithm returns a numerical target for each example, such as how much revenue will be generated from a new marketing campaign.

Classification, in which the algorithm attempts to label each example by choosing between two or more different classes. Choosing between two classes is called binary classification, such as determining whether or not someone will default on a loan. Choosing between more than two classes is referred to as multiclass classification.

1.3.2 Unsupervised Learning:

When an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of uncorrelated

values. They are quite useful in providing humans with insights into the meaning of data and new useful inputs to supervised machine learning algorithms.

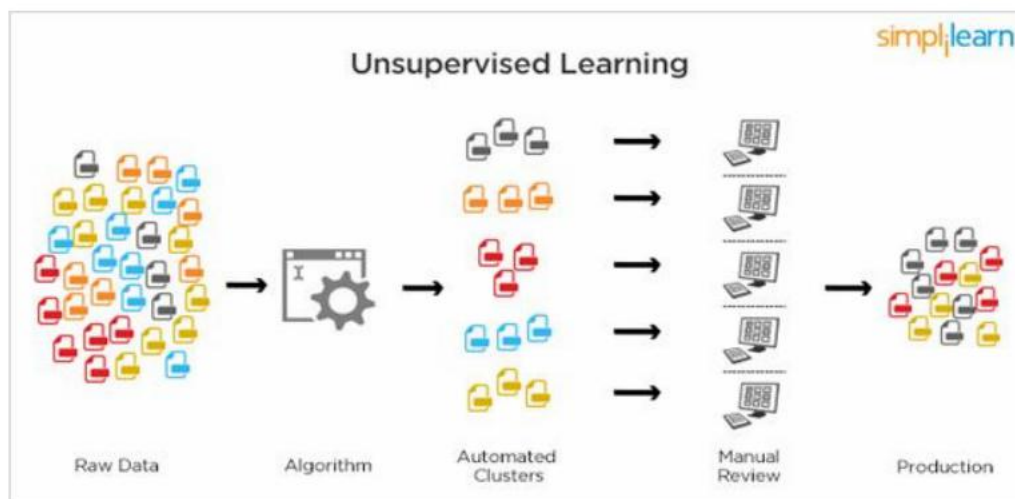


Figure 1.3.2.1: Unsupervised Learning

Popular techniques where unsupervised learning is used also include self-organizing maps, nearest neighbor mapping, singular value decomposition, and k-means clustering. Basically, online recommendations, identification of data outliers, and segment text topics are all examples of unsupervised learning.

1.3.3 Semi Supervised Learning:

As the name suggests, semi-supervised learning is a bit of both supervised and unsupervised learning and uses both labeled and unlabeled data for training. In a typical scenario, the algorithm would use a small amount of labeled data with a large amount of unlabeled data.

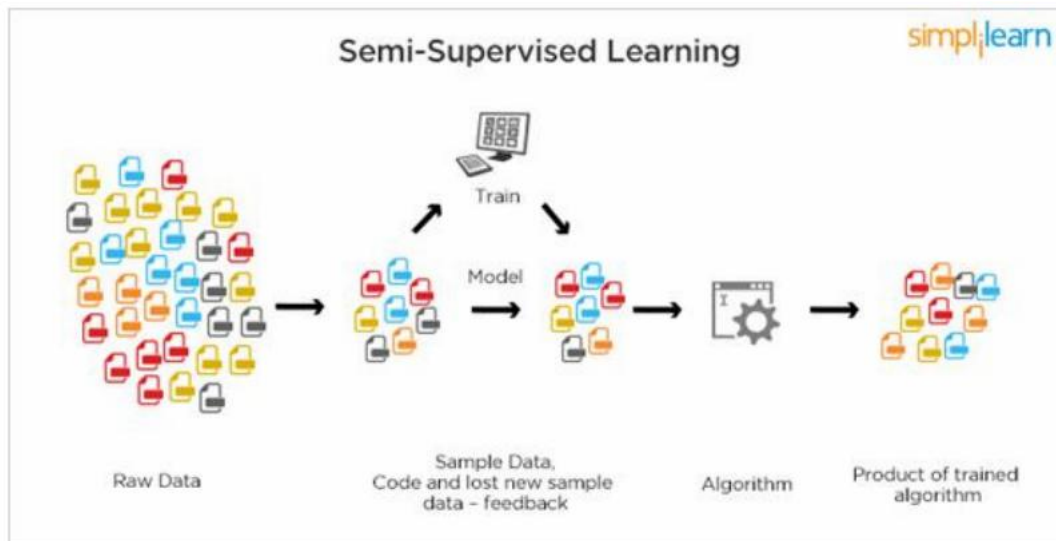


Figure 1.3.3.1 :Semi Supervised Learning

1.4 RELATION BETWEEN DATA MINING,MACHINE LEARNING AND DEEP LEARNING:

Machine learning and data mining use the same algorithms and techniques as data mining, except the kinds of predictions vary. While data mining discovers previously unknown patterns and knowledge, machine learning reproduces known patterns and knowledge—and further automatically applies that information to data, decision-making, and actions.

Deep learning, on the other hand, uses advanced computing power and special types of neural networks and applies them to large amounts of data to learn, understand, and identify complicated patterns. Automatic language translation and medical diagnoses are examples of deep learning.

CHAPTER 2

DEEP LEARNING

2.1 INFORMATION ABOUT DEEP LEARNING:

Deep learning is an artificial intelligence(AI)function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network.

2.1.1 IMPORTANCE OF MACHINE LEARNING:

The ability to process large numbers of features makes deep learning very powerful when dealing with unstructured data. However, deep learning algorithms can be overkill for less complex problems because they require access to a vast amount of data to be effective. For instance, Image Net, the common benchmark for training deep learning models for comprehensive image recognition, has access to over 14 million images.

If the data is too simple or incomplete, it is very easy for a deep learning model to become over fitted and fail to generalize well to new data. As a result, deep learning models are not as effective as other techniques (such as boosted decision trees or linear models) for most practical business problems such as understanding customer churn, detecting fraudulent transactions and other cases with smaller data sets and fewer features. In certain cases like multi classification, deep learning can work for smaller, structured data sets.

2.2 USES OF DEEP LEARNING:

Deep learning applications are used in industries from automated driving to medical devices. Automated Driving: Automotive researchers are using deep learning to automatically detect objects such as stop signs and traffic lights. In addition, deep learning is used to detect pedestrians, which helps decrease accidents.

2.3 TYPES OF LEARNING ALGORITHMS:

2.3.1 Convolutional Neural Network:

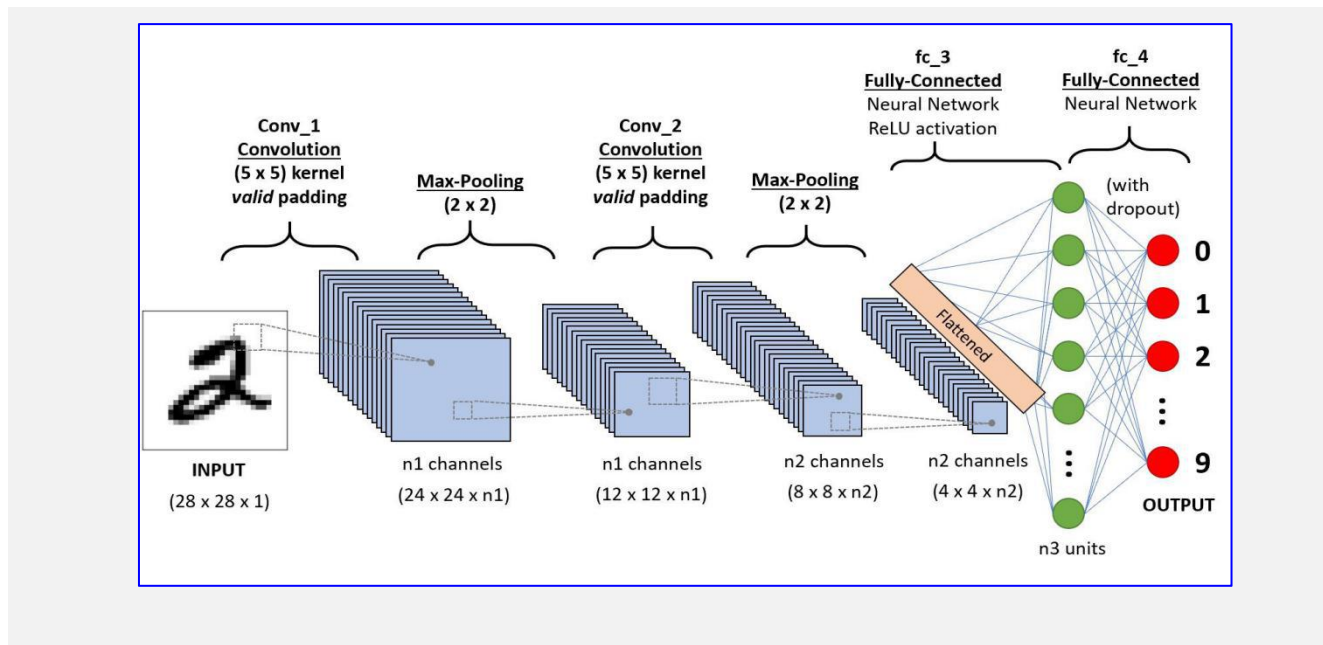


Figure 2.3.1.1: A CNN sequence to classify handwritten digits

Convolutional Neural Network (CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a CNN is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, CNN have the ability to learn these filters/characteristics.

The architecture of a CNN is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

2.3.2 Recurrent Neural Network(RNN):

Recurrent Neural Network(RNN) are a type of Neural Network where the output from previous step are fed as input to the current step. In traditional neural networks, all the inputs and outputs are independent of each other, but in cases like when it is required to predict the next word of a sentence, the previous words are required and hence there is a need to remember the previous words. Thus RNN came into existence, which solved this issue with the help of a Hidden Layer. The main and most important feature of RNN is Hidden state, which remembers some information about a sequence.

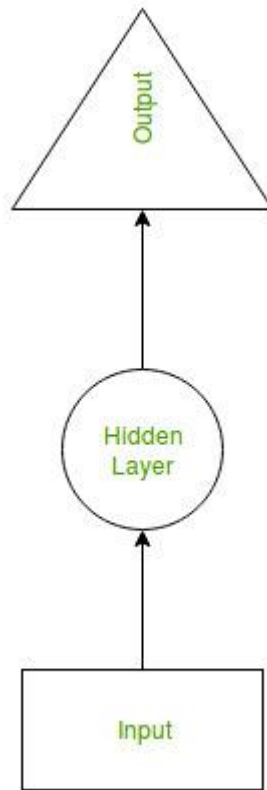


Figure 2.3.2.1: RNN

RNN have a “memory” which remembers all information about what has been calculated. It uses the same parameters for each input as it performs the same task on all the inputs or hidden layers to produce the output. This reduces the complexity of parameters, unlike other neural networks.

2.3.3 Long short-term memory (LSTM):

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture[1] used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition,[2] speech recognition[3][4] and anomaly detection in network traffic or IDS(intrusion detection systems).

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.

LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. LSTMs were developed to deal with the vanishing gradient problem that can be encountered when training traditional RNNs. Relative insensitivity to gap length is an advantage of LSTM over RNN, hidden Markov models and other sequence learning methods in numerous applications.

CHAPTER 3

PYTHON

Basic programming language used for machine learning is : PYTHON

3.1 INTRODUCTION TO PYTHON:

- Python is a high-level, interpreted, interactive and object-oriented scripting language.
- Python is a general purpose programming language that is often applied in scripting roles.
- Python is Interpreted: Python is processed at run time by the interpreter. You do not need to compile your program before executing it. This is like PERL and PHP.
- Python is Interactive: You can sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented: Python supports the Object-Oriented style or technique of programming that encapsulates code within objects.

3.2 HISTORY OF PYTHON:

- Python was developed by GUIDO VAN ROSSUM in early 1990's
- Its latest version is 3.7 , it is generally called as python3

3.3 FEATURES OF PYTHON:

- Easy-to-learn: Python has few keywords, simple structure, and a clearly defined syntax, This allows the student to pick up the language quickly.
- Easy-to-read: Python code is more clearly defined and visible to the eyes.
- Easy-to-maintain: Python's source code is fairly easy-to-maintaining.
- A broad standard library: Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- Portable: Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- Extendable: You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- Databases: Python provides interfaces to all major commercial databases.
- GUI Programming: Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

3.4 HOW TO SETUP PYTHON:

- Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.
- The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python.

3.4.1 Installation(using python IDLE):

- Installing python is generally easy, and nowadays many Linux and Mac OS distributions include a recent python.
- Download python from www.python.org
- When the download is completed, double click the file and follow the instructions to install it.
- When python is installed, a program called IDLE is also installed along with it. It provides a graphical user interface to work with python.



Figure 3.4.1.1:Python download

3.4.2 Installation(using Anaconda):

- Python programs are also executed using Anaconda.
- Anaconda is a free open source distribution of python for large scale data processing, predictive analytics and scientific computing.
- Conda is a package manager that quickly installs and manages packages.

- In WINDOWS:
- In windows
- Step 1: Open Anaconda.com/downloads in a web browser.
- Step 2: Download python 3.4 version for (32-bits graphic installer/64 -bit graphic installer)
- Step 3: select installation type(all users)
- Step 4: Select path(i.e. add anaconda to path & register anaconda as default python 3.4) next click install and next click finish
- Step 5: Open Jupyter notebook (it opens in default browser)

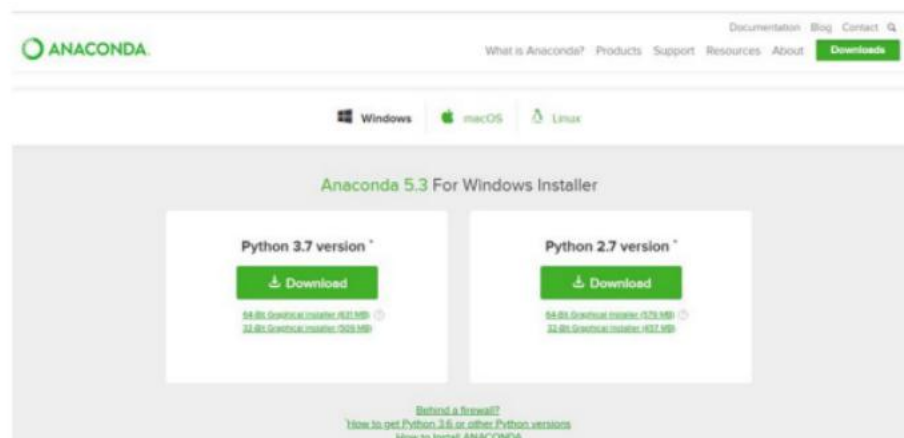


Figure 3.4.2.1: Anaconda download



Figure 3.4.2.2: Jupyter notebook

3.5 PYTHON VARIABLE TYPES:

- Variables are nothing but reserved memory locations to store values. This means that when you create a variable you reserve some space in memory.
- Variables are nothing but reserved memory locations to store values.
- Based on the data type of a variable, the interpreter allocates memory and decides what can be stored in the reserved memory.
- Python variables do not need explicit declaration to reserve memory space. The declaration happens automatically when you assign a value to a variable.
- Python has various standard data types that are used to define the operations possible on them and the storage method for each of them.
- Python has five standard data types –
 - o Numbers
 - o Strings
 - o Lists
 - o Tuples
 - o Dictionary

3.5.1 Python Numbers:

- Number data types store numeric values. Number objects are created when you assign a value to them.
- Python supports four different numerical types – int (signed integers) long (long integers, they can also be represented in octal and hexadecimal) float (floating point real values) complex (complex numbers).

3.5.2 Python Strings:

- Strings in Python are identified as a contiguous set of characters represented in the quotation marks.
- Python allows for either pairs of single or double quotes.
- Subsets of strings can be taken using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the string and working their way from -1 at the end.
- The plus (+) sign is the string concatenation operator and the asterisk (*) is the repetition operator.

3.5.3 Python Lists:

- Lists are the most versatile of Python's compound data types.
- A list contains items separated by commas and enclosed within square brackets([]).
- To some extent, lists are similar to arrays in C. One difference between them is that all the items belonging to a list can be of different data types.
- The values stored in a list can be accessed using the slice operator ([] and [:]) with indexes starting at 0 in the beginning of the list and working their way to end -1.
- The plus (+) sign is the list concatenation operator, and the asterisk (*) is the repetition operator.

3.5.4 Python Tuples:

- A tuple is another sequence data type that is similar to the list.
- A tuple consists of a number of values separated by commas. Unlike lists, however, tuples are enclosed within parentheses.
- The main differences between lists and tuples are: Lists are enclosed in brackets ([]) and their elements and size can be changed, while tuples are enclosed in parentheses (()) and cannot be updated.
- Tuples can be thought of as read-only lists.
- For example – Tuples are fixed size in nature whereas lists are dynamic. In other words, a tuple is immutable whereas a list is mutable. You can't add elements to a tuple. Tuples have no append or extend method. You can't remove elements from a tuple. Tuples have no remove or pop method.

3.5.5 Python Dictionary:

- Python's dictionaries are kind of hash table type. They work like associative arrays or hashes found in Perl and consist of key-value pairs. A dictionary key can be almost any Python type, but are usually numbers or strings. Values, on the other hand, can be any arbitrary Python object.
- Dictionaries are enclosed by curly braces ({ }) and values can be assigned and accessed using square braces ([]).
- You can use numbers to "index" into a list, meaning you can use numbers to find out what's in lists. You should know this about lists by now, but make sure you understand that you can only use numbers to get items out of a list.

- What a dict does is let you use anything, not just numbers. Yes, a dict associates one thing to another, no matter what it is.

3.6 PYTHON FUNCTION:

3.6.1 Defining a Function:

You can define functions to provide the required functionality. Here are simple rules to define a function in Python. Function blocks begin with the keyword `def` followed by the function name and parentheses (i.e.()).

Any input parameters or arguments should be placed within these parentheses. You can also define parameters inside these parentheses. The code block within every function starts with a colon (:) and is indented.

The statement `return [expression]` exits a function, optionally passing back an expression to the caller. A return statement with no arguments is the same as `return None`.

3.6.2 Calling a Function:

Defining a function only gives it a name, specifies the parameters that are to be included in the function and structures the blocks of code. Once the basic structure of a function is finalized, you can execute it by calling it from another function or directly from the Python prompt.

3.7 PYTHON USING OOPS CONCEPTS:

3.7.1 Class:

- Class:

A user-defined prototype for an object that defines a set of attributes that characterize any object of the class. The attributes are data members (class variables and instance variables) and methods, accessed via dot notation.

- Class variable:

A variable that is shared by all instances of a class. Class variables are defined within a class but outside any of the class's methods. Class variables are not used as frequently as instance variables are.

- Data member:

A class variable or instance variable that holds data associated with a class and its objects.

- Instance variable:

A variable that is defined inside a method and belongs only to the current instance of a class.

- Defining a Class:

We define a class in a very similar way how we define a function. Just like a function ,we use parentheses and a colon after the class name(i.e. (:)) when we define a class. Similarly, the body of our class is indented like a functions body is.

```
def my_function():  
    # the details of the  
    # function go here
```

```
class MyClass():  
    # the details of the  
    # class go here
```

Figure 3.7.1.1:Defining a class

3.7.2 `__init__` method in Class:

- The init method — also called a constructor — is a special method that runs when an instance is created so we can perform any tasks to set up the instance.
- The init method has a special name that starts and ends with two underscores: `__init__()`.

CHAPTER 4

CASE STUDY

4.1 PROBLEM STATEMENT:

The Main Aim of this project is to predict the stock prices of a particular company using the stock market data set. In this project, I am going to use LSTM networks to predict stock prices. It's important to note that there are always other factors that affect the price of stocks, such as the political atmosphere and the market. However, we won't focus on those factors for this project. This type of problem comes under time series prediction.

4.2 DATA SET:

The given data set consists of the following parameters:

Date: The date on which we are predicting the stock

Open: The value at which the stock is opened.

High: Maximum value of the stock

Low: Minimum value of the stock

Close: The value at which the stock is closed

Volume: The volume of stocks bought

Name: The name of the company

4.3 OBJECTIVE OF THE CASE STUDY:

In Stock Market Prediction, the aim is to predict the future value of the financial stocks of a company. The recent trend in stock market prediction technologies is the use of machine learning which makes predictions based on the values of current stock market indices by training on their previous values. Machine learning itself employs different models to make prediction easier and authentic.

CHAPTER 5

DATA PREPROCESSING

5.1 READING THE DATASET:

We can read the data set from the database or

we can read the data from client.

5.2 IMPORTING THE LIBRARIES:

We have to import the libraries as per the requirement of the algorithm.

```
1 import numpy as np
2 import pandas as pd
3 import seaborn as sns
4 import matplotlib.pyplot as plt
5 %matplotlib inline
6 import keras
```

Figure 5.2.1: Importing the required libraries

5.3 VERSIONS OF PACKAGES:

```
1 print(pd.__version__)  
2 print(np.__version__)  
3 print(sns.__version__)  
4 print(keras.__version__)
```

```
1.0.5  
1.18.5  
0.10.1  
2.3.1
```

Figure 5.3.1: versions of packages

5.4 IMPORTING THE DATA-SET:

Pandas in python provide an interesting method `read_csv()`. The `read_csv` function reads the entire data set from a comma separated values file and we can assign it to a Data Frame to which all the operations can be performed. It helps us to access each and every row as well as columns and each and every value can be access using the data frame. Any missing value or NaN value have to be cleaned.

```
1 df = pd.read_csv('My Drive/summer internship/all_stocks_5yr.csv')  
2 df.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	15.07	15.12	14.63	14.75	8407500	AAL
1	2013-02-11	14.89	15.01	14.26	14.46	8882000	AAL
2	2013-02-12	14.45	14.51	14.10	14.27	8126000	AAL
3	2013-02-13	14.30	14.94	14.25	14.66	10259500	AAL
4	2013-02-14	14.94	14.96	13.16	13.99	31879900	AAL

Figure 5.4.1 : Reading the data set

5.5 HANDLING MISSING VALUES:

Data can have missing values for a number of reasons such as observations that were not recorded and data corruption.

Handling missing data is important as many machine learning algorithms do not support data with missing values.

```
1 data.isnull().sum()
```

```
date      0  
open      0  
dtype: int64
```

Figure 5.5.1: checking for missing values

We don't have any missing values in the data set.

CHAPTER 6

FEATURE SELECTION

6.1 SELECTING A PARTICULAR COMPANY:

```
1 data =df[df.Name=='AAP']  
2 data.head()
```

	date	open	high	low	close	volume	Name
2518	2013-02-08	78.34	79.72	78.0100	78.90	1298137	AAP
2519	2013-02-11	78.65	78.91	77.2300	78.39	758016	AAP
2520	2013-02-12	78.39	78.63	77.5132	78.60	876859	AAP
2521	2013-02-13	78.90	79.13	77.8500	78.97	1038574	AAP
2522	2013-02-14	78.66	79.72	78.5850	78.84	1005376	AAP

Figure 6.1.1: Selecting a particular company

HERE I HAVE SELECTED A COMPANY NAMED AAP

6.1.1 ABOUT COMPANY:

Advance Auto Parts, Inc. (Advance) is an American automotive aftermarket parts provider. Headquartered in Raleigh, North Carolina, it serves both professional installer and do-it-yourself (DIY) customers. As of July 13, 2019, Advance operated 4,912 stores and 150 Worldpac branches in the United States and Canada. The Company also serves 1,250 independently owned Carquest branded stores across these locations in addition to Mexico, the Bahamas, Turks and Caicos and British Virgin Islands. The company's stores and branches offer a broad selection of brand name, original equipment manufacturer (OEM) and private label automotive replacement parts, accessories, batteries and maintenance items for domestic and imported cars, vans, sport utility vehicles and light and heavy duty trucks.

6.1.2 CHECK RAW DATA:

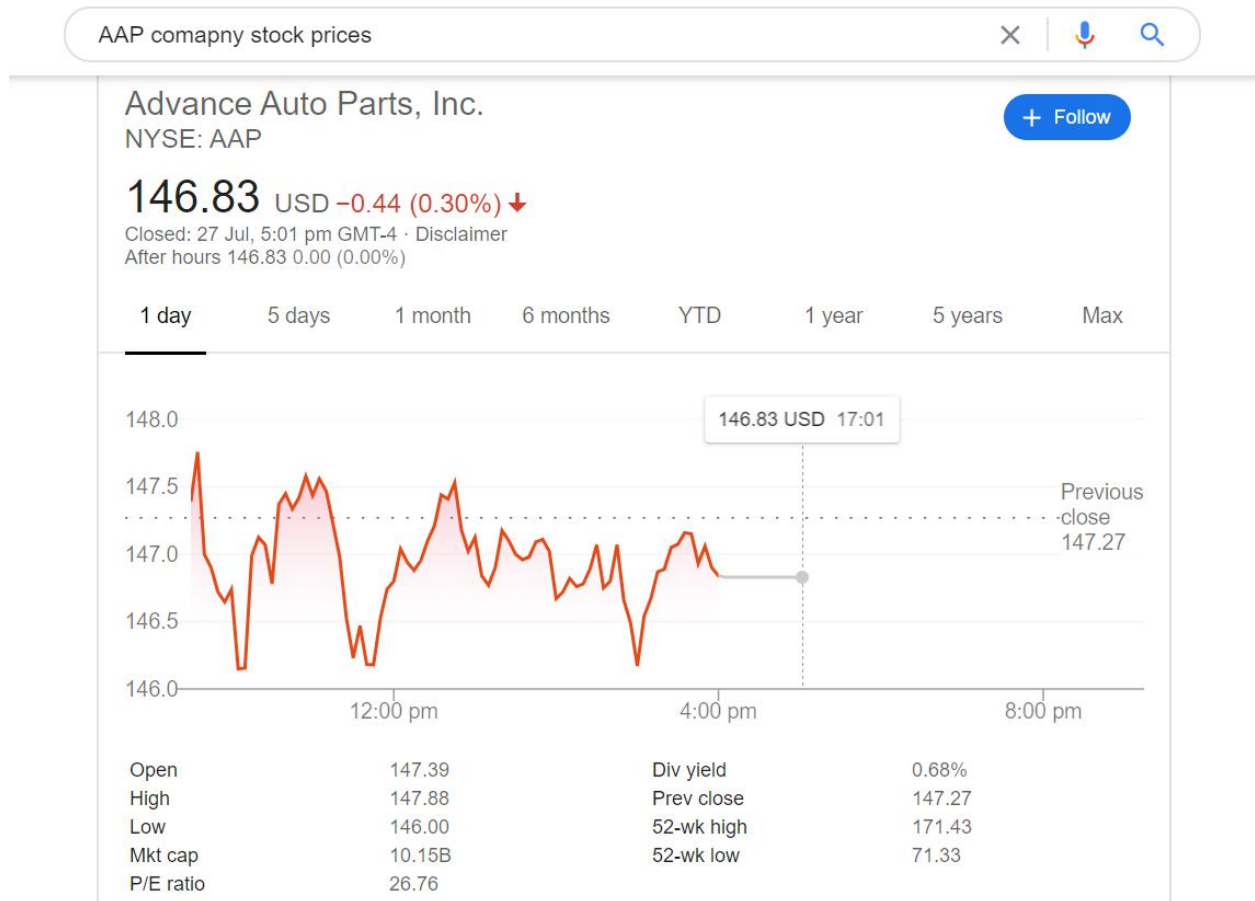


Figure 6.1.2.1: checking the stock price of a company for unknown data

6.2 SELECTING RELEVANT FEATURE FOR ANALYSIS:

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in.

```
1 data = data[['date', 'open']]
2 data
```

	date	open
2518	2013-02-08	78.34
2519	2013-02-11	78.65
2520	2013-02-12	78.39
2521	2013-02-13	78.90
2522	2013-02-14	78.66
...
3772	2018-02-01	116.24
3773	2018-02-02	115.92
3774	2018-02-05	113.05
3775	2018-02-06	108.28
3776	2018-02-07	112.09

1259 rows × 2 columns

Figure 6.2.1: Selecting relevant features

6.3 SPLITTING THE DATA INTO TRAIN AND TEST:

Splitting the data : after the preprocessing is done then the data is split into train and test sets.

- In Machine Learning in order to access the performance of the classifier. You train the classifier using 'training set' and then test the performance of your classifier on unseen 'test set'. An important point to note is that during training the classifier only uses the training set . The test set must not be used during training the classifier. The test set will only be available during testing the classifier.
- training set - a subset to train a model.(Model learns patterns between Input and Output)
- test set - a subset to test the trained model.(To test whether the model has correctly learnt)

1.

- The amount or percentage of Splitting can be taken as specified (i.e. train data = 75% , test data =25% or train data = 80% , test data= 20%) .
- First we need to identify the input and output variables and we need to separate the input set and output set.
- In scikit learn library we have a package called `model_selection` in which `train_test_split` method is available .we need to import this method.
- This method splits the input and output data to train and test based on the percentage specified by the user and assigns them to four different variables(we need to mention the variables) .

As we work with data sets, a machine learning algorithm works in two stages. We usually split the data around 20%-80% between testing and training stages. Under supervised learning, we split a data set into a training data and test data in Python ML.

```
1 ## train_test_split
2 X_train=df1.iloc[:1002,:5]
3 X_test=df1.iloc[1002:,:5]
4 y_train=df1['d6'][:1002]
5 y_test=df1['d6'][1002:]
6 print(X_train.shape)
7 print(X_test.shape)
8 print(y_train.shape)
9 print(y_test.shape)
```

```
(1002, 5)
(251, 5)
(1002,)
(251,)
```

Figure 6.3.1: Train Test Split

6.4 SCALING THE DATA:

It is a step of Data preprocessing which is applied to independent variables or features of data. It basically helps to normalize the data within a particular range. Sometimes, it also helps in speeding up the calculations in an algorithm.

Formula used in Back end Standardization replaces the values by their Z scores.

$$z = \frac{x - \mu}{\sigma}$$

```
1 #scaling the data
2 from sklearn.preprocessing import StandardScaler
3 sc=StandardScaler()
4 sc.fit(X_train)
5 X_train_sc=pd.DataFrame(sc.transform(X_train),columns=X_train.columns)
6 X_test_sc=pd.DataFrame(sc.transform(X_test),columns=X_train.columns)
7 X_train_sc.describe()
```

	d1	d2	d3	d4	d5
count	1.002000e+03	1.002000e+03	1.002000e+03	1.002000e+03	1.002000e+03
mean	-6.331152e-16	7.659652e-16	5.285193e-17	-2.047597e-15	-2.261442e-16
std	1.000499e+00	1.000499e+00	1.000499e+00	1.000499e+00	1.000499e+00
min	-1.896850e+00	-1.901980e+00	-1.907164e+00	-1.912368e+00	-1.917553e+00
25%	-5.752281e-01	-5.699688e-01	-5.652483e-01	-5.590802e-01	-5.359346e-01
50%	3.000091e-01	3.002195e-01	3.024915e-01	3.024348e-01	3.024373e-01
75%	7.239645e-01	7.250333e-01	7.303431e-01	7.334698e-01	7.344461e-01
max	2.058381e+00	2.058310e+00	2.058560e+00	2.058609e+00	2.058739e+00

Figure 6.4.1: scaling the data

CHAPTER 7

MODEL BUILDING

7.1 Introduction to LSTM

Long Short Term Memory is a kind of recurrent neural network. In RNN output from the last step is fed as input in the current step. LSTM was designed by Hochreiter & Schmidhuber. It tackled the problem of long-term dependencies of RNN in which the RNN cannot predict the word stored in the long term memory but can give more accurate predictions from the recent information. As the gap length increases RNN does not give efficient performance. LSTM can by default retain the information for long period of time. It is used for processing, predicting and classifying on the basis of time series data.

7.1.1 Structure Of LSTM:

LSTM has a chain structure that contains four neural networks and different memory blocks called cells.

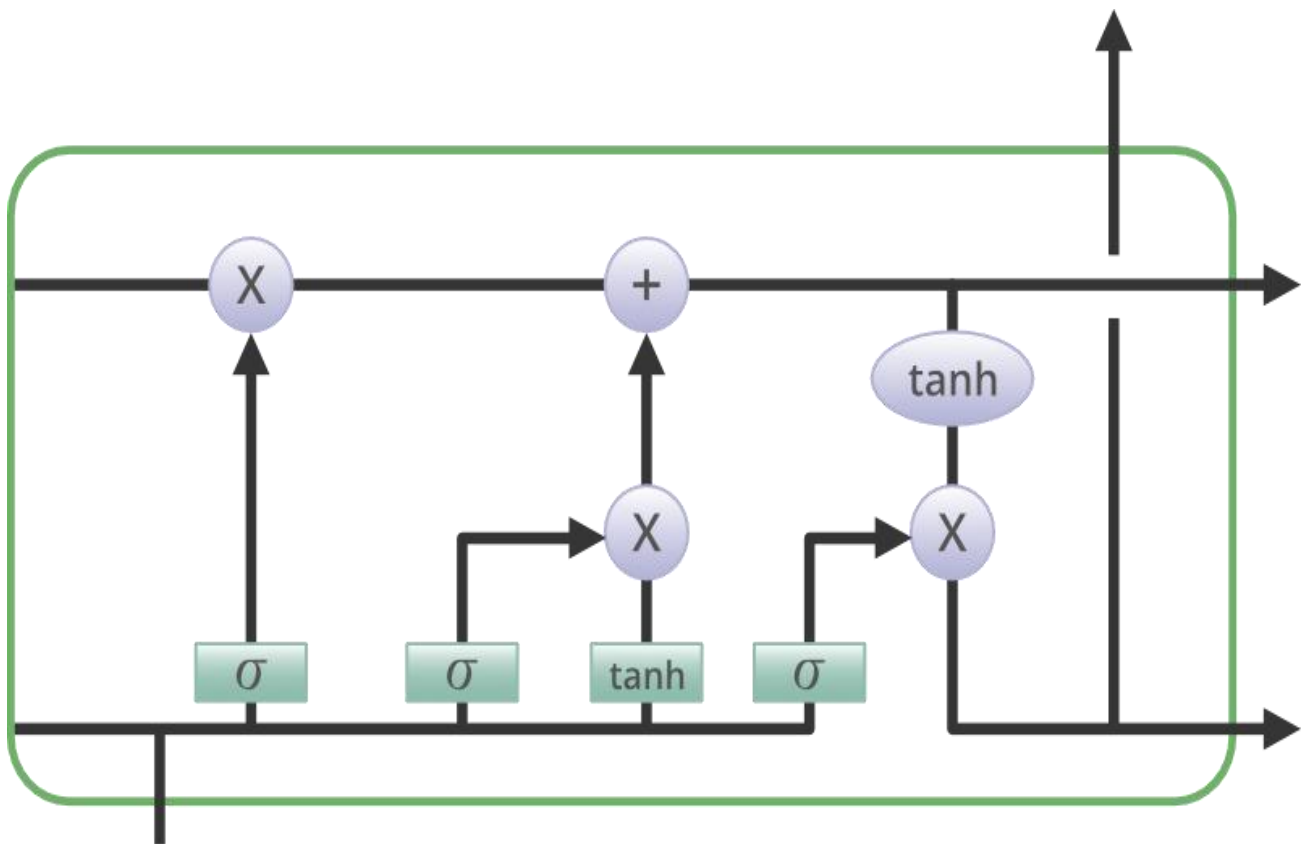


Figure 7.1.1.1: Structure of LSTM

Information is retained by the cells and the memory manipulations are done by the gates. There are three gates –

1. Forget Gate:

The information that no longer useful in the cell state is removed with the forget gate. Two inputs x_t (input at the particular time) and h_{t-1} (previous cell output) are fed to the gate and multiplied with weight matrices followed by the addition of bias. The resultant is passed through an activation function which gives a binary output. If for a particular cell state the output is 0, the piece of information is forgotten and for the output 1, the information is retained for the future use.

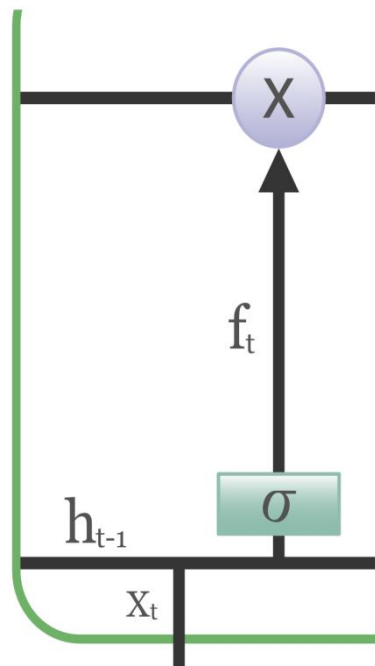


Figure 7.1.1.2:Forget gate

2. Input gate:

Addition of useful information to the cell state is done by input gate. First, the information is regulated using the sigmoid function and filter the values to be remembered similar to the forget gate using inputs h_{t-1} and x_t . Then, a vector is created using tanh function that gives output from -1 to +1, which contains all the possible values from h_{t-1} and x_t . At last, the values of the vector and the regulated values are multiplied to obtain the useful information

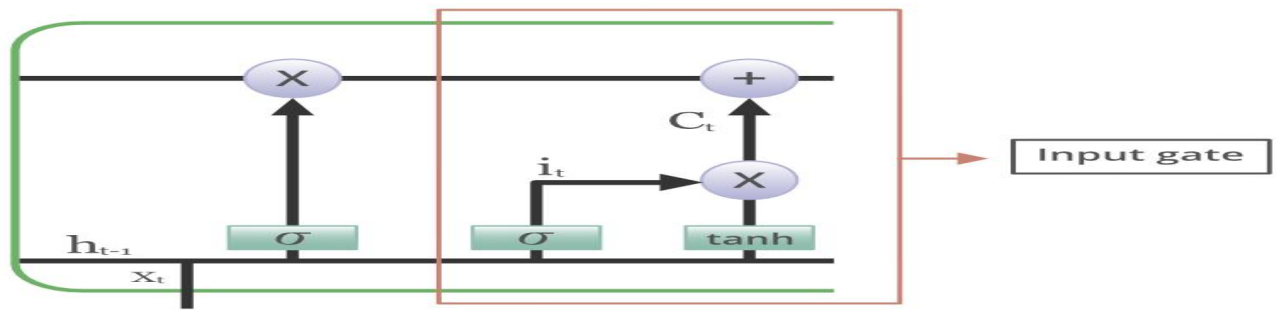


Figure 7.1.1.3: Input gate

3. Output gate:

The task of extracting useful information from the current cell state to be presented as an output is done by output gate. First, a vector is generated by applying tanh function on the cell. Then, the information is regulated using the sigmoid function and filter the values to be remembered using inputs h_{t-1} and x_t . At last, the values of the vector and the regulated values are multiplied to be sent as an output and input to the next cell.

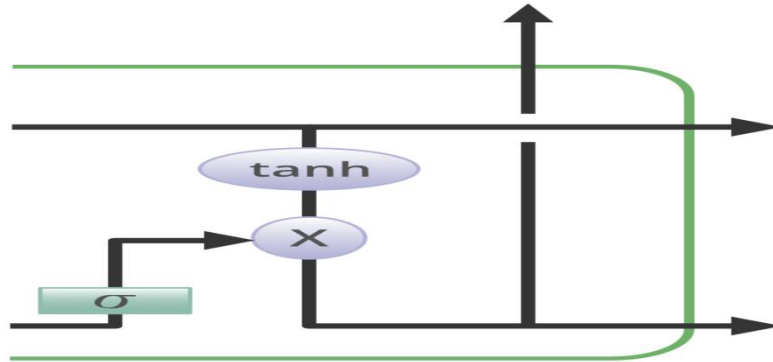


Figure 7.1.1.4: Output gate

7.2 BUILDING LSTM MODEL:

In order to build the LSTM, we need to import a couple of modules from [Keras](#):

- 1.Sequential for initializing the neural network
- 2.Dense for adding a densely connected neural network layer
- 3.LSTM for adding the Long Short-Term Memory layer
- 4.Dropout for adding dropout layers that prevent over fitting

We add the LSTM layer and later add a few Dropout layers to prevent over fitting.

```
1 from keras.models import Sequential
2 from keras.layers import Dense,Dropout,LSTM
3 model=Sequential()
4 model.add(LSTM(256,input_shape=(5,1)))
5 model.add(Dense(1))
6 model.summary()
```

Using TensorFlow backend.
Model: "sequential_1"

Layer (type)	Output Shape	Param #
=====	=====	=====
lstm_1 (LSTM)	(None, 256)	264192
dense_1 (Dense)	(None, 1)	257
=====	=====	=====
Total params: 264,449		
Trainable params: 264,449		
Non-trainable params: 0		
=====		

Figure 7.2.1: building the LSTM model

7.3 COMPILE THE MODEL:

You can either instantiate an optimizer before passing it to `model.compile()` or you can pass it by its string identifier. In the latter case, the default parameters for the optimizer will be used.

Metrics

A metric is a function that is used to judge the performance of your model.

Metric functions are similar to loss functions, except that the results from evaluating a metric are not used when training the model. Note that you may use any loss function as a metric.

Losses

The purpose of loss functions is to compute the quantity that a model should seek to minimize during training.

```
1 model.compile(optimizer='adam',loss='mse')
```

Figure 7.3.1: compiling the model

7.3.1 MODEL FITTING:

Model fitting is a measure of how well a machine learning model generalizes to similar data to that on which it was trained. A model that is well-fitted produces more accurate outcomes. A model that is overfitted matches the data too closely.

Model fitting is the essence of machine learning. If your model doesn't fit your data correctly, the outcomes it produces will not be accurate enough to be useful for practical decision-making. A properly fitted model has hyper parameters that capture the complex relationships between known variables and the target variable, allowing it to find relevant insights or make accurate predictions.

```
1 history = model.fit(X_train_sc,y_train,epochs=150,validation_data=(X_test_sc,y_test))
```

Figure 7.3.1.1: fitting the model

```

1 tr_loss = history.history['loss']
2 val_loss = history.history['val_loss']
3 ep = list(range(1,151))
4 plt.plot(ep,tr_loss,color='r')
5 plt.plot(ep,val_loss,color='b')
6 plt.title('visualization')
7 plt.xlabel('ep')
8 plt.ylabel('tr_loss and val_loss')
9 plt.show()

```

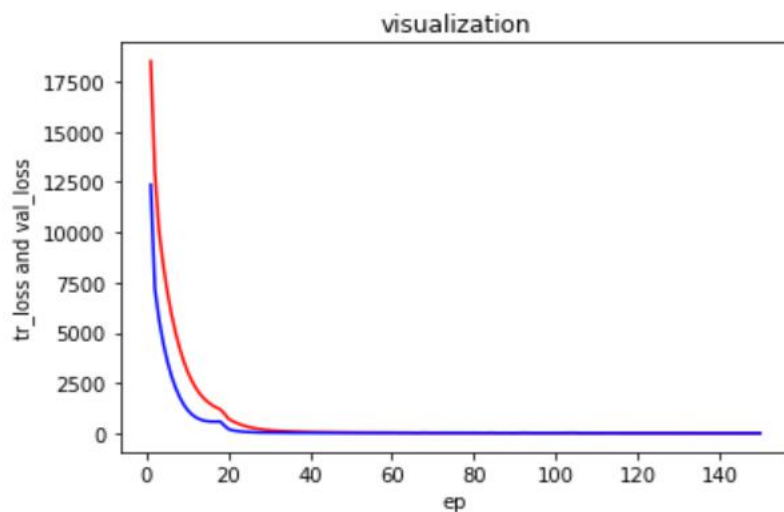


Figure 7.3.1.2: visualizing the loss values

7.4 EVALUATE THE MODEL:

I have plotted predicted values and actual values on y-axis and range on x-axis.

```
1 plt.plot(range(len(X_test_sc)),model.predict(X_test_sc).flat,color='r')#predicted values
2 plt.plot(range(len(X_test_sc)),y_test,color='b')#actual values
3 plt.title('predicting the open value')
4 plt.xlabel('range')
5 plt.ylabel('predicted value and true value')
6 plt.show()
```

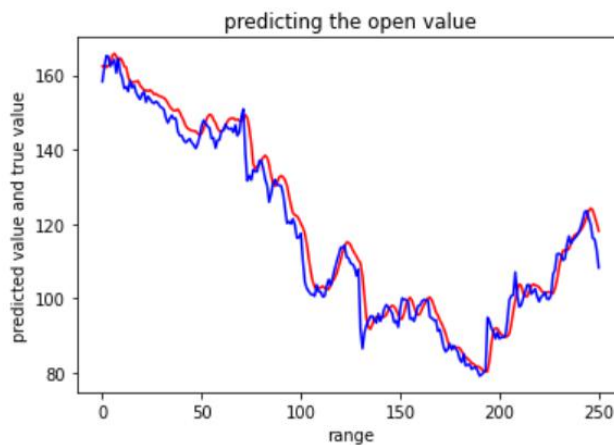
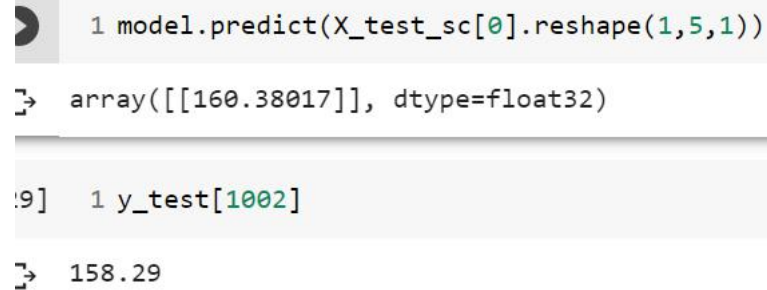


Figure 7.4.1: Evaluating the model with test values.

For every range, both the actual values and predicted are same.

7.5 COMPARE THE RESULT:

Comparing the model predicted values and the actual values.



The image shows a Jupyter Notebook interface with two code cells. The first cell contains the code `1 model.predict(X_test_sc[0].reshape(1,5,1))` and its output is `array([[160.38017]], dtype=float32)`. The second cell contains the code `:9] 1 y_test[1002]` and its output is `158.29`. The predicted value is approximately 160.38 and the actual value is 158.29.

```
1 model.predict(X_test_sc[0].reshape(1,5,1))  
array([[160.38017]], dtype=float32)  
:9] 1 y_test[1002]  
158.29
```

Figure 7.5.1:predicted and actual values

From the above values,both the values are almost equal.

CONCLUSION

In this project, I used Deep Learning to predict the stock prices. After importing the data, I have handled the missing values. I have selected particular company name called “AAP” to predict the future values. I have Selected relevant features for the analysis. I applied training and testing on the data, then I applied scaling for the data. I have used LSTM for model building. I used adam and mse for compiling the model. I got train loss as 12.4 and validation loss as 15.8. I have evaluated the model, and compared the result with the actual value. model predicted value is 160.38 and actual value is 158.29. Both the values are similar.

REFERENCES

1. https://en.wikipedia.org/wiki/Machine_learning
2. <https://github.com/surajr/Stock-Predictor-using-LSTM/blob/master/Stock-Predictor-using-LSTM.ipynb>
3. Github Project Repository Link: <https://github.com/Mallika39/stock-price-prediction>

