

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

SUPRIYA S
(Reg.No.37120075)



DEPARTMENT OF INFORMATION TECHNOLOGY SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

December 2020



SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

(Established under Section 3 of UGC Act, 1956)

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai -600119

www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **SUPRIYA S** (Reg.no.37120075) who carried out the project entitled “**STOCK MARKET PREDICTION**” under our supervision from JUL 2020 to DEC 2020.

Internal Guide

Dr.R.M.GOMATHI M.TECH.,Ph.D.

Head of the Department

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

Submitted for Viva voce Examination held on 20.01.2021.

Internal Examiner

External Examiner

DECLARATION

I, SUPRIYA S (Reg.No.37120075) hereby declare that the Project Report entitled **“STOCK MARKET PREDICTION”** done by me under the guidance of **Dr.R.M.GOMATHI M.TECH.,Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Information Technology .

DATE: 20.01.2021

PLACE: CHENNAI

A handwritten signature in black ink, appearing to read 'Supriya S', with a stylized flourish at the end.

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.

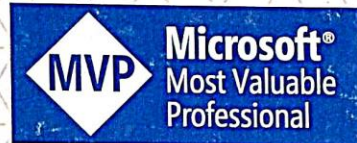
I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.R.M.GOMATHI M.TECH.,Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

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.



KaaShiv InfoTech

Shivanantha Building, X-41, 5th Floor, 2nd Avenue,
Anna Nagar, Chennai - 40 | www.kaashivinfotech.com



ORACLE

Certified Expert



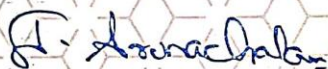
This is to Certify that

Mr./Ms. SUPRIYA.S
a Student of SATHYABAMA INSTITUTE OF
SCIENCE AND TECHNOLOGY has done the
Internship in our Company held from 29-11-2018
to 03-12-2018 and Completed the Training

Successfully.

KAASHIV INFOTECH
X-41, Shivanantha Building
5th Floor, 2nd Avenue
Anna Nagar, Chennai-600 044


Venkatesan Prabu.J
MANAGING DIRECTOR


Arunachalam.J
PROJECT MANAGER

***This Hands-On Training is Provided by 10 Years Microsoft Awarded Most Valuable Professional
- Recognized Top Azure Specialist in the World.



KaaShiv InfoTech

SOFTWARE DEVELOPMENT & ELECTRONICS / IOT RESEARCH COMPANY

X-41, shivanantha Building, 5th Floor, 2nd Avenue, Anna Nagar, Chennai - 40
www.kaashivinfotech.com



Microsoft®
Most Valuable
Professional

ORACLE®

Certified Expert

Certificate of Completion

Mr./Ms. SUPRIYA.S

a Student of SATHYARAMA INSTITUTE OF SCIENCE AND TECHNOLOGY

has done his/her In-Plant Training in Our Company held from 29-11-2018

to 03-12-2018 and completed the Training Successfully.

J. VENKATESAN PRABU
Managing Director

KAASHIV INFOTECH
X-41, Shivanantha Building
5th Floor, 2nd Avenue
Anna Nagar, Chennai-600 040
Ph: 044-42057542

J. ARUNACHALAM
Project Manager

***This Hands-On Training is Provided by 10 Years Microsoft Awarded Most Valuable Professional, 13 International Certified Expert - Recognized Top Azure Specialist in the World.

Abstract

The project aims to provide retail investors with a third-party investment mobile application to navigate through the stock market. This is achieved through the use of machine learning and mobile web technologies. Several stock price prediction approaches and models are developed including dense, feedforward neural networks, recurrent neural networks, simple linear regressions, and linear interpolations. Model architectures and hyperparameters are optimized and automatically searched by evolution algorithm. Promising results are found for trend prediction. The project serves as a foundation for democratizing machine learning technologies to the general public in the context of discovering investment opportunities. It paves the way for extending and testing out new models and developing Auto ML in the financial context in the future.

Table of Contents

CHAPTER NO	TITLE	PAGE NO
	Abstract	07
1. Introduction		
	1.1 Overview	10
	1.2 Objectives	10
	1.3 Application	11
2. Literature Survey		
	2.1 Stock Price Predictions	12
	2.2 Neural Network	12
	2.3 Evolution Algorithm	13
3. System Analysis		
	3.1 Existing System	14
	3.2 Proposed System	14
4. System Requirement		
	4.1 Hardware Requirement	15
	4.2 Software Requirement	15
	4.3 Python Technology and Platform	15
	4.4 Python Library	16
5. Module Description		
	5.1 List of Module	17
	5.2 Collection dataset	17
	5.3 Pre-processing	17
	5.4 Data input	18
	5.5 Result	18

6. Methodology	
6.1 Predicting Stock Price	19
6.2 Model	19
6.3 Performance Evaluation	19
6.4 Server	19
6.5 Application Implementation	20
7. System Architecture	21
8. Conclusion	23
9. References	24
10. Appendix	28
A. Projects Screenshots	28
B. Sample Code	30

CHAPTER – 1

Introduction

1.1 Overview

Retail investors spend a lot of time finding investment opportunities. Wealthier investors could seek professional financial advisory services, but for typical retail investors, the costs are prohibitive. Thus, retail investors have to figure out the market themselves and make informed decisions on their own. This makes investment very stressful in modern societies. Unfortunately, humans are irrational in nature. Without quantitative, data-driven models, decisions get swayed by cognitive biases or personal emotions, resulting in unnecessary losses. Even if investors are cautious enough, most do not have sufficient skills to process a huge volume of data required to make good judgments. Institutional investors rely on sophisticated models supported by technologies to avoid traps, but retail investors do not have access to such technologies and often find themselves falling behind the market. Inspired by the increasing popularity of deep learning algorithms for forecasting application, these algorithms might serve as potential tools to find hidden patterns in the trend of stock prices, this information could be useful to provide extra insights for retail investors when making investment decisions.

1.2 Objectives

The ultimate goal of our application is to serve retail investors as a third-party investment tool that uses machine learning to help them navigate in the fast-changing stock market. The project aims to introduce and democratize the latest machine learning technologies for retail investors. No prediction is 100% accurate. Therefore, the upper bound and lower bound of the stock prices will be displayed to illustrate the trading range the investors should be looking at. This application serves as a supplementary quantitative tool for investors to see the market at a different perspective with the help of technology. This project is divided into 2 parts, namely a research component and an application component, aiming to provide retail investors with stock price predictions using different machine learning models in a good user experience way for reference.

1.3 Application

This project aims to provide stock price predictions based on the latest machine learning technologies to all retail investors. A mobile web application is developed to provide predictions in an intuitive way. Different models' performance and accuracy can also be compared. The application also serves as another user interface (UI) in visualizing results from the research apart from Jupiter notebooks with lots of tables and graphs.

CHAPTER – 2

Literature Survey

2.1 Stock Price Predictions

From the research paper “Machine Learning in Stock Price Trend Forecasting” written by Y. Dai and Y. Zhang in Stanford University, they used features like PE ratio, PX volume, PX EBITDA, 10-day volatility, 50-day moving average, etc. to predict the next-day stock price and a long-term stock price [2]. The machine learning algorithms used in the research are Logistic Regression, Gaussian Discriminant Analysis, Quadratic Discriminant Analysis, and SVM. The accuracy ratio is defined as the number of days that the model correctly classified the testing data over the total number of testing days. With the short-term model predicting the next day stock price, it has very low accuracy, the Quadratic Discriminant Analysis is the best among all models, it scored a 58.2% accuracy. With the long-term model predicting the next n days stock prices, the longer the time frame, the better in the accuracy for SVM. With a time, window of 44 days, the SVM model's accuracy reached 79.3%. Apart from that, it was found that by increasing the number of features, the accuracy increased. When all of the 16 features were used, the accuracy of the model reached 79%, while it fell to 64% when only 8 features were used, and 55% if only 1 feature was used. Our project will also investigate how the timeframe would affect the accuracy of price predictions of different models. As models have to reach a certain threshold to have significance for the users to work as a reference, it is essential for us to optimize our model to figure out what the optimal parameters and model structure are for our stock price prediction purpose.

2.2 Neural Network

A neural network attempts to learn a function that maps the input features to the output predictions, serving as a universal function approximator. It consists of a network of neurons, each of which represents a weighted sum of inputs. Outputs from neurons are fit into activation functions which introduce non-linearity to the system, and then passed to some other neurons. In a typical dense feedforward neural network, the network consists of layers of neurons stacked together, with neurons between individual layers fully connected. Optimization of neural networks is usually done through backpropagation

with gradient descent, which essentially propagates the error from the output layer back to the input layer, while computing the gradient of the error against each parameter in the process.

2.3 Evolution Algorithm

Researches have shown that large-scale evolution can auto-generate neural network model architectures and hyperparameters with performance comparable with state-of-the-art human-designed models. In a research in 2017, a large-scale evolution for discovering image classification neural networks was run. It started with a huge population of randomized simple 1-layer models, then slowly evolved the population by removing a poor model and generating a new model by mutating some parameters of a good model in each iteration. After hundreds of hours of running the algorithm with huge computing power, most models in the population achieved state-of-the-art results on CIFAR datasets. In each iteration, only a simple mutation that changed 1 parameter was applied, which allowed searching in a large search space. The paper showed the possibility of finding good models by using lots of computational power to replace human machine learning experts and has set the foundation of democratizing machine learning with Auto ML.

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 Requirement

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

Module Description

5.1 List Of Modules

- Collecting Dataset
- Pre-processing
 - Data cleaning
 - Data transformation
 - Data selection
- Data input
- Result

5.2 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.

5.3 Pre-Processing

- It is the gathering of task related information based on some targeted variables to analyse and produce some valuable outcome.
- However, some of the data may be noisy, i.e. may contain inaccurate values, incomplete values or incorrect values.
- Hence, it is must to process the data before analysing it and coming to the results.
- Data pre-processing can be done by data cleaning, data transformation, data selection.
- Data cleaning includes Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies.

- Data transformation may include smoothing, aggregation, generalization, transformation which improves the quality of the data.
- Data selection includes some methods or functions which allow us to select the useful data for our system.

5.4 Data Input

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

5.5 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.

CHAPTER – 6

Methodology

6.1 Predicting Stock Price

When predicting stock price, the saved model will first be loaded. Then, a feature vector specified by the input options is built with the build predict dataset script, which is the same as the build training dataset except it returns a flatten 1D feature vector. The feature vector is inputted into the model to predict stock price. For 10-day predict, the predictions are directly outputted. For 1-day predict, the predicted stock price is appended to the raw dataset as if it happened before, then a new feature vector is generated for predicting the stock price for the day after, the process is repeated to predict the stock prices for all next 10 days.

6.2 Model

Different common neural network models are tested, including dense neural network, simple recurrent neural networks (RNNs), long short-term memory networks (LSTMs) and gated recurrent unit networks (GRUs). Different model architectures are tested by changing the number of hidden layers, the number of hidden units per hidden layer, and the activation function or recurrent activation function used in each hidden layer.

6.3 Performance Evaluation

As mentioned above, apart from the mean squared error that a model tries to minimize, different finance-specific scores are introduced to evaluate and compare performance of different models, namely model accuracy score, model trend score and stock buy/sell score. The scores are also designed to convey useful and meaningful messages to help investors understand a stock and make investment decisions.

6.4 Server

For local development and testing, the Flask micro web framework is used to serve local saved data like raw stock price data and saved predictions file. It is written in Python and integrates well with the existing backend architecture. Moreover, it has a much

more lightweight interface than other popular Python web frameworks such as Django and allows easier implementation of application programming interfaces.

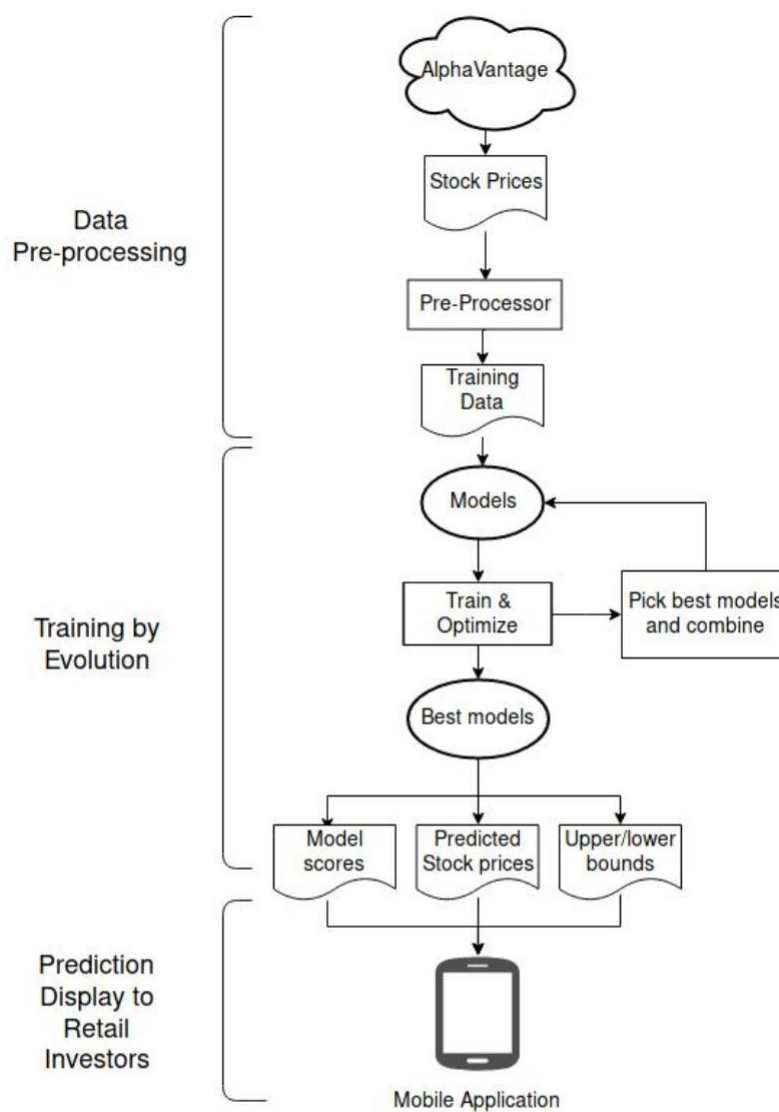
6.5 Application Implementation

Company information is collected from the IEX Stock API. A Firebase Cloud Function is written to get the data from the API and store it in Firebase Cloud Firestore, which the application will access through another Firebase Callable Cloud Function.

CHAPTER – 7

System Architecture

The architecture of the system follows a client-server model, where the server and the client are loosely coupled.



System Architecture Diagram

After relevant stock data are retrieved from the third-party data provider through the cloud, the backend pre-processes the data and builds the models. After that, predictions are made and the prediction results will be stored on another cloud, which can be retrieved from the mobile application. The advantages of the loosely coupled architecture include improved scalability and ease of collaboration. The workload for the cloud which serves the models and the one which serves the mobile application will be very different. One cloud serves the model prediction results, which are simple text files; another cloud serves the mobile application with a lot of rich user content such as images and large UI libraries. Having two clouds to adapt to two different demand patterns is more efficient, especially since cloud providers these days usually serve content on demand. Also, the separation allows different team members in the team to focus on different parts after agreeing on a common interface. It speeds up development as team members responsible for different parts of the system do not need to take care of the underlying implementation details. Also, it is easier to swap out different components, e.g., to replace the models the team could simply make changes to the backend, while the frontend remains unaffected.

CHAPTER – 8

Conclusion

The project lays the foundation for democratizing machine learning technologies for retail investors, connecting predictions made by machine learning models to retail investors through a mobile application. It helps investors navigate through the stock markets with additional analysis and help them make more informed decisions. The findings demonstrated that the application provides significance in trend prediction. When compared to the baseline, the prediction shows useful trend tendency with the real stock trend. Through the application interface, the user can easily compare the predictions and model scores from different machine learning models, then choosing the one that fits their preference. The models used in the application will continue to improve itself by searching for a better model topology, structure and hyperparameters through evolution algorithm. The findings concluded the usefulness of evolution algorithm in lowering the mean squared error when predicting stock prices, which is helpful for improving the trend prediction for retail investors. Therefore, with the application and research findings, to large extent the project team achieved the aim of creating an user-friendly system for retail investors whom does not have previous technical knowledge to navigate the machine model predictions result with useful benchmarks. There are 4 possible further improvements building upon the findings of this project. First, multiple approaches to framing the problems could be explored in the future, such as predicting whether the stock price goes up or down (binary classification) based on the previous stock prices. Other features could be incorporated, such as market news and sentiment. Combined with the development of more advanced machine learning techniques, the accuracy of the information provided to retail investors might be improved significantly.

CHAPTER – 9

References

- [1] “Survey Finds Hong Kong Securities Market Attracts Wide Range of Investors,” *HKEX*, 13 Jul2017; http://www.hkex.com.hk/news/news-release/2017/170713news?sc_lang=en.
- [2] Y. Dai and Y. Zhang, “Machine Learning in Stock Price Trend Forecasting,” *Stanford University*; [http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.p df](http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf).
- [3] J. Patel, S. Shah, P. Thakkar, and K. Kotecha, “Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques,” *Expert Systems with Applications: An International Journal*, Vol. 42, Jan. 2015, pp. 259-268
- [4] B. Wanjawa and L. Muchemi, “ANN Model to Predict Stock Prices at Stock Exchange Markets,” arXiv:1502.06434 [q-fin.ST], 2014
- [5] D. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction*, Wiley, 2001
- [6] R. Williams and D. Zipser, “Gradient-based learning algorithms for recurrent networks and their computational complexity”, in *Back-propagation: Theory, Architectures and Applications*, Hillsdale, NJ: Erlbaum, 1992, pp. 433 - 486
- [7] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, “Gradient flow in recurrent nets: the difficulty of learning long-term dependencies”, in *A Field Guide to Dynamical Recurrent Neural Networks*, S. C. Kremer and J. F. Kolen, eds., IEEE press, 2001
- [8] S. Hochreiter and J. Schmidhuber, “Long Short-term Memory”, *Neural Computation*, vol. 9, no. 8, pp. 1735 - 1780, 1997
- [9] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation”, arXiv:1406.1078 [cs.CL], 2014

- [10] W. Gail, G. Yoav, and Y. Eran, "On the Practical Computational Power of Finite Precision RNNs for Language Recognition", arXiv:1805.04908 [cs.NE], 2018
- [11] J. Chung, C. Gulcehre, K. Cho and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling". arXiv:1412.3555 [cs.NE]. 2014
- [12] E. Real, et al., "Large-Scale Evolution of Image Classifiers," arXiv:1703.01041 [cs.NE]. Jun 2017.
- [13] D. Alajbeg, Z. Bubas and D. Vasic, "Price Distance To Moving Averages And Subsequent Returns", *International Journal of Economics, Commerce and Management*, Vol. V, Dec 2017, pp. 33 - 47
- [14] Progressive Web Apps, Google. Available:
<https://developers.google.com/web/progressive-web-apps/>
- [15] Neoteric, "Single-page application vs. multiple-page application", Medium. 2016.
Available: <https://medium.com/@NeotericEU/single-page-application-vs-multiple-page-application-2591588efe58>
- [16] Keras: The Python Deep Learning library, Keras. Available: <https://keras.io/>.
- [17] Documentation of scikit-learn 0.19.2.
Available: <http://scikit-learn.org/stable/documentation.html>.
- [18] Alpha Vantage API Documentation, Alpha Vantage.
Available: <https://www.alphavantage.co/documentation/>.
- [19] NumPy v1.14 Manual, The SciPy community.
Available: <https://docs.scipy.org/doc/numpy-1.14.5/>.

[20] pandas: powerful Python data analysis toolkit, Pandas.

Available: <http://pandas.pydata.org/pandas-docs/version/0.23/>.

[21] Hello, Colaboratory - Colaboratory - Google, Google.

Available: <https://colab.research.google.com>.

[22] TensorBoard: Visualizing Learning, Google.

Available: https://www.tensorflow.org/guide/summaries_and_tensorboard.

[23] Welcome to Flask — Flask 1.0.2 documentation, Pallets Team.

Available: <http://flask.pocoo.org/docs/1.0/>.

[24] Cloud Storage, Google. Available: <https://firebase.google.com/docs/storage/>.

[25] Getting Started, The Investors Exchange.

Available: <https://iextrading.com/developer/docs/>.

[26] Cloud Functions for Firebase, Google.

Available: <https://firebase.google.com/docs/functions/>.

[27] Cloud Firestore, Google. Available: <https://firebase.google.com/docs/firestore/>.

[28] React – A JavaScript library for building user interfaces, Facebook Inc;

<https://reactjs.org/>.

[29] React Router: Declarative Routing for React.js, React Training.

Available: <https://reacttraining.com/react-router/web/guides/philosophy>.

[30] Read Me - Redux, Redux. Available: <https://redux.js.org/>.

[31] Immutable collections for JavaScript, Facebook Inc.

Available: <https://github.com/facebook/immutable-js/>.

[32] Introduction - Material Design, Google.

Available: <https://material.io/design/introduction/>.

[33] Material UI, Material UI Team. Available: <https://material-ui.com/>

[34] Using Google Charts, Google.

Available: <https://developers.google.com/chart/interactive/docs/>.

[35] Facebook Login, Facebook.

Available: <https://developers.facebook.com/docs/facebook-login/>

[36] Firebase Authentication, Google.

Available: <https://firebase.google.com/docs/auth/>

[37] unittest — Unit testing framework, Python Software Foundation.

Available: <https://docs.python.org/3.6/library/unittest.html>

[38] E. Real, A. Aggarwal, Y. Huang, Q. Le, “Regularized Evolution for Image Classifier Architecture Search,” arXiv:1802.01548 [cs.NE]. Feb 2018.

[39] GitHub features: the right tools for the job, Github Inc; <https://github.com/features>.

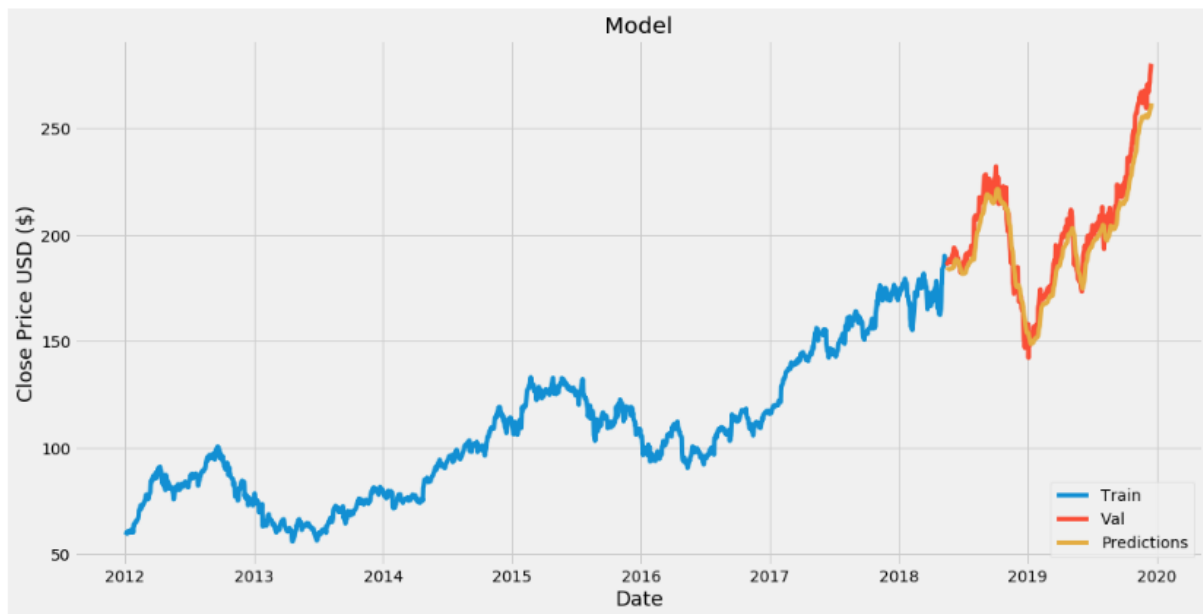
CHAPTER – 10

Appendix

Projects Screenshots

	High	Low	Open	Close	Volume	Adj Close
Date						
2012-01-03	58.928570	58.428570	58.485714	58.747143	75555200.0	51.115936
2012-01-04	59.240002	58.468571	58.571430	59.062859	65005500.0	51.390648
2012-01-05	59.792858	58.952858	59.278572	59.718571	67817400.0	51.961189
2012-01-06	60.392857	59.888573	59.967144	60.342857	79573200.0	52.504375
2012-01-09	61.107143	60.192856	60.785713	60.247143	98506100.0	52.421093
...	...	...	...	...	...	...
2019-12-11	271.100006	268.500000	268.809998	270.769989	19689200.0	270.769989
2019-12-12	272.559998	267.320007	267.779999	271.459991	34327600.0	271.459991
2019-12-13	275.299988	270.929993	271.459991	275.149994	33396900.0	275.149994
2019-12-16	280.790009	276.980011	277.000000	279.859985	32046500.0	279.859985
2019-12-17	281.769989	278.799988	279.570007	280.410004	28539600.0	280.410004





	Close	Predictions
Date		
2018-05-17	186.990005	183.353653
2018-05-18	186.309998	183.596985
2018-05-21	187.630005	183.603516
2018-05-22	187.160004	183.663666
2018-05-23	188.360001	183.685089
...	...	...
2019-12-11	270.769989	257.086731
2019-12-12	271.459991	257.929352
2019-12-13	275.149994	258.822205
2019-12-16	279.859985	260.060303
2019-12-17	280.410004	261.837372

400 rows × 2 columns

Sample 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()
```