

PROFESSIONAL TRAINING REPORT
at
Sathyabama Institute of Science and Technology
(Deemed to be University)

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in
Information Technology

By

VAISHNAVI D

(Reg. No. 37120079)



DEPARTMENT OF INFORMATION
TECHNOLOGY

SCHOOL OF COMPUTING

SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGYUNIVERSITY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMILNADU.



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai –
600 119

www.sathyabama.ac.in



SCHOOL OF COMPUTING

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report is the bonafide work of (Reg. No. 37120079) who underwent the professional training in "**STOCK MARKET PREDICTION**" under our supervision from June 2020 to August 2020.

Internal Guide

Ms.K. INDIRA M.Tech.,

Head of the Department

Dr. R.SUBHASHINI M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I, VAISHNAVI.D(Reg. No. 37120079) hereby declare that the Professional Training Report on “**STOCK MARKET PREDICTION**” done by me under the guidance of Ms. K. INDIRA M.Tech., at Sathyabama institute is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph.D, Dean**, School of Computing , **Dr.SUBHASHINI M.E., Ph.D.** , Head of the Department of Information and Technology for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.,K.INDIRA M.Tech.**,for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Information Technology** who were helpful in many ways for the completion of the project.

Certificate of Completion

This is to certify that **Vaishnavi D** successfully
completed 14.5 total hours of **Learn Ethical
Hacking From Scratch** online course on Aug. 15,
2020

Zaid Sabih *z Security*
Zaid Sabih, Instructor z Security, Instructor

&

 Udemy

Certificate no: UC-6e1dea02-696d-4ec9-a717-326c26423e09
Certificate url: udemy.com/certificate/UC-6e1dea02-696d-4ec9-a717-326c26423e09

#BeAble

ABSTRACT

There is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price. Stock Market Analysis of stocks using data mining will be useful for new investors to invest in stock market based on the various factors considered by the software. Stock market includes daily activities like Sensex calculation, exchange of shares. The exchange provides an efficient and transparent market for trading in equity, debt instruments and derivatives. Our aim is to create software that analyses previous stock data of certain companies, with help of certain parameters that affect stock value. We are going to implement these values in data mining algorithms and we will be able to decide which algorithm gives the best result. This will also help us to determine the values that particular stock will have in near future. We will determine the patterns in data with help of machine learning algorithms. Two versions of prediction system will be implemented; one using linear regression and other using Support Vector Machines. The experimental objective will be to compare the forecasting ability of machine learning algorithms. We will test and evaluate both the systems with same test data to find their prediction accuracy.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	Abstract	6
	List of figures	9
1	INTRODUCTION	
	1.1 Machine learning -overview	10
	1.2 What is Machine Learning	10
	1.3 Machine Learning Applications	11
	1.4 Machine Learning –Tasks	11
2	LITERATURE SURVEY	15
3	SYSTEM ANALYSIS	
	3.1 Existing System	17
	3.2 Proposed System	17
4	SYSTEM REQUIREMENT	
	4.1 Hardware Requirement	18
	4.2 Software Requirement	18
	4.3 Python Technology	18
	4.4 Python Platform	19
	4.5 Python Library	19

5	SYSTEM ARCHITECTURE	20
6	MODULE DESCRIPTION	
	6.1 List of modules	21
	6.2 Collection dataset	21
	6.3 Data-Cleaning	22
	6.4 Feature Engineering	22
	6.5 Score Model	23
	6.6 Evaluate Model	24
	6.7 Result	24
7	CONCLUSIONS	25
	REFERENCES	26
	APPENDIX	
	(A) Source code	27
	(B) Screenshots	31

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
1	System architecture	20
2	Collection of Dataset	21
3	Training Set	22
4	New Data	24
5	prediction of stock price on graph	24

CHAPTER 1

INTRODUCTION

1.1 MACHINE LEARNING – OVERVIEW

Machine learning is a very hot topic for many key reasons, and because it provides the ability to automatically obtain deep insights, recognize unknown patterns, and create high performing predictive models from data, all without requiring explicit programming instructions.

This high level understanding is critical if ever involved in a decision-making process surrounding the usage of machine learning, how it can help achieve business and project goals, which machine learning techniques to use, potential pitfalls, and how to interpret the results.

1.2 WHAT IS MACHINE LEARNING?

Machine learning is a subfield of computer science, but is often also referred to as predictive analytics, or predictive modeling. Its goal and usage is to build new and/or leverage existing algorithms to learn from data, in order to build generalizable models that give accurate predictions, or to find patterns, particularly with new and unseen similar data.

Imagine a dataset as a table, where the rows are each observation (aka measurement, data point, etc), and the columns for each observation represent the features of that observation and their values .At the outset of a machine learning project, a dataset is usually split into two or three subsets. The minimum subsets are the training and test datasets, and often an optional third validation dataset is created as well. Once these data subsets are created from the primary dataset, a predictive model or classifier is trained using the training data, and then the model's predictive accuracy is determined using the test data.

As mentioned, machine learning leverages algorithms to automatically model and find patterns in data, usually with the goal of predicting some target output or response. These algorithms are heavily based on statistics and mathematical optimization.

Optimization is the process of finding the smallest or largest value (minima or maxima) of a function, often referred to as a loss, or cost function in the minimization case. One of the most popular optimization algorithms used in machine learning is called gradient descent, and another is known as the the normal equation. In a nutshell, machine learning is all about automatically learning a highly accurate predictive or classifier model, or finding unknown patterns in data, by leveraging learning algorithms and optimization techniques.

1.3 MACHINE LEARNING APPLICATIONS

As we move forward into the digital age, One of the modern innovations we've seen is the creation of **Machine Learning**. This incredible form of artificial intelligence is already being used in various industries and professions. For Example, Image and Speech Recognition, Medical Diagnosis, Prediction, Classification, Learning Associations, Statistical Arbitrage, Extraction, Regression. Today we're looking at all these Machine Learning Applications in today's modern world.

1.4 MACHINE LEARNING – TASKS

The **most common machine learning tasks** that one may come across while trying to solve a machine learning problem. Under each task are also listed a set of **machine learning methods** that could be used to resolve these tasks. Following are the key machine learning tasks briefed later in this article:

- Feature selection
- Regression
- Classification
- Clustering

- Multivariate querying
- Density estimation
- Dimension reduction
- Testing and matching

Following are top 8 most common machine learning tasks while solving an advanced analytics problem:

1. Feature Selection:

Feature selection is one of the critical tasks which would be used when building machine learning models. Feature selection is important because selecting right features would not only help build models of higher accuracy but also help achieve objectives related to building simpler models, reduce overfitting etc. The following are some of the techniques which could be used for feature selection:

- Filter methods which helps in selecting features based on the outcomes of statistical tests.
 - Pearson's correlation
 - Linear discriminant analysis (LDA)
 - Analysis of Variance (ANOVA)
 - Chi-square tests
- Wrapper methods which helps in feature selection by using a subset of features and determining the model accuracy.
 - Forward selection
 - Backward elimination
 - Recursive feature elimination
- Regularization techniques which penalizes one or more features appropriately to come up with most important features.
 - LASSO (L1) regularization
 - Ridge (L2) regularization

2. Regression:

Regression tasks mainly deal with estimation of numerical values (**continuous variables**). Some of the examples include estimation of

housing price, product price, stock price etc. Some of the following ML methods could be used for solving regressions problems:

- Kernel regression (*Higher accuracy*)
- Gaussian process regression (*Higher accuracy*)
- Regression trees
- Linear regression
- Support vector regression
- LASSO

3. Classification:

Classification tasks is simply related with predicting a category of a data (**discrete variables**). Some of the common use cases could be found in the area of healthcare such as whether a person is suffering from a particular disease or not. It also has its application in financial use cases such as determining whether a transaction is fraud or not. Kernel discriminant analysis (*Higher accuracy*)

- K-Nearest Neighbors (*Higher accuracy*)
- Artificial neural networks (ANN) (*Higher accuracy*)
- Support vector machine (SVM) (*Higher accuracy*)
- Random forests (*Higher accuracy*)
- Decision trees
- Boosted trees
- Logistic regression
- naive Bayes
- Deep learning

4. Clustering:

Clustering tasks are all about finding natural groupings of data and a label associated with each of these groupings (clusters). Some includes customer segmentation, product features identification for product roadmap Mean-shift (*Higher accuracy*)

- Hierarchical clustering
- K-means
- Topic models

5. Multivariate querying:

Multivariate querying is about querying or finding similar objects. Some of the following ML methods could be used for such problems:

- Nearest neighbors
- Range search
- Farthest neighbors

6. Density estimation:

Density estimation problems are related with finding likelihood or frequency of objects. In probability and statistics, density estimation is the construction of an estimate, based on observed data, of an unobservable underlying probability density function.

- Kernel density estimation (*Higher accuracy*)
- Mixture of Gaussians
- Density estimation tree

7. Dimension reduction:

Dimension reduction is the process of reducing the number of random variables under consideration, and can be divided into feature selection and feature extraction. Manifold learning/KPCA (*Higher accuracy*)

- Principal component analysis
- Independent component analysis
- Gaussian graphical models
- Non-negative matrix factorization
- Compressed sensing

8. Testing and matching:

Testing and matching tasks relates to comparing data sets. Following are some of the methods that could be used for such kind of problems:

- Minimum spanning tree
- Bipartite cross-matching
- N-point correlation

CHAPTER 2

LITERATURE SURVEY

1. Efficacy of News Sentiment for Stock Market Prediction

Stock Market trend prediction will always remain a challenging task due to stochastic nature. The enormous amount of data generated by the news, blogs, reviews, financial reports and social media are considered a treasure of knowledge for researchers and investors.

2. Literature review on Artificial Neural Networks Techniques Application for Stock Market Prediction and as Decision Support Tools

Objectives: This literature review is aiming to explore the use Artificial Neural Network (ANN) techniques in the field of stock market prediction. Design: Content analysis research technique. Data sources: Information retrieved from ProQuest electronic databases. Review methods: Utilizing key terms and phrases associated with Artificial Neural Network Stock Market Prediction from 2013-2018. Out of the 129 scholarly journal reviewed, there are 4 stock market studies met the inclusion criteria.

3. Stock Market Prediction based on Social Sentiments using Machine Learning

Machine learning and artificial intelligence techniques are being used in conjunction with data mining to solve a plethora of real world problems. These techniques have proven to be highly effective, yielding maximum accuracy with minimal monetary investment and also saving huge amounts of time. To add to their annual income, nowadays, people have started looking at stock investments as a lucrative option. With expert guidance and intelligent planning, we can almost double our annual revenue through stock returns.

4. Predicting the Effects of News Sentiments on the Stock Market

Stock market forecasting is very important in the planning of business activities. Stock price prediction has attracted many researchers in multiple disciplines including computer science, statistics, economics, finance, and operations research. Recent studies have shown that the vast amount of online information in the public domain such as Wikipedia usage pattern, news stories from the mainstream media, and social media discussions can have an observable effect on investors' opinions towards financial markets.

5. Optimised Prediction Model for Stock Market Trend Analysis

The main objective of this work is to add to the academic understanding of stock market analysis using some well defined algorithms and machine learning techniques. Stock price forecasting is a popular and important topic in financial studies and at academic levels. Share Market is not a neat place for analyzing since there are no significant rules to estimate or predict the price of share in the share market. Many a method like technical analysis, fundamental analysis, time series analysis and statistical analysis, etc. have been used in an attempt to analyze the share trends in the market but none of these methods have so far proved to be a universal approach for acceptance as a prediction tool.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

investing money into unpredictable, unstable, and uncontrollable facets can be extremely risky. Like the lottery, the success of stock market trading is partly attributed to luck. Many people have lost vast amounts of money through poor investment decisions that they've made.

Recently, investors with shares in loan-giving companies and American car manufacturers, which were previously a fairly stable investment, have suffered severe losses due to the economic crisis. Investors must understand and accept this risk as an intrinsic part of investing.

There are, however, attractive benefits to successful financial investments. With intelligent decisions, investing can yield significant capital gains, stability, and security. By analyzing the trends of the stock market, the companies one is invested in, and by following an investment strategy, one can be successful in the stock market.

There has been much research into various ways of analyzing the stock market as a means of facilitating intelligent investment decisions. These "intelligent decisions" are paramount to the success of an investment, and will be examined in this experiment.

3.2 PROPOSED SYSTEM

Two versions of prediction system will be implemented; one using linear regression and other using Support Vector Machines.

The experimental objective will be to compare the forecasting ability of machine learning algorithms.

We will test and evaluate both the systems with same test data to find their prediction accuracy.

CHAPTER 4

SYSTEM REQUIREMENTS

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

4.1 HARDWARE REQUIREMENTS

System	: Pentium IV 2.4 GHz
Hard Disk	: 40 GB
Floppy Drive	: 1.44 Mb
Monitor	: 15 VGA Colour
Mouse	: Logitech
Ram	: 512 Mb

4.2 SOFTWARE REQUIREMENTS

Operating system	: Windows 10
IDE	: anaconda navigator
Coding Language	: python

4.3 PYTHON TECHNOLOGY

Python is an interpreted , object-oriented programming language similar to PERL, that has gained popularity because of its clear syntax and readability. Python is said to be relatively easy to learn and portable, meaning its statements can be interpreted in a number of operating systems, including UNIX-based systems, Mac OS, MS-DOS, OS/2, and various versions of Microsoft Windows 98. Python was created by Guido van Rossum, a former resident of the Netherlands,. The source

code is freely available and open for modification and reuse. Python has a significant number of users. A notable feature of Python is its indenting of source statements to make the code easier to read. Python offers dynamic data type, ready-made class, and interfaces to many system calls and libraries. It can be extended, using the C or C++ language. Python can be used as the script in Microsoft's Active Server Page (ASP) technology. The scoreboard system for the Melbourne (Australia) Cricket Ground is written in Python. Z Object Publishing Environment, a popular Web application server, is also written in the Python language

4.4 PYTHON PLATFORM

Apart from Windows, Linux and MacOS, CPython implementation runs on 21 different **platforms**. Ironpython is a framework based **Python** implementation and it is capable of running in both Windows, Linux and in other environments where .NET framework is available.

4.5 PYTHON LIBRARY

Machine Learning, is the science of programming a computer by which they are able to learn from different kinds of data. They are typically used to solve various problems. People used to perform Machine Learning tasks by manually coding all the algorithms and mathematical and statistical formula. This made the process time consuming, tedious and inefficient. But in the modern days, it is become very much easy and efficient compared to the olden days by various python libraries, frameworks, and modules. Today, Python is one of the most popular programming languages for this task and it has replaced many languages in the industry, one of the reason is its vast collection of libraries. Python libraries that used in Machine Learning are:

- | | |
|-----------------|---------------|
| 1. Numpy | 6. Keras |
| 2. Scipy | 7. PyTorch |
| 3. Scikit-learn | 8. Pandas |
| 4. Theano | 9. Matplotlib |
| 5. TensorFlow | |

CHAPTER 5

SYSTEM ARCHITECTURE

Design is a multi- step that focuses on data structure software architecture, procedural details, procedure etc... and interface among modules. The design procedure also decode the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design change continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at relatively primary stage in its revolution.

Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However methods for software designs do exit, criteria for design qualities are existing and design notation can be applied

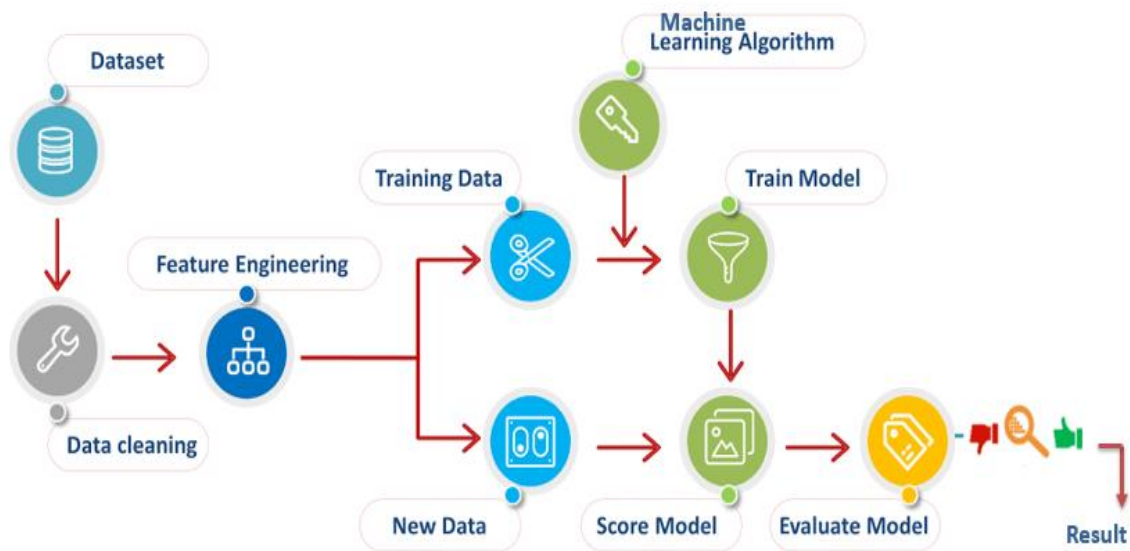


Fig 1 SYSTEM ARCHITECTURE

CHAPTER 6

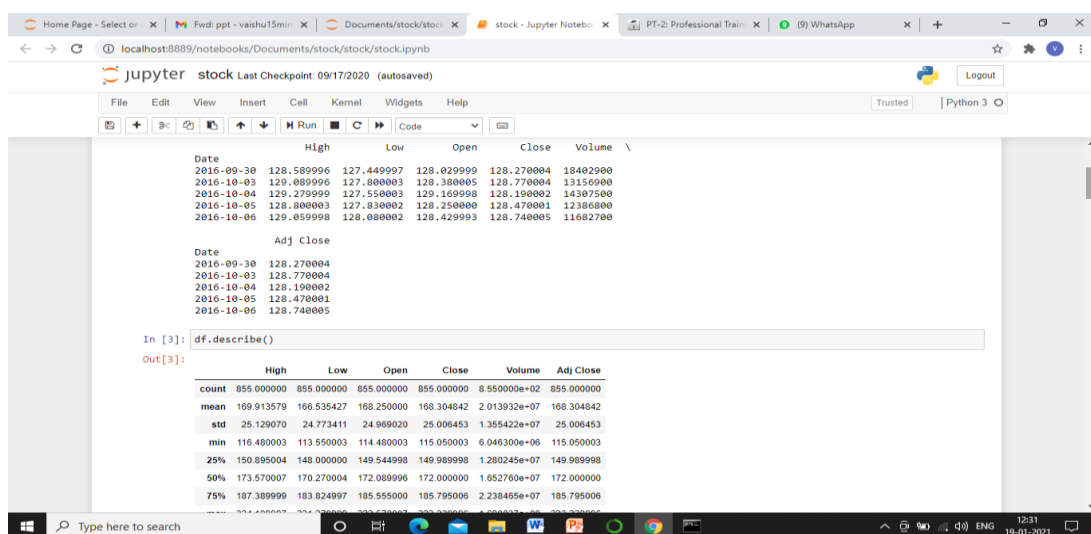
MODULE DESCRIPTION

6.1 LIST OF MODULES

- Collecting Dataset
- Data Cleaning
- Feature Engineering
 - Training Data
 - New Data
- Score model
- Evaluate model
- Result

6.2 COLLECTING DATASET

- Data Collection is one of the most important tasks in building a machine learning model.
- We collect the specific dataset based on requirements from internet.
- The dataset contains some unwanted data also. So first we need to pre-process the data and obtain perfect data set for algorithm.



The screenshot shows a Jupyter Notebook interface with a browser window at the top. The notebook has a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running cells, and code execution. The main area displays two code cells. The first cell contains a table of stock data with columns: Date, High, Low, Open, Close, Volume, and Adj Close. The data is for dates from 2016-09-30 to 2016-10-06. The second cell contains a code snippet: `In [3]: df.describe()`. The output of this code is displayed below, showing statistical summary for the columns: count, mean, std, min, 25%, 50%, and 75% for High, Low, Open, Close, Volume, and Adj Close.

	High	Low	Open	Close	Volume	Adj Close
count	855.000000	855.000000	855.000000	855.000000	8.550000e+02	855.000000
mean	169.913579	166.535427	168.250000	168.304842	2.013932e+07	168.304842
std	25.129070	24.773411	24.969020	25.006453	1.355422e+07	25.006453
min	116.480003	113.550003	114.480003	115.050003	6.046300e+06	115.050003
25%	150.895004	148.000000	149.544998	149.989998	1.280245e+07	149.989998
50%	173.570007	170.270004	172.089996	172.000000	1.652700e+07	172.000000
75%	187.389999	183.824997	185.555000	185.795006	2.238465e+07	185.795006

Fig 2 Collection of dataset

6.3 DATA CLEANING:

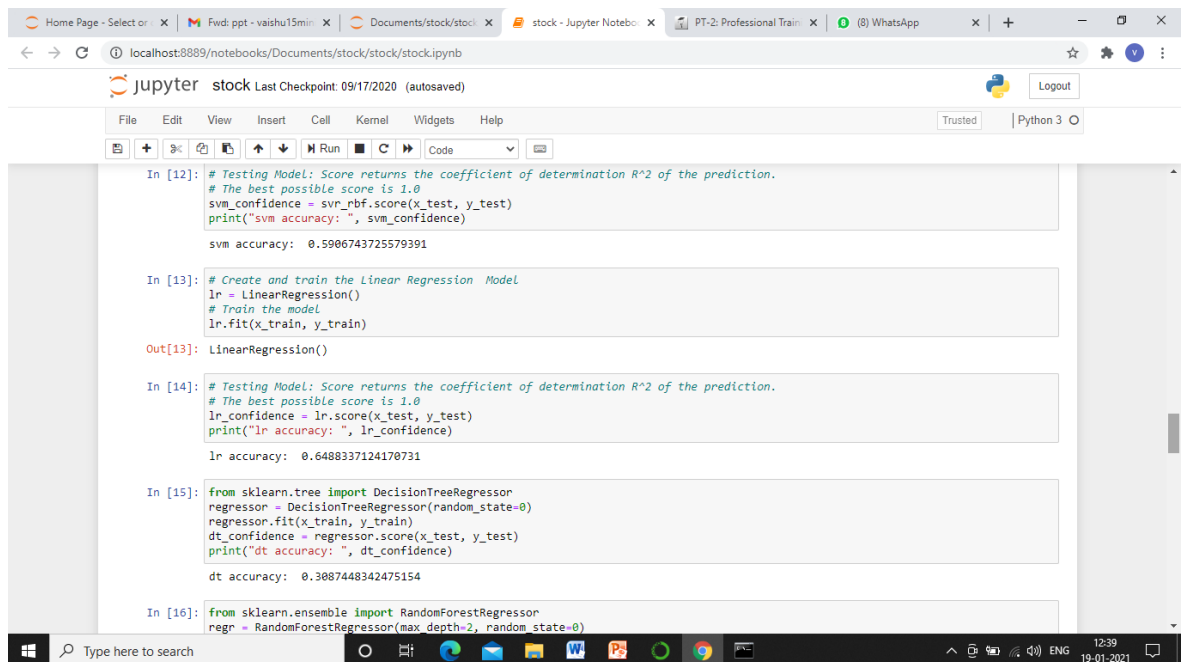
- Data cleaning includes Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.
- Data cleaning also includes some methods or functions which allow us to select the useful data for our system.

6.4 FEATURE ENGINEERING:

- Feature Engineering allows data which is trained data and new data and it divides the data into two types: training data and new data.

TRAINING DATA:

- Training data allows the data which we collected in the datasets and these collected data are trained using machine learning algorithm.
- Also predict the future using the trained data when we give the new data.



```
In [12]: # Testing Model: Score returns the coefficient of determination R^2 of the prediction.
# The best possible score is 1.0
svm_confidence = svm_rbf.score(x_test, y_test)
print("svm accuracy: ", svm_confidence)

svm accuracy: 0.5906743725579391

In [13]: # Create and train the Linear Regression Model
lr = LinearRegression()
# Train the model
lr.fit(x_train, y_train)

Out[13]: LinearRegression()

In [14]: # Testing Model: Score returns the coefficient of determination R^2 of the prediction.
# The best possible score is 1.0
lr_confidence = lr.score(x_test, y_test)
print("lr accuracy: ", lr_confidence)

lr accuracy: 0.648837124170731

In [15]: from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state=0)
regressor.fit(x_train, y_train)
dt_confidence = regressor.score(x_test, y_test)
print("dt accuracy: ", dt_confidence)

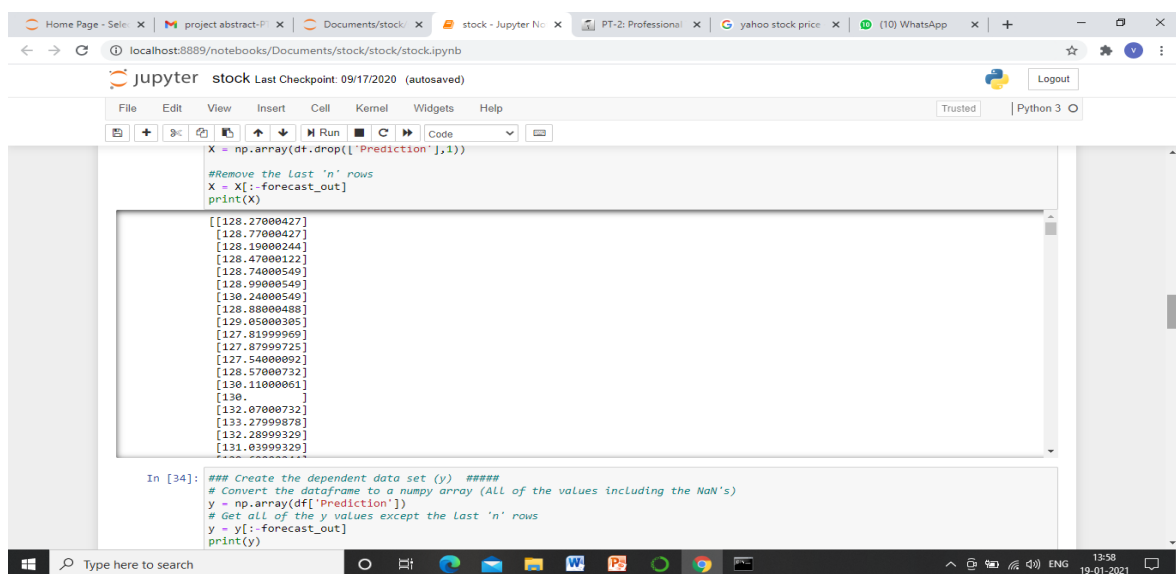
dt accuracy: 0.3087448342475154

In [16]: from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(max_depth=2, random_state=0)
```

Fig 3: Training set

NEW DATA:

- Dataset values converted into array values which is going to given to the algorithm to find accuracy.
- Select the algorithm based on the accuracy and analyse the data by using the machine learning algorithm.



The screenshot shows a Jupyter Notebook window titled 'stock' with a last checkpoint of 09/17/2020. The code in the cell is as follows:

```
X = np.array(df.drop(['Prediction'],1))

#Remove the Last 'n' rows
X = X[:-forecast_out]
print(X)
```

The output of the code is a list of 20 numerical values:

```
[[128.27000427]
 [128.77000427]
 [128.19000244]
 [128.47000122]
 [128.74000549]
 [128.99000549]
 [130.24000549]
 [128.88000488]
 [129.05000385]
 [127.81999669]
 [127.87999725]
 [127.54000092]
 [128.57000732]
 [130.11000061]
 [130.]
 [132.07000732]
 [133.27999878]
 [132.28999329]
 [131.05999329]
 [132.28999329]]
```

Below the output, the code for the next cell is visible:

```
In [34]: ### Create the dependent data set (y) #####
# Convert the dataframe to a numpy array (ALL of the values including the NaN's)
y = np.array(df['Prediction'])
# Get all of the y values except the Last 'n' rows
y = y[:-forecast_out]
print(y)
```

Fig 4 new data

6.5 SCORE MODEL:

- Score model is a major part because its role is to get data from new data and then the new data is trained by using train model.
- Then the new data is trained and we predict the future data by using datasets are trained in train model.
- Here, prediction accuracy is analysed by the data using machine learning algorithm.

6.6 EVALUATE MODEL:

- Evaluate model is used to predict whether the data prediction is right or wrong.
- And it calculates the accuracy of data which we gets trained and gets the accuracy of prediction using machine learning algorithm.

6.7 RESULT

- Based on that dataset we can get the result used our machine learning algorithm to predict the result.
- It will show the future values of particular stock.

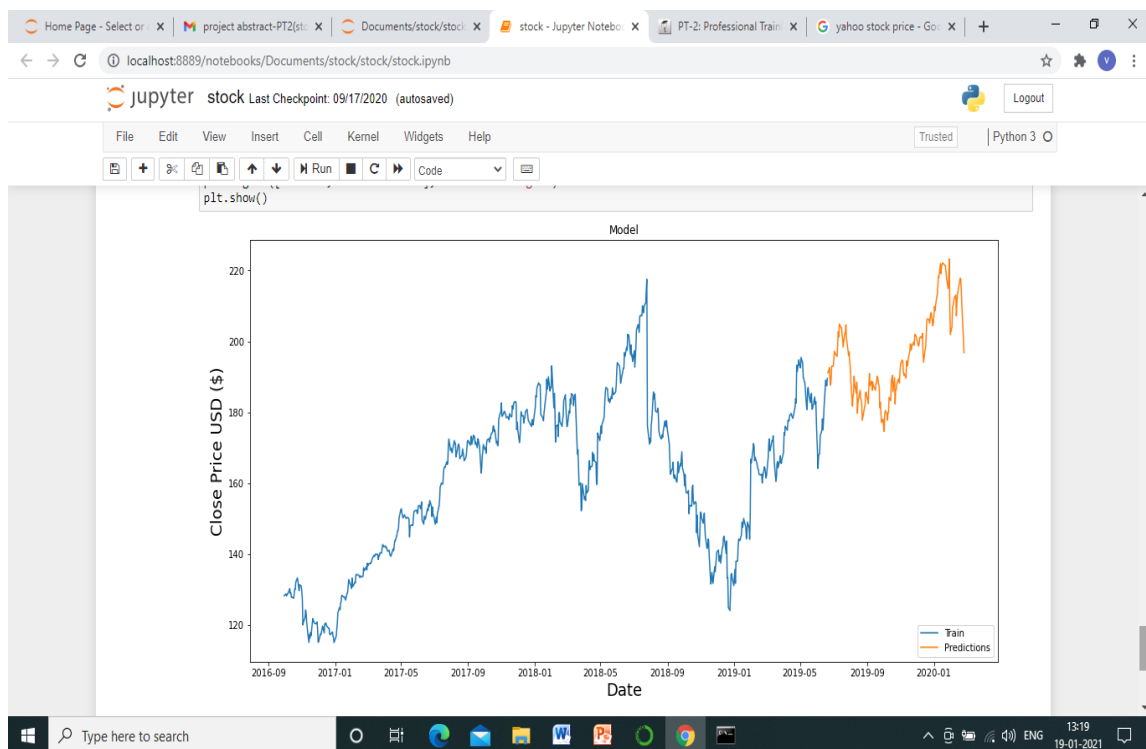


Fig 5: Prediction of stock price on graph

CHAPTER 7

CONCLUSION

This paper was an attempt to determine the future prices of the stocks of a company with greater accuracy and reliability using machine learning techniques. The primary contribution of the researchers being the application of the novel svm Model as a means of determining the stock prices. Both the techniques have shown an improvement in the accuracy of predictions, thereby yielding positive results with the svm model proving to be more efficient. The results are quite promising and has led to the conclusion that it is possible to predict stock market with more accuracy and efficiency using machine learning techniques. In the future, the accuracy of the stock market prediction system can be further improved by utilizing a much bigger dataset than the one being utilized currently. Furthermore, other emerging models of Machine Learning could also be studied to check for the accuracy rate resulted by them. Sentiment analysis though Machine Learning on how news affects the stock prices of a company is also a very promising area. Other deep learning based models can also be used for prediction purposes.

REFERENCES:

- [1] (2016). PhishMe Q1 2016 Malware Review. [Online]. Available: <https://phishme.com/project/phishme-q1-2016-malware-review/>
- [2] A. Belabed, E. Aimeur, and A. Chikh, "A personalized whitelist approach for phishing webpage detection," in Proc. 7th Int. Conf. Availability, Rel. Security (ARES), Aug. 2012, pp. 249–254.
- [3] Y. Cao, W. Han, and Y. Le, "Anti-phishing based on automated individual white-list," in Proc. 4th ACM Workshop Digit. Identity Manage., 2008, pp. 51–60.
- [4] T.-C. Chen, S. Dick, and J. Miller, "Detecting visually similar Web pages: Application to phishing detection," ACM Trans. Internet Technol., vol. 10, no. 2, pp. 1–38, May 2010.
- [5] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, "Clientside defense against Web-based identity theft," in Proc. 11th Annu. Netw. Distrib. Syst. Security Symp. (NDSS), 2004, pp. 1–16
- [6] C. Inc. (Aug. 2016). Couldmark Toolbar. [Online]. Available: <http://www.cloudmark.com/desktop/ie-toolbar>
- [7] J. Corbetta, L. Invernizzi, C. Kruegel, and G. Vigna, "Eyes of a human, eyes of a program: Leveraging different views of the Web for analysis and detection," in Proceedings of Research in Attacks, Intrusions and Defenses (RAID). Gothenburg, Sweden: Springer, 2014.
- [8] X. Deng, G. Huang, and A. Y. Fu, "An antiphishing strategy based on visual similarity assessment," Internet Comput., vol. 10, no. 2, pp. 58–65, 2006.
- [9] Z. Dong, K. Kane, and L. J. Camp, "Phishing in smooth waters: The state of banking certificates in the US," in Proc. Res. Conf. Commun., Inf. Internet Policy (TPRC), 2014, p. 16.

APPENDIX:

(A) SOURCE CODE:

```
# This program predicts stock prices by using machine learning models

#Install the dependencies

#import quandl

import numpy as np

import pandas_datareader as web

from sklearn.linear_model import LinearRegression

from sklearn.linear_model import LogisticRegression

from sklearn.svm import SVR

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt

#Get the stock data

#df = quandl.get("WIKI/AMZN")

df = web.DataReader('FB', data_source='yahoo', start='2016-10-01', end='2020-02-25')

print(df.head())

df.describe()

#import math

#data = df.filter(['Close'])

#dataset = data.values

#training_data_len = math.ceil( len(dataset) * .8 )

#raining_data_len

plt.figure(figsize=(16,8))

plt.title('Close Price History')

plt.plot(df['Close'])

plt.xlabel('Date', fontsize=18)
```

```

plt.ylabel('Close Price USD ($) ', fontsize=18)

plt.show()

# Get the Adjusted Close Price
####df = df[['Adj. Close']]

df = df[['Close']]

#Take a look at the new data

print(df.head())

# A variable for predicting 'n' days out into the future

forecast_out = 30 #n=30' days

#Create another column (the target or dependent variable) shifted 'n' units up

####df['Prediction'] = df[['Adj. Close']].shift(-forecast_out)

df['Prediction'] = df[['Close']].shift(-forecast_out)

#print the new data set

print(df.tail())

#### Create the independent data set (X) #####

# Convert the dataframe to a numpy array

X = np.array(df.drop(['Prediction'],1))

#Remove the last 'n' rows

X = X[:-forecast_out]

print(X)

#### Create the dependent data set (y) #####

# Convert the dataframe to a numpy array (All of the values including the NaN's)

y = np.array(df['Prediction'])

# Get all of the y values except the last 'n' rows

y = y[:-forecast_out]

print(y)

# Split the data into 80% training and 20% testing

```

```

x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=44)

# Create and train the Support Vector Machine (Regressor)

svr_rbf = SVR(kernel='rbf', C=1e3, gamma=0.1)

svr_rbf.fit(x_train, y_train)

# Testing Model: Score returns the coefficient of determination R^2 of the prediction.

# The best possible score is 1.0

svm_confidence = svr_rbf.score(x_test, y_test)

print("svm accuracy: ", svm_confidence)

# Create and train the Linear Regression Model

lr = LinearRegression()

# Train the model

lr.fit(x_train, y_train)

# Testing Model: Score returns the coefficient of determination R^2 of the prediction.

# The best possible score is 1.0

lr_confidence = lr.score(x_test, y_test)

print("lr accuracy: ", lr_confidence)

from sklearn.tree import DecisionTreeRegressor

regressor = DecisionTreeRegressor(random_state=0)

regressor.fit(x_train, y_train)

dt_confidence = regressor.score(x_test, y_test)

print("dt accuracy: ", dt_confidence)

from sklearn.ensemble import RandomForestRegressor

regr = RandomForestRegressor(max_depth=2, random_state=0)

regr.fit(x_train, y_train)

rf_confidence = regr.score(x_test, y_test)

print("rf accuracy: ", rf_confidence)

# Set x_forecast equal to the last 30 rows of the original data set from Adj. Close column

```

```

x_forecast = np.array(df.drop(['Prediction'],1))[-forecast_out:]

print(x_forecast)

# Print linear regression model predictions for the next 'n' days

lr_prediction = lr.predict(x_forecast)

print(lr_prediction)

# Print support vector regressor model predictions for the next 'n' days

svm_prediction = svr_rbf.predict(x_forecast)

print(svm_prediction)

import math

data = df.filter(['Close'])

dataset = data.values

training_data_len = math.ceil( len(dataset) * .8 )

training_data_len

train = df[:training_data_len]

valid = df[training_data_len:]

#valid['Predictions'] = x_train

#Visualize the data

plt.figure(figsize=(16,8))

plt.title('Model')

plt.xlabel('Date', fontsize=18)

plt.ylabel('Close Price USD ($)', fontsize=18)

plt.plot(train['Close'])

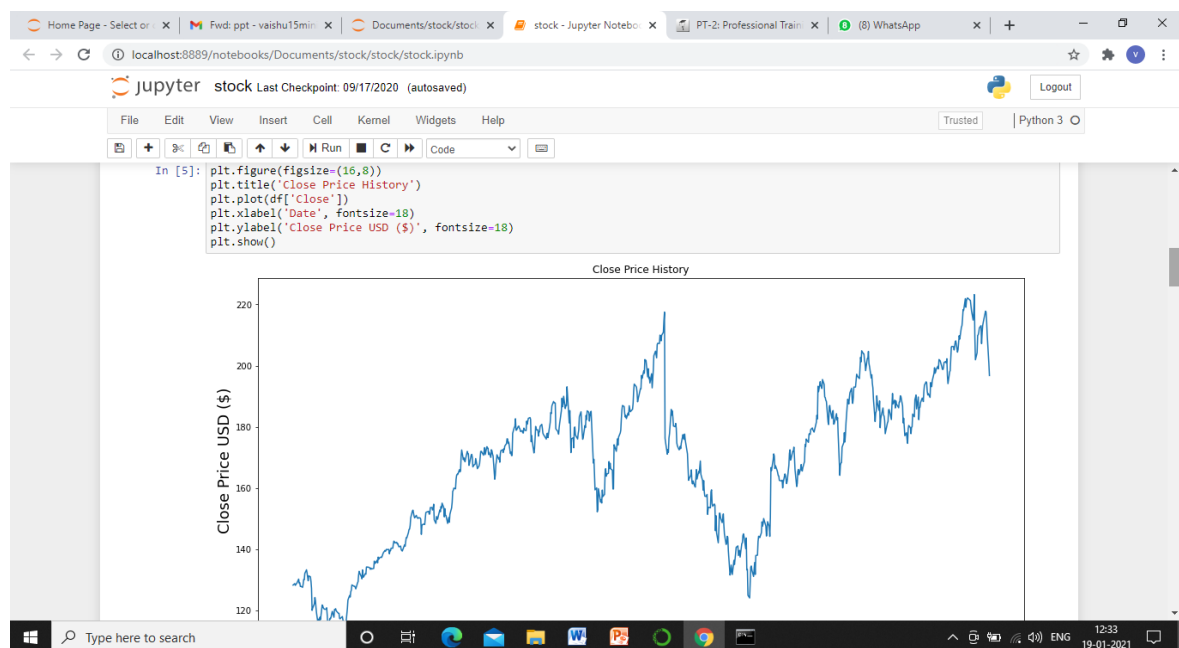
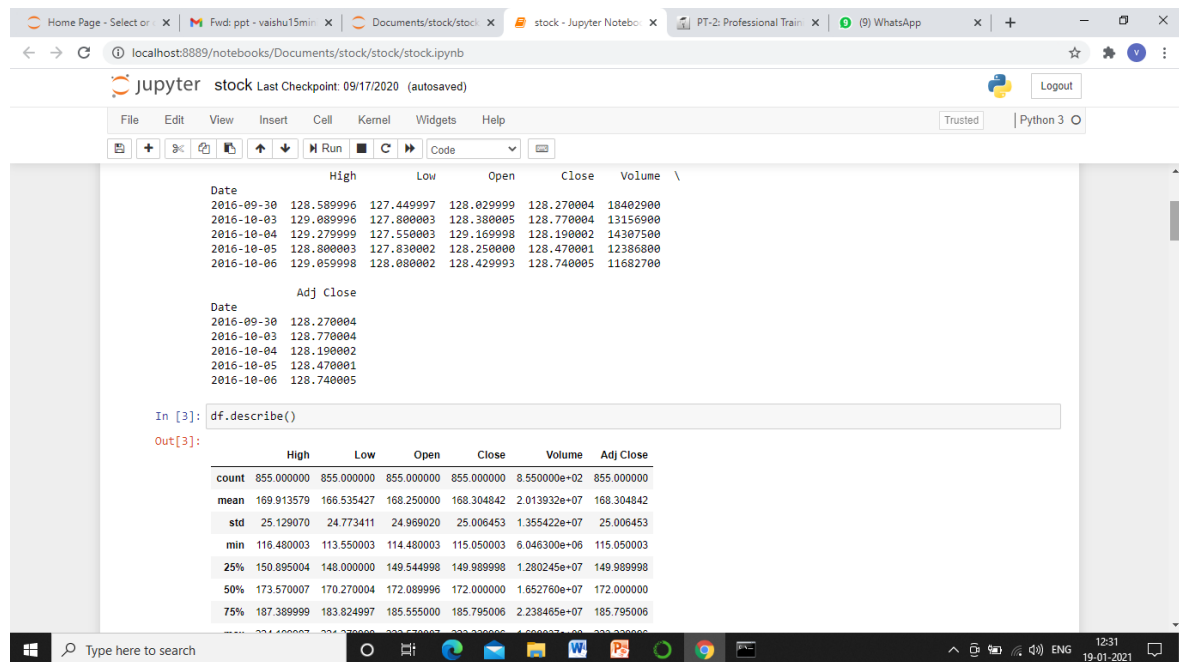
plt.plot(valid[['Close']])

plt.legend(['Train', 'Predictions'], loc='lower right')

plt.show()

```

(B) SCREENSHOTS:



Home Page - Select or X Fwd: ppt - vaishu15min X Documents/stock/stock X stock - Jupyter Notebo PT-2: Professional Train X (8) WhatsApp X + - X

localhost8889/notebooks/Documents/stock/stock/stock.ipynb

Jupyter stock Last Checkpoint: 09/17/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [6]:

```
# Get the Adjusted Close Price
###df = df[['Adj. Close']]
df = df[['Close']]
#Take a look at the new data
print(df.head())
```

Date	Close
2016-09-30	128.270004
2016-10-03	128.770004
2016-10-04	128.190002
2016-10-05	128.470001
2016-10-06	128.740005

In [7]:

```
# A variable for predicting 'n' days out into the future
forecast_out = 30 # 'n=30' days
#Create another column (the target or dependent variable) shifted 'n' units up
###df['Prediction'] = df[['Adj. Close']].shift(-forecast_out)
df['Prediction'] = df[['Close']].shift(-forecast_out)
#print the new data set
print(df.tail())
```

Date	Close	Prediction
2020-02-19	217.490005	NaN
2020-02-20	214.580002	NaN
2020-02-21	210.179993	NaN
2020-02-24	200.720001	NaN
2020-02-25	196.770004	NaN

Type here to search

12:35 19-01-2021

Home Page - Select or X Fwd: ppt - vaishu15min X Documents/stock/stock X stock - Jupyter Notebo PT-2: Professional Train X (8) WhatsApp X + - X

localhost8889/notebooks/Documents/stock/stock/stock.ipynb

Jupyter stock Last Checkpoint: 09/17/2020 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

In [8]:

```
### Create the independent data set (X) #####
# Convert the dataframe to a numpy array
X = np.array(df.drop(['Prediction'],1))

#Remove the Last 'n' rows
X = X[:-forecast_out]
print(X)
```

[128.27000427]
[128.77000427]
[128.19000244]
[128.47000122]
[128.74000549]
[128.99000549]
[130.24000549]
[128.88000488]
[129.05000305]
[127.81999969]
[127.87999725]
[127.54000092]
[128.57000732]
[130.11000061]
[130.]
[132.07000732]
[133.27999878]
[132.28999329]
[131.03999329]

In [9]:

```
### Create the dependent data set (y) #####
# Convert the dataframe to a numpy array (ALL of the values including the NaN's)
y = np.array(df['Prediction'])
```

Type here to search

12:36 19-01-2021

```
Home Page - Select or X | Fwd: ppt - vaishu15min X | Documents/stock/stock X | stock - Jupyter Notebo X | PT-2: Professional Train X | (8) WhatsApp X | + | - | X
localhost8889/notebooks/Documents/stock/stock/stock.ipynb
jupyter stock Last Checkpoint: 09/17/2020 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
In [9]: ### Create the dependent data set (y) #####
# Convert the dataframe to a numpy array (ALL of the values including the NaN's)
y = np.array(df['Prediction'])
# Get all of the y values except the Last 'n' rows
y = y[:-forecast_out]
print(y)

[119.01999664 115.08000183 117.19999695 116.33999634 117.79000092
 117.01999664 121.76999664 121.47000122 120.83999634 120.37999725
 120.41000366 120.87000275 118.41999817 115.09999847 115.40000153
 117.43000031 117.30999756 117.94999695 118.91000366 119.68000031
 117.76999664 120.30999756 120.20999908 120.56999969 119.87000275
 119.23999786 119.08999634 119.04000092 117.40000153 117.26999664
 118.01000214 116.91999817 116.34999847 115.05000305 116.86000061
 118.69000244 120.66999817 123.41000366 124.90000153 124.34999847
 126.08999634 126.62000275 128.33999634 127.87000275 127.91999817
 127.55000305 127.04000092 128.92999268 129.36999512 131.47999573
 132.77999878 132.17999268 130.97999573 130.32000732 133.22999573
 130.83999634 130.97999573 132.05999756 131.83999634 134.19999695
 134.13999939 134.19000244 134.05000305 133.85000061 133.44000244
 133.83999634 133.52999878 133.72000122 136.11999512 135.36000061
 135.44000244 136.41000366 135.53999329 137.41999817 136.75999451
 137.16999817 137.41999817 137.30000305 137.72000122 138.24000549
 138.78999329 139.60000061 139.32000732 139.72000122 139.99000549
 139.83999634 139.94000244 138.50999451 139.50999634 139.52999878
 140.33999634 140.32000732 141.75999451 142.6499939 142.41000366
 140.33999634 140.32000732 141.75999451 142.6499939 142.41000366]

In [10]: # Split the data into 80% training and 20% testing
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=44)

In [11]: # Create and train the Support Vector Machine (Regressor)
```

```
Home Page - Select or X | Fwd: ppt - vaishu15min X | Documents/stock/stock X | stock - Jupyter Notebo X | PT-2: Professional Train X | (8) WhatsApp X | + | - | X
localhost8889/notebooks/Documents/stock/stock/stock.ipynb
jupyter stock Last Checkpoint: 09/17/2020 (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted Python 3
In [12]: # Testing Model: Score returns the coefficient of determination R^2 of the prediction.
# The best possible score is 1.0
svm_confidence = svm_rbf.score(x_test, y_test)
print("svm accuracy: ", svm_confidence)

svm accuracy: 0.5906743725579391

In [13]: # Create and train the Linear Regression Model
lr = LinearRegression()
# Train the model
lr.fit(x_train, y_train)

Out[13]: LinearRegression()

In [14]: # Testing Model: Score returns the coefficient of determination R^2 of the prediction.
# The best possible score is 1.0
lr_confidence = lr.score(x_test, y_test)
print("lr accuracy: ", lr_confidence)

lr accuracy: 0.6488337124170731

In [15]: from sklearn.tree import DecisionTreeRegressor
regressor = DecisionTreeRegressor(random_state=0)
regressor.fit(x_train, y_train)
dt_confidence = regressor.score(x_test, y_test)
print("dt accuracy: ", dt_confidence)

dt accuracy: 0.3087448342475154

In [16]: from sklearn.ensemble import RandomForestRegressor
regr = RandomForestRegressor(max_depth=2, random_state=0)
```

