A MAJOR-PROJECT REPORT

ON

"PREDICTIVE ANALYSIS OF MARKET TRENDS FOR STOCK PRICE PREDICTIONS USING MACHINE LEARNING AND DEEP LEARNING"

Submitted to

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

DEEMED TO BE UNIVERSITY

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

PRABUDDHA KUMAR GHOSH	1405049
PRADEEP KUMAR	1405050
PRASHANT KUMAR	1405051
PRINCE JHA	1405053

UNDER THE GUIDANCE OF
PROF. SUDAN JHA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

2017-2018

A MAJOR-PROJECT REPORT

ON

"PREDICTIVE ANALYSIS OF MARKET TRENDS FOR STOCK PRICE PREDICTIONS USING MACHINE LEARNING AND DEEP LEARNING"

Submitted to

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

DEEMED TO BE UNIVERSITY

In Partial Fulfillment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

PRABUDDHA KUMAR GHOSH	1405049
PRADEEP KUMAR	1405050
PRASHANT KUMAR	1405051
PRINCE JHA	1405053

UNDER THE GUIDANCE OF
PROF. SUDAN JHA

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAR, ODISHA - 751024

2017-2018

Kalinga Institute of Industrial Technology

Deemed to be University
School of Computer Engineering
Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"Predictive Analysis of Market Trends for Stock Price Predictions using Machine Learning and Deep Learning"

submitted by

PRABUDDHA KUMAR GHOSH	**1405049**
PRADEEP KUMAR	**1405050**
PRASHANT KUMAR	**1405051**
PRINCE JHA	**1405053**

is a record of bonafide work carried out by them, in the partial fulfillment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science) at KIIT, Deemed to be University, Bhubaneswar. The work has been done during the year 2017-2018, under the guidance of us.

Date: / /

(Prof. SUDAN JHA)
Project Guide

(Prof. PINAKI CHATTERJEE)
Project Coordinator

ACKNOWLEDGEMENT

We, as a team, have worked together on building this project from scratch. A lot of thinking has been done in designing the project before the actual implementation has taken place. We have taken the aid and assistance of different resources available across the internet for gaining the domain knowledge regarding the the development of the project. For building this project, we had to gain some knowledge regarding stock market. We got inspired by the work of different youtubers, along with researchers who were doing some research work on stock price prediction using Machine Learning and Deep learning.

We learnt a great deal about Machine Learning from the online available courses which helped us build a strong base for building this project. We have taken the help of a course on Udemy regarding machine learning. For Deep learning, we have taken the assistance of Google Datacamp.

We are grateful to our project guide **Prof. Sudan Jha**, School of Computer Science Engineering for providing proper guidance and guiding us in the correct direction and always being there as a support system. Lastly, we would remiss in our duties if we do not thank our parents who have been a constant moral support. Also, we would like to thank our parents who have supported us and provided assistance during the development of this project.

Prabuddha Kumar Ghosh
Pradeep Kumar
Prashant Kumar
Prince Jha

ABSTRACT

The project aims at building a predictive model of market trends for stock price predictions using Deep learning and Machine learning. There are a lot of factors such as past prices,sentiment analysis, brand value among others which can be used in harmony to predict the results with a high accuracy rate. We have implemented different algorithms to achieve highest accuracy.The factors taken into consideration are opening price and Google trends value. Implemented machine learning algorithms are Linear Regressor, SGD Regressor, Decision Tree Regressor, Random Forest Regressor and KNN Regressor. Besides, we have also implemented Deep Neural Networks Algorithm.

Keywords: Stock prediction; machine learning; deep learning; predictive analysis; regression; random forest; neural networks

Contents

List of Figures

Chapter 1

Introduction

Figure 1.1: STOCK MARKET

1.1 OUTLINE

People have been investing in stock markets for a long time. People have become very rich, when stock price of a particular company they have invested in rises and vice-versa. However, people often faced difficulties in handling stock market because they were unable to respond quickly to changes. This is where computers came in. Different algorithms were used to learn the behavior of stock market which was later utilized to predict the outcomes.

1.2 STOCK PRICE FLUCTUATION

Stock prices are directly proportional to the demand. If the demand of any product increases, the stock price or market value of that product increases. Similarly, if the demand of any product falls, we will face a sudden fall in the stock price of the product. To make a profit by investing in the stock market, we need to analyze and understand the time of the fall and rise of the stock market. To be able to do this, we need to have a detailed study of the behaviour of the stock market history. However, we have to keep in mind that if the stock market behaved in a certain way to the occurrence of a particular event, it is not necessary that the stock market will behave in the exact similar pattern this time. To say the least, stock market is unpredictable. However, we can study and provide the best available results.

1.3 STOCK PRICE DETERMINATION

At any given instance, stock is dependent on demand,as we have stated earlier. Moreover, it is also dependent on the supply. Suppose, there is a demand for a particular product. For the stock to soar, it is necessary that there must be enough supply to fulfill the requirement. So, to determine the stock price of a product in the near future, we need to analyze and understand what will be the demand for that particular product and also the corresponding supply. If the supply and demand of the product is healthy, the stock price will also be very high.

1.4 PROBLEM STATEMENT

Can we predict stock prices with the aid of machine learning? Investors make educated guesses by analyzing data. They read the news, company history, industry trends among others to make an educated decision. There are lots of features that go into making a decision. Top firms like Morgan Stanley, Citigroup, D.E. Shaw hire quantitative analysts to build predictive models. We are now living in the age of algorithms. Records of prices for traded commodities go back thousands of years. In Finance, the field of quantitative analysts is about 25 years old and even now it is still not fully accepted, understood or widely used.

1.5 MACHINE LEARNING

Machine learning has become the state of the art computer science field in recent years. It provides computers the ability to learn from the data and predict the outcome based on the user input without being explicitly programmed. Machine learning does an excellent work on finding the pattern that exist between the data. These patterns can vary from data set to data set and the machine learning algorithms does an eloquent job in finding them provided apt algorithm is chosen to build the model.

1.6 MOTIVATION

Machine learning has been playing an important role in helping grow a business. The organizations which are using new technologies like Machine learning are yielding superior results as compared to other organizations. Thus, more and more organizations are now moving towards incorporating machine learning into their work. Leading companies like JP Morgan are deeply involved with Machine Learning now. JP Morgan uses Machine learning in multiple areas:

- Companies that provide card services use Machine Learning to aim their advertisements at different card holders.
- Trading groups use it for better risk modelling, client modelling etc.
- Transaction processing groups use it for deriving insights of flow of funds, catching illicit transfers etc.

With the help of machine learning, the machines get the power of self-learning. Machines can study the data provided to them, and from that data, machines learn different patterns without being hard coded for the same. This technology is at its initial phases and there is a huge scope in the future. So, being in the prime of our learning curve, we were very much interested in knowing how this new technology in being used to yield better results.

1.7 ABOUT

We will train our machine learning model using data sets consisting of historical data.

Our model will consider the following features which will be discussed in detail later:

- Google Trends
- Historical Price
- It will visualize the data over a time series.
- It would predict the upcoming prices on a range of dates.

1.8 CONTRIBUTION

In the field of stock market analysis, not much work has been done with the aid of machine learning. Machine learning is one new technology which has been paying dividends in many new fields where it is being applied. Stock market prediction is a very unpredictable and difficult task since it depends on a series of factors. Slight change in the factors can result in fall or rise of the stock market. We think using machine learning in stock market analysis can prove to be a major game changer and we would like to develop the foundation so that more work is done in this field. When we test more and more algorithms in this field, we will be able to identify which algorithm is giving us the best results and we can apply it in analysis and prediction of stock markets.

Chapter 2

Literature Survey

1. A Hybrid Machine Learning System for Stock Market Forecasting by Rohit Choudhry, and Kumkum Garg have used a hybrid machine learning system based on Genetic Algorithm (GA) and Support Vector Machines (SVM) for stock market prediction. It was suggested that instead of taking only technical indicators, if we took economic and political conditions into account, we would obtain better results. The results showed that the GA-SVM system gives better results than the stand alone SVM system. [1]

2. Forecasting Stock Market Short-term trends using neuro-fuzzy based methodology by George S. Atsalakis and Kimon P. Valavanis proposed a system which is composed of an adaptive neuro fuzzy inference system controller used to control the stock market. Obtained results are a huge improvement on the previous Efficient Market Hypothesis.[2]

3. Predicting direction of stock price index movement using artificial neural networks and support vector machines by Yaup Kara et al attempted to develop two efficient models and compared their performances in predicting the direction of movement in stock market. Experimental results showed that Artificial Neural Networks yielded better results than that of SVM Model.[3]

4. ML-KNN:A lazy learning approach to multi-label learning by Min-Ling Zhang et al uses text categorization to work on several topics at the same time. Euclidean metric is used to calculate the distance between two instances. This is a new multi-label form of KNN. [4]

5. Using artificial neural network models in stock market index prediction by Erkam Guresen et al evaluates the effectiveness of neural network models on stock market prediction. The models which were analysed were Multi-layer perceptron (MLP), Dynamic artificial neural network (DAN2) and the hybrid neural networks. The results indicate that classical ANN model MLP outperforms DAN2 and GARCH-MLP with little difference.[5]

6. Credit rating analysis with support vector machines and neural networks by Zan Huang et al has used Support Vector Machines with Back propagation Neural Network as a benchmark. It has obtained around 0.8 accuracy for both BNN and SVM methods. A comparative study has been made to understand the contribution of each of the factors in the obtained result.[6]

7. Forecasting stock market movement direction with support vector machine by Wei Huang et al has compared the performance of Linear Discriminant Analysis, Quadratic Discriminant Analysis and Elman Back propagation Neural networks. It has been observed that SVM outperforms other classification methods. A model combining SVM with other classification models was proposed.[7]

8. Forecasting returns: New European evidence by Steven J. Jordon et al tries to predict the aggregate returns using a wide range of predictors. It was observed that macro and technical predictors statistically improve forecast accuracy and generate gains for investors. It was also observed that simple forecast combinations consistently yield substantial benefits both in forecast accuracy and economic gain.[8]

9. Stock Market Prediction Using Artificial Neural Networks by Birgul Egeli et al aimed at using artificial neural networks to get predictions on Istanbul Stock Exchange (ISE) market index value. The features taken into consideration were previous days index value, previous days TL/USD exchange rate, previous days overnight interest rate and some dummy variables to represent the week days. Six ANN models and two MA models were implemented. Results showed that ANN models were more accurate than MA models. Among the ANN models, GFF Network Model provided the best result.[9]

10. Deep Learning for Event-Driven Stock Prediction by Xiao Ding et al suggested a deep learning method for stock market prediction driven by events. Events were collected from news articles and were trained using neural tensor network. After this, a convolutional neural network was used to get the influences of short-term and long-term events on variations in the stock prices. Results displayed that discrete event-based method results were surpassed by event embedding-based document.

11. A fusion model of HMM, ANN and GA for stock market forecasting by MD. Rafiul Hassan et al proposed to use a fusion model by putting together a Hidden Markov Model (HMM), Artificial Neural Networks (ANN) and Genetic Algorithms (GA) to forecast financial market behaviour. Results indicated that the predicted values of the new tool was better than previously implemented model which used only a single HMM.[11]

12. Genetic algorithms approach to feature discretization in articial neural networks for the prediction of stock price index by Kyoung-jae Kim et al suggested a Genetic Algorithm approach to feature seperation and the finding out the weights for Artificial Neural Networks (ANN) to get the price index. Limitations of previously implemented methods have been properly mitigated by using a hybrid of GA and ANN. One limitation of this approach is that the processing elements number in the hidden layers is kept static at 12. [12]

13. A LSTM-based method for stock returns prediction: A case study of China stock market by Kai Chen et al aimed at demonstrating the power of LSTM in stock market predictions. This model showed improvements ranging from 14.3 percentage to 27.2 percentage over random prediction method. Future scope of this approach include the inclusion of MACD and other features in the learning feature sets and evaluation their contributions.[13]

14. Volatility forecasting and risk management for commodity markets in the presence of asymmetry and long memory by Walid Chkili et al explored the relevance of asymmetry and long memory in modelling and forecasting market risk. Results indicated that volatility of commodity returns can be better described by non-linear volatility models. Results also show that precious metals differ largely from crude and oil in terms of modelling returns and asymmetry. [14]

15. Stock Market Prediction System proposed by Takashi Kimoto, Kazuo Asakawa, Morio Yoda and Masakazu Takeoka suggests a time to buy and sell stock on the Tokyo Stock Exchange. It is themed upon modular neural networks. They selected the said application to check whether neural networks could develop a successful model in which their predicting abilities could be used for stock market prediction. The prediction system displayed accurate results.[15]

16. Stock Prediction using Numerical and Textual Information proposed by Ryo Akita, Akira Yoshihara, Takashi Matsubara and Kuniaki Uehara proposes an application of deep learning models, Paragraph Vector, and Long Short-Term Memory (LSTM), to financial time series forecasting. They proposed a method which changes articles in the newspaper into their distributed representations using Paragraph Vector and certain models the effects of events that happened in the past on opening prices with LSTM. Results showed that LSTM was able to capture the influence of time series on input data better than the other models.[16]

17. The study made by WERNER F. M, De Bondt and Richard Thaler on market efficiency investigates whether the overreaction of people to unexpected and dramatic news events affects stock prices. Results showed that the data depends on the hypothesis of overreaction.[17]

18. Predicting Stock Market Trends using Recurrent Deep Neural Networks proposed by Akira Yoshihara, Kazuki Fujikawa, Kazuhiro Seki and Kuniaki Uehara proposes an approach to market trend prediction based on RNN-RBM to model effects of past events and combine it with Deep Belief Network(DBN).[18]

19. A study conducted by George.S.Atsalakis and Kimon.P.Valvanis surveys more than 100 published articles that focuses on neural and neuro-fuzzy techniques and applied to forecast stock markets using Soft-Computing methods. The research contributed to the classification of soft-computing techniques applied to different stock markets that may be used for further analysis and evaluation. [19]

20. B.Wuthrich, V. Cho, S. Leung, D. Permunetilleke, K. Sankaran, J. Zhang and W. Lam proposed a model that uses Data Mining techniques to forecasts Daily Stock Market from Textual Web Data. The technique exploited textual information along with numeric time series data in order to increase the quality of input. They used several learning techniques like rule-based, nearest neighbour and neural networks to predict the outcomes and then these techniques are analyzed with each other. Results showed that Rule based technique proved to be the best and yields surprisingly good results for this comparatively difficult application. [20]

Chapter 3

Software Requirements Specification

3.1 OPERATING SYSTEM

- Linux
- Windows

3.2 PROGRAMMING LANGUAGES AND FRAMEWORKS

- Python
- Pandas
- Scikit Learn
- Numpy
- Matplotlib
- Keras

3.3 ALGORITHMS USED

Machine Learning has become the state of the art computer science field in recent years. It provides computers the ability to learn from the data and predict the outcome based on the user input without being explicitly programmed. Deep learning is one of the major subfield of machine learning which tries to mimic the actual brain. Deep learning algorithms are built on neural networks with many hidden layers.

- Linear Regression
- Random Forest
- Decision Tree Regression
- SGD Regressor
- KNN Regression
- Deep Neural Networks

3.4 DESIGN TOOLS

- Flow Chart
- Class Diagram

Chapter 4

Requirement Analysis

4.1 FUNCTIONAL REQUIREMENTS

- Historical price of stocks
- Google Trend Values of the stock

4.2 NON-FUNCTIONAL REQUIREMENTS

- System Requirement -The system should have decent computing power to train the machine learning model.
- User-Friendly -The software should have a user-friendly interface so that the application use should be impromptu.
- Reliability -The system should be reliable and the model should be re-trained on the new data regularly.
- Performance -The response time of the server to any request should be quick and at the same time provide user with the accurate results.
- Independent-The different modules being used should be minimally dependent on each other as much as possible.

Chapter 5

System Design

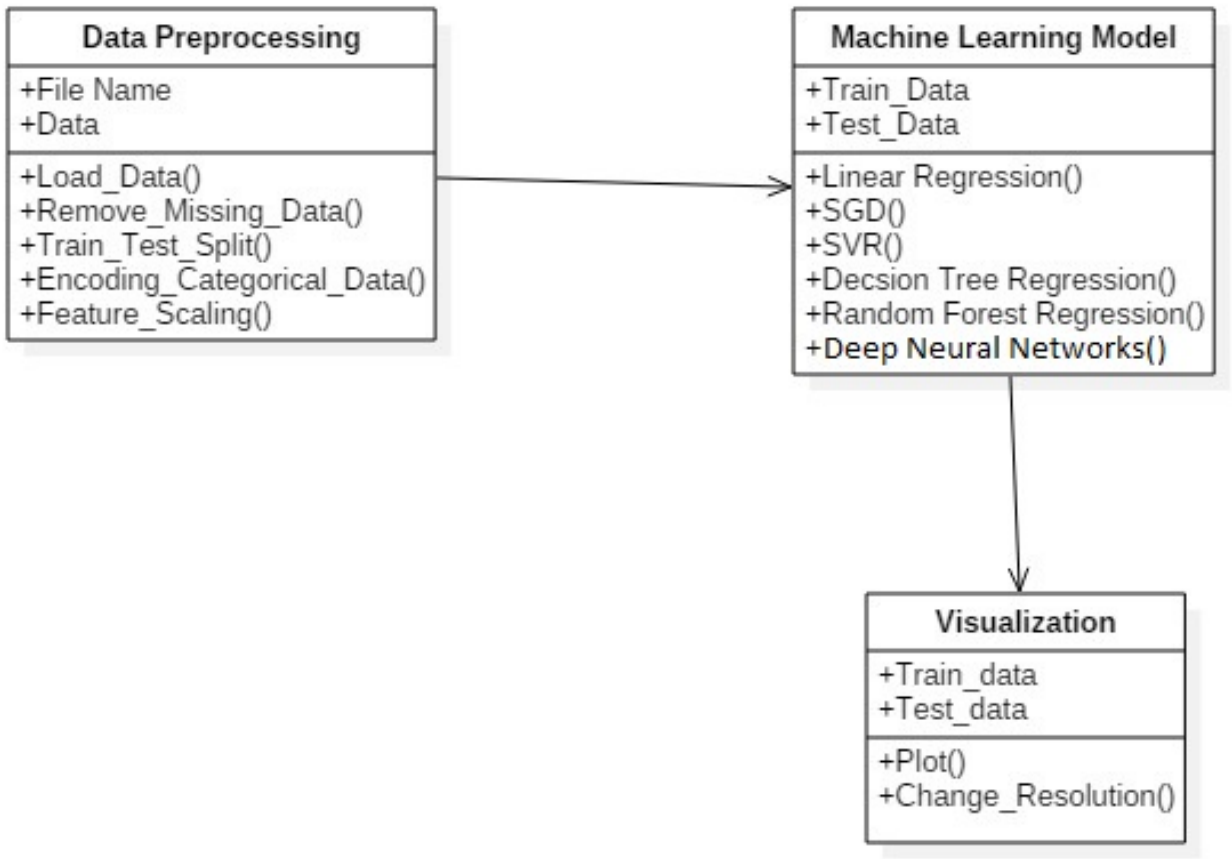

Figure 5.1: CLASS DIAGRAM

Chapter 6

System Testing

The dataset is divided into train and test data and then we match the actual values with the predicted values. For simplicity, we are attaching only ten actual and their corresponding predicted values for each algorithm we have performed.

6.1 Linear Regression

Linear Regression is a machine learning algorithm in which one variable is dependent while other variables are independent. In order to apply this algorithm, there should be a linear dependency between the dependent and independent variable.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by linear regression algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	114.9825627	-0.344456847
2016-12-16	116.470001	115.0076082	-1.255596108
2016-12-19	115.800003	115.4340115	-0.316054828
2016-12-20	116.739998	115.4590571	-1.097259656
2016-12-21	116.800003	115.4841026	-1.126627026
2016-12-22	116.349998	115.5091481	-0.722690085
2016-12-23	115.589996	115.5341936	-0.04827615
2016-12-27	116.519997	115.927098	-0.508838839
2016-12-28	117.519997	115.9521435	-1.334116355
2016-12-29	116.449997	115.977189	-0.406018044
			-0.715993394

6.2 Stochastic Gradient Descent

It is a unsupervised machine learning algorithm in which the gradient of mini-batch of samples is calculated rather than taking the entire dataset at a time. It is computationally very faster in comparison to Batch Gradient Descent.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by SGD regressor algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	108.972979	-5.552970937
2016-12-16	116.470001	109.1948491	-6.246374034
2016-12-19	115.800003	113.0789682	-2.349770924
2016-12-20	116.739998	113.300838	-2.945999708
2016-12-21	116.800003	113.5227078	-2.805903353
2016-12-22	116.349998	113.7445776	-2.239295612
2016-12-23	115.589996	113.9664474	-1.404575358
2016-12-27	116.519997	117.536018	0.871971358
2016-12-28	117.757887	117.7578877	0.20242572
2016-12-29	116.449997	117.9797575	1.313662979
			-2.115682987

6.3 Decision Tree Test

It is a supervised learning method that can be used for both classification as well as regression problems. In this, each node represents a test on an attribute, each branch represents the output of the test and each leaf node represents class label. The model created by it predict the value of target variable by learning some rules inferred from data model.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by decision tree regressor algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	113.6999985	-1.45605698
2016-12-16	116.470001	113.6999985	-2.378296966
2016-12-19	115.800003	113.6999985	-1.813475342
2016-12-20	116.739998	116.5613327	-0.153045488
2016-12-21	116.800003	116.5613327	-0.204341005
2016-12-22	116.349998	116.5613327	0.181637047
2016-12-23	115.589996	116.5613327	0.840329383
2016-12-27	116.519997	116.5613327	0.035475198
2016-12-28	117.519997	116.5613327	-0.815745681
2016-12-29	116.449997	116.5613327	0.09560816
			-0.566791167

6.4 Random Forest Test

Random Forest is the collection of several decision trees and all the trees are different in their structure. Here the training data is not randomize instead the algorithm is randomized. Random Forest algorithm has set a benchmark in ensemble learning method.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by random forest regressor algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	113.6076988	-1.536053255
2016-12-16	116.470001	113.6076988	-2.457544583
2016-12-19	115.800003	115.4498973	-0.302336521
2016-12-20	116.739998	117.1778973	0.375106482
2016-12-21	116.800003	117.0562996	0.219432015
2016-12-22	116.349998	117.5805982	1.057671011
2016-12-23	115.589996	117.6141993	1.751192465
2016-12-27	116.519997	117.0686974	0.470906638
2016-12-28	117.519997	115.189399	-1.983150153
2016-12-29	116.449997	114.4034015	-1.757488667
			-0.416226457

6.5 KNN Regressor Test

KNN is a machine learning algorithm which utilizes the simplicity of Euclidean Distance for calculating the distance of K nearest neighbor. The value of K is chosen by the programmer and depends on his/her intuition. In KNN Regressor the predicted value is the average of K nearest neighbor values.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by KNN regressor algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	111.844999	-3.063787564
2016-12-16	116.470001	111.8079995	-4.002748742
2016-12-19	115.800003	115.1919984	-0.525047137
2016-12-20	116.739998	115.1919984	-1.326023322
2016-12-21	116.800003	115.6539987	-0.98116804
2016-12-22	116.349998	116.5509987	0.172755224
2016-12-23	115.589996	117.1169991	1.321051261
2016-12-27	116.519997	114.105999	-2.071745676
2016-12-28	117.519997	114.105999	-2.905035813
2016-12-29	116.449997	114.105999	-2.012879399
			-1.539462921

6.6 Deep Neural Networks

Neural Networks are very effective to identify hidden features which other algorithms miss out. In this project we implemented Deep Neural Network which contains multiple hidden layers with Input and Output layers and activation function as Rectified Linear Unit(ReLU). The Deep Learning model performed extremely well and higher accuracy was achieved in comparison to other Machine Learning models. The reason for higher accuracy for Deep Learning is the ability of Deep Neural Network to capture the relations between the price and the features affecting it. DNN captures the relation between different seasons in a year and how the price of stock gets affected during a particular season or month.

The actual data in the table is from the test dataset corresponding to a particular date. The predicted data is the value predicted by deep learning algorithm on test data. Percentage change represents the percentage of absolute difference between predicted data and actual data.

Date	**Actual Values**	**Predicted Values**	**Percentage Change**
2016-12-15	115.379997	115.669861	0.251225522
2016-12-16	116.470001	116.075905	-0.338366959
2016-12-19	115.800003	116.603333	0.693721916
2016-12-20	116.739998	116.560379	-0.153862432
2016-12-21	116.800003	116.415207	-0.329448622
2016-12-22	116.349998	116.456017	0.091120758
2016-12-23	115.589996	116.491814	0.780186894
2016-12-27	116.519997	116.820786	0.258143673
2016-12-28	117.519997	116.895782	-0.531156412
2016-12-29	116.449997	116.943314	0.423629895
			0.114519423

6.7 Error Function

The error is calculated in each of the algorithm using the formula:

$$ErrorPercentage : ((PredictedValue - ActualValue)/ActualValue) * 100$$

```
def calculateError(predicted,actual):
    n = len(predicted)
    s = 0
    for i in range(n):
        s += ((predicted[i]-actual[i])/actual[i]*100)
    return s/float(n);
```

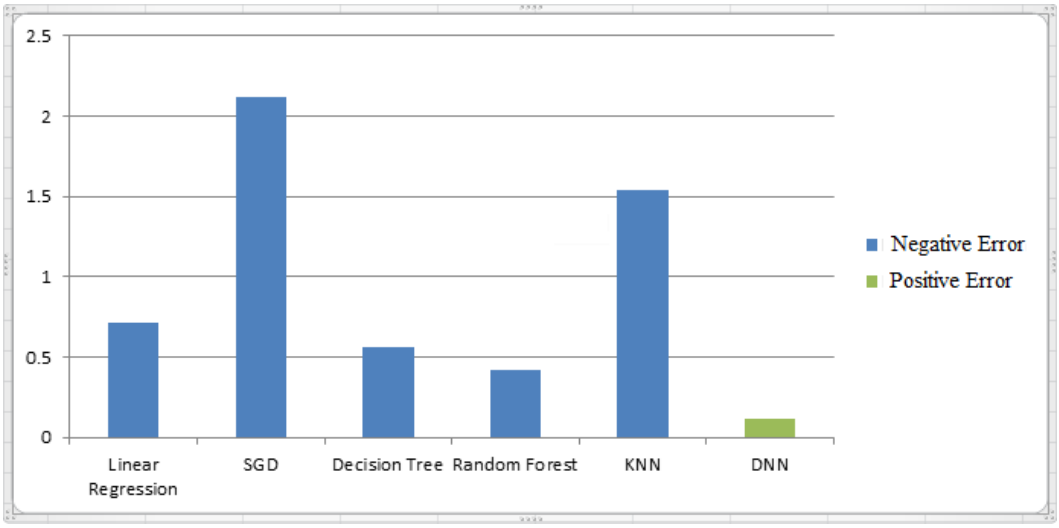

Figure 6.1: ERROR VALUE (COLUMN BAR)

The column bar graph helps us to understand the error in the predicted value over the actual value. Smaller the column, better is the results obtained. Error can be either positive or negative: Positive indicating that predicted value is greater than the actual value and negative indicates that the actual value is greater than the predicted value. Green indicates positive error, while blue indicates negative error.

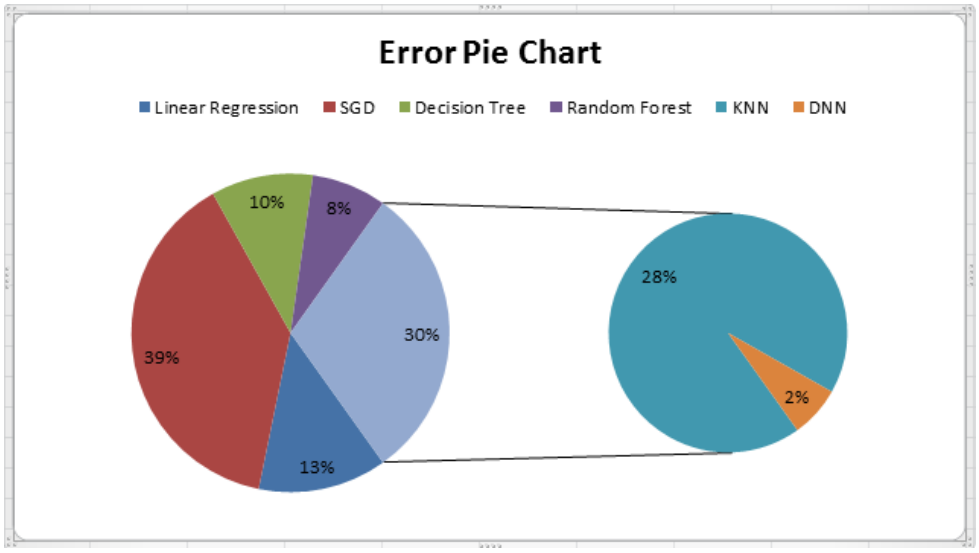

Figure 6.2: ERROR VALUE (PIE CHART)

If the cumulative error is represented as a pie, we can understand what part of the total error is resulted by a particular algorithm.
From the obtained graphical results, we can see that the least error is found in Deep Neural Networks. Here, error is found to be 0.114519423. The error in DNN is positive. All other errors obtained from the regression algorithms are found to be negative.

Following Deep neural networks, the least error among the regression algorithms is found in Linear Regression, followed by Decision Tree. Error in Random Forest is comparatively higher, while the error is even more in SGD. Error value is maximum while implementing KNN algorithm.

Algorithm	**Error Percentage**
Linear Regression	-0.715993394
Stochastic Gradient Descent	-2.115682987
Decision Tree	-0.566791167
Random Forest	-0.416226457
K-Nearest Neighbours	-1.539462921
Deep Neural Networks	0.114519423

Chapter 7

Project Planning

7.1 COMPONENTS AND WORKING

The main components are:

- Data Processing
- Data Visualization
- Applying algorithms
- Predicting future prices

7.2 WORK FLOW

Work Flow is as follows:

- Dataset is prepared using historical data and google trend values.
- Data is be visualized.
- The machine learning model is trained on the above stated dataset.
- The model is able to predict the future prices.

Chapter 8

Implementation

8.1 DATA PREPROCESSING

The dataset contains following columns:

- Date - Date
- Open - Opening price of particular date
- High - Highest price at particular date
- Low - Lowest price at particular date
- Close - Closing price at particular date
- Adj Close - Adj.Close price at particular date
- Volume - Volume of stock traded
- From all the columns we need Date and Open for our research work. We extracted Date and open from the table to make our new dataset.
- Date is in the format YYYY-MM-DD so we need to break it in numerical form i.e, YY,MM,DD. We decompose the Date column into three columns.

New Dataset Structure Before Decomposition of Date:

Date(YYYY-MM-DD) (Date Object)	**Open (Float)**

After Decomposition of Date:

Year(YYYY) Integer	**Month(MM) Integer**	**Day(DD) Integer**	**Open (Float)**

- Dataset contains data from Year 2007 to 2016. We split the data into train and test data. Train dataset contains data from range 2007-2014 and Test Dataset contains from 2015-2016.
- The Date here is the features and the Open price is our target values which needs to be predicted. X(features) = [Year,Month,Day], y(target) = [Open]
- Feature scaling has been done on data for faster convergence rate of algorithms and also to maintain standardisation in data. This concludes our data preprocessing.
- Function to plot graph is written which takes dataset as parameters and plots the graph.
- To train the model different Regression algorithms have been used.
- The Train data is fitted into the algorithm and the model is trained.
- The models accuracy is then tested by giving Test data as input and evaluating the predicted result against the known value from Test data.
- The graph is plotted between Actual value vs Predicted value.

Below is the flow chart for above procedure:

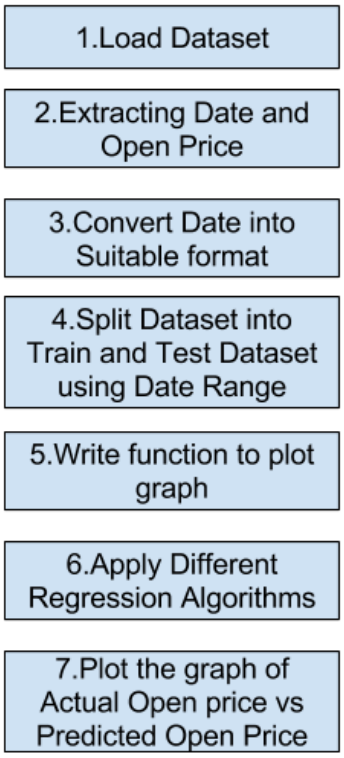

Figure 8.1: FLOW CHART

8.2 ALGORITHMS

- Algorithm for pre-processing of data

<table>
<tr><td colspan="2">Algorithm 1: Pre-processing of stock dataset and extracting relevant features</td></tr>
<tr><td colspan="2">Input: Dataset (Date, Open, High, Low, Close, Adj Close, Volume)
Initialize: Download the dataset from yahoo finance and Load the dataset
Output: New Dataset containing Day, Month, Year, Open Price</td></tr>
<tr><td>1.</td><td>Load the dataset

Dataset contains following fields

Date ← Date
Open ← Opening price at particular date
High ← Highest price at particular date
Low ← Lowest price at particular date
Close ← Closing price at particular date
Volume ← Volume of stock traded</td></tr>
<tr><td>2.</td><td>Select only relevant features and drop others

Dataset representation

<table><tr><td>Date (yyyy-mm-dd) (Date Object)</td><td>Open Price (Float)</td></tr></table></td></tr>
<tr><td>2.</td><td>If a column is empty</td></tr>
<tr><td>3.</td><td>Remove the row containing that column</td></tr>
<tr><td>4.</td><td>End If</td></tr>
<tr><td>5.</td><td>Split the date into day, month, year

New Dataset Representation

<table><tr><td>Year (yyyy)
Integer</td><td>Month (MM)
Integer</td><td>Day (DD)
Integer</td><td>Open (Float)</td></tr></table></td></tr>
<tr><td>7.</td><td>Save the dataset</td></tr>
</table>

Figure 8.2: PREPROCESSING OF THE DATA SET

- Algorithm for processing Google trends data set

Algorithm 2: *Processing of Google trend dataset*	
Input: *Dataset (Date, Trend Value)* ***Initialize:*** *Download the dataset from google and Load the dataset* ***Output:*** *Dataset Containing Day, Month, Year, Open Price, Trend Value*	
1.	*Load the dataset*
2.	***If*** *a column is empty*
3.	*Remove the row containing that column*
4.	***End If***
5.	***If*** *a date is missing*
6.	*Create a new row with trend value as previous value*
7.	***End If***
8.	*Combine the Stock Price dataset with current dataset*
9.	*Save the dataset*

Figure 8.3: PROCESSING OF GOOGLE TRENDS DATA SET

- Algorithm for creation of machine learning models

Algorithm 3: *Create machine learning model and predict the value*	
Input: *Dataset (Day, Month, Year, Open Price, Trend Value)* ***Initialize:*** *Load the Dataset* ***Output:*** *Predicting future 15 days values with percentage error and graph to visualize*	
1.	*Load the dataset*
2.	*Split the dataset into train and test set* *Train Set* ← *120 days of data* Test Set ← 15 days of data
3.	*Feature Scaling* *Formula X' = (X – Xmin) / (Xmax – Xmin)* *Where,* *X'* *Scaled Value* *X* ← *Feature* *Xmin* ← *Minimum value of feature X* *Xmax* ← *Maximum value of feature X*
4.	*Choose a Machine Learning Model*
5.	*Initialize the Model*
6.	*Train the model on training dataset*
7.	*Test the model on Test dataset*
8.	*Compare the predicted value of test data from the actual value*
9.	*Find the total error*
10.	*Visualize the data*

Figure 8.4: CREATION OF MODEL TO PREDICT THE VALUES

8.2.1 LINEAR REGRESSION

Linear Regression is a machine learning algorithm in which one variable is dependent while other variables are independent. In order to apply this algorithm, there should be a linear dependency between the dependent and independent variable.

```
from sklearn.linear_model import LinearRegression,Ridge
from sklearn.metrics import explained_variance_score
model_linear_reg = LinearRegression()
model_linear_reg.fit(X_train,y_train)
df_lr = pd.DataFrame(index=df_test.index)
df_lr['Actual'] = y_test
df_lr['Predicted'] = model_linear_reg.predict(X_test)
print("% Change : "+str(calculateError(df_lr['Predicted'],df_lr['Actual'])))
plotDataframe(df_lr,"Linear Regression")
```

Listing 8.1: Code snippet for linear regression

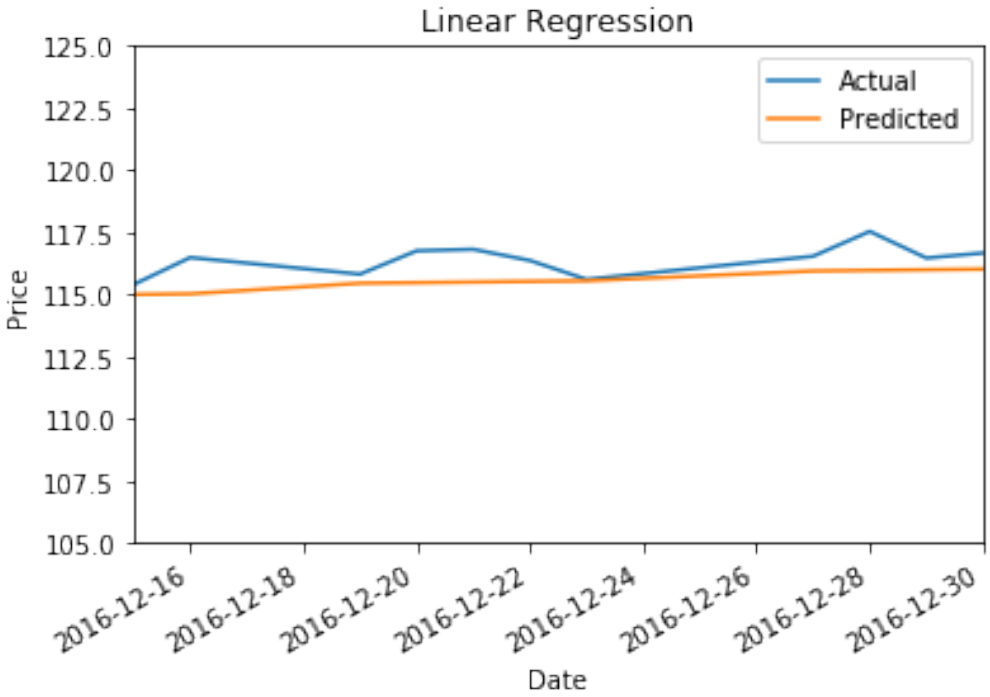

Figure 8.5: LINEAR REGRESSION GRAPH

Error: Margin of error ranges from **-0.04827615** to **-1.334116355**.

8.2.2 SGD REGRESSOR

It is a unsupervised machine learning algorithm in which the gradient of mini batch of samples is calculated rather than taking the entire data set at a time. It is computationally very faster in comparison to Batch Gradient Descent.

```
from sklearn.linear_model import SGDRegressor
model_dt = SGDRegressor(max_iter=1000,alpha=0.0001)
model_dt.fit(X_train,y_train)
df_sgd = pd.DataFrame(index=df_test.index)
df_sgd['Actual'] = y_test
df_sgd['Predicted'] = model_dt.predict(X_test)
print("% Change : "+str(calculateError(df_sgd['Predicted'],df_sgd['Actual'])))
plotDataframe(df_sgd,"SGD Regressor")
```

Listing 8.2: Code snippet for SGD regressor

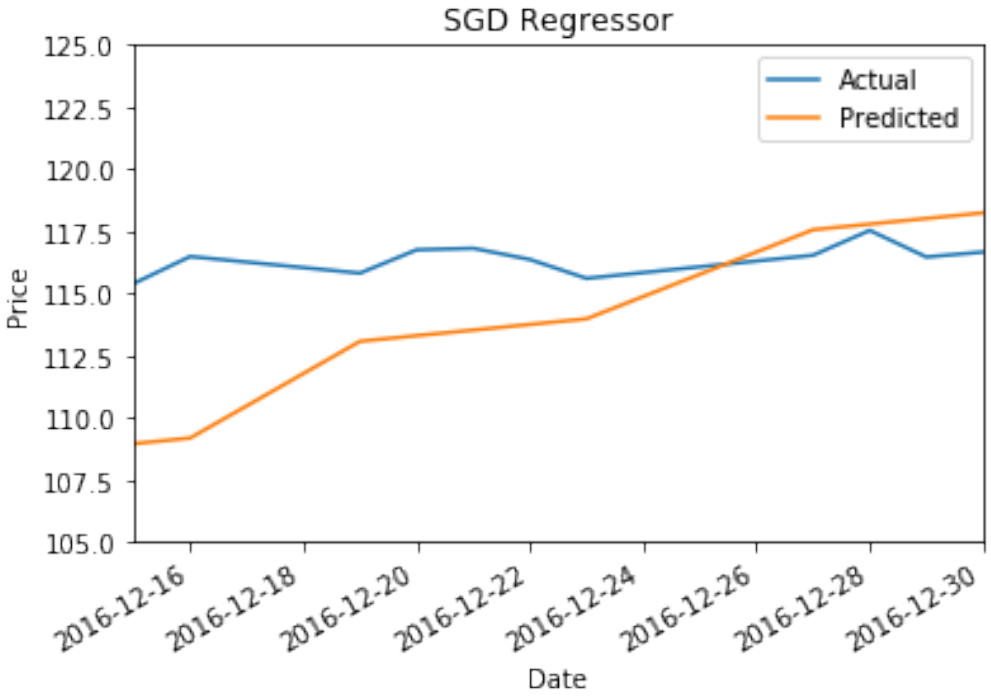

Figure 8.6: SGD REGRESSION GRAPH

Error: Margin of error ranges from **0.20242572** to **-6.246374034**.

8.2.3 DECISION TREE

It is a supervised learning method that can be used for both classification as well as regression problems. In this, each node represents a test on an attribute, each branch represents the output of the test and each leaf node represents class label. The model created by it predict the value of target variable by learning some rules inferred from data model.

```
from sklearn.tree import DecisionTreeRegressor
model_dt = DecisionTreeRegressor(max_depth=2)
model_dt.fit(X_train,y_train)
df_dt = pd.DataFrame(index=df_test.index)
df_dt['Actual'] = y_test
df_dt['Predicted'] = model_dt.predict(X_test)
print("% Change : "+str(calculateError(df_dt['Predicted'],df_dt['Actual'])))
plotDataframe(df_dt,"Decision Tree Regressor")
```

Listing 8.3: Code snippet for decision tree regressor

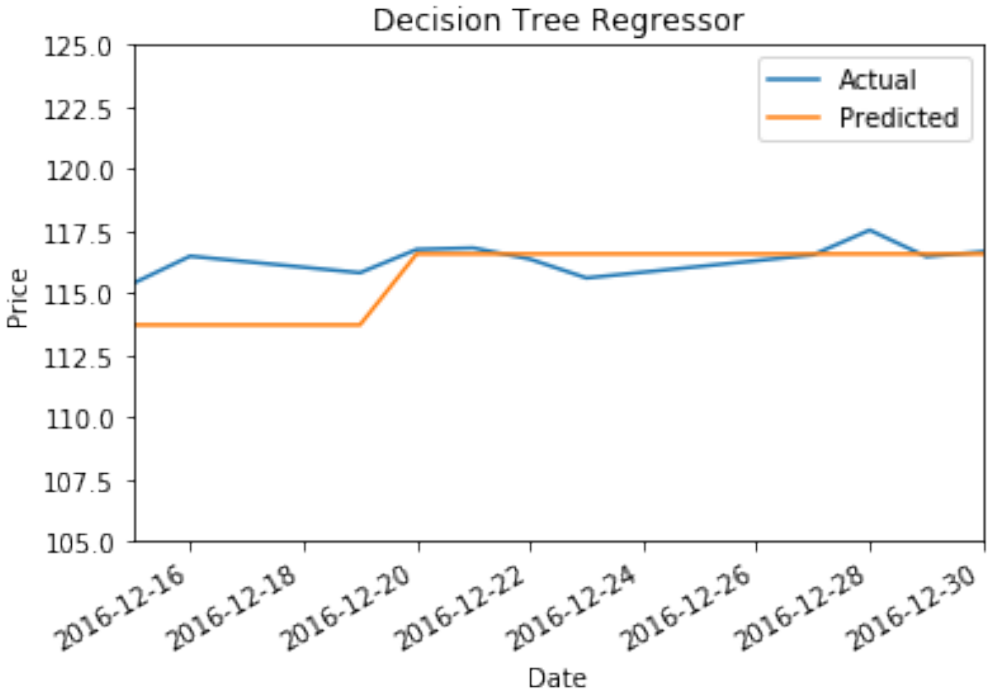

Figure 8.7: DECISION TREE GRAPH

Error: Margin of error ranges from **0.035475198** to **-2.378296966**.

8.2.4 RANDOM FOREST

Random Forest is the collection of several decision trees and all the trees are different in their structure. Here the training data is not randomize instead the algorithm is randomized. Random Forest algorithm has set a benchmark in ensemble learning method.

```
from sklearn.ensemble import RandomForestRegressor
model_rf = RandomForestRegressor(n_estimators=100,random_state=45)
model_rf.fit(X_train,y_train)
df_rf = pd.DataFrame(index=df_test.index)
df_rf['Actual'] = y_test
df_rf['Predicted'] = model_rf.predict(X_test)
print("% Change : "+str(calculateError(df_rf['Predicted'],df_rf['Actual'])))
plotDataframe(df_rf,"Random Forest Regressor")
```

Listing 8.4: Code snippet for random forest regressor

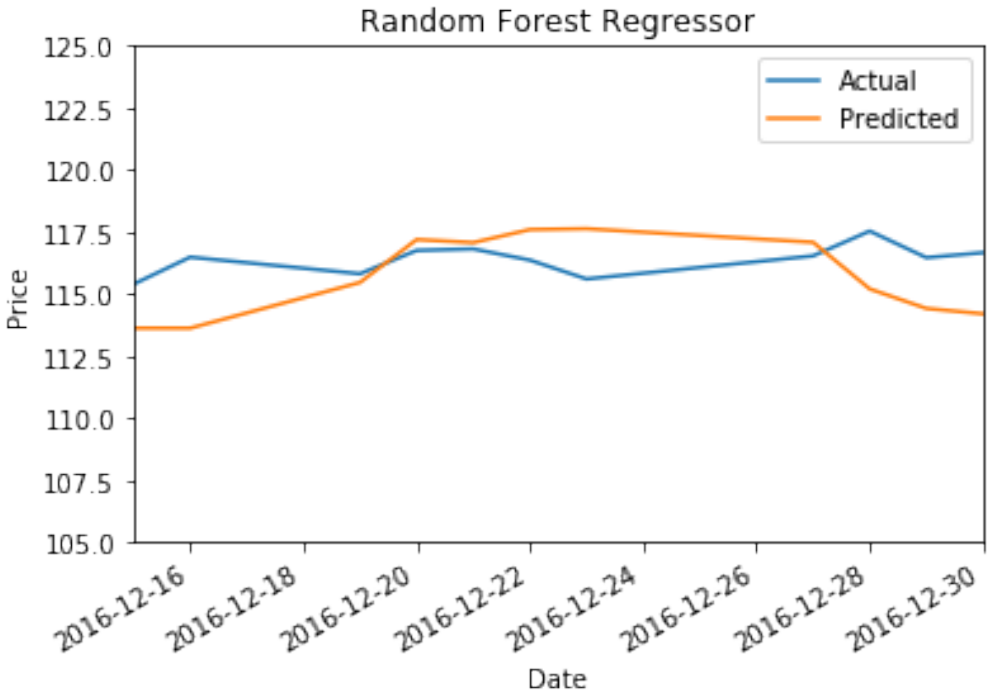

Figure 8.8: RANDOM FOREST GRAPH

Error: Margin of error ranges from **0.219432015** to **-2.457544583**.

8.2.5 KNN REGRESSOR

KNN is a machine learning algorithm which utilizes the simplicity of Euclidean Distance for calculating the distance of K nearest neighbor. The value of K is chosen by the programmer and depends on his/her intuition. In KNN Regressor the predicted value is the average of K nearest neighbor values.

```
model = KNeighborsRegressor(n_neighbors=10)
model.fit(X_train, y_train)
df_knn = pd.DataFrame(index=df_test.index)
df_knn['Actual'] = y_test
df_knn['Predicted'] = model.predict(X_test)
print("% Change : "+str(calculateError(df_knn['Predicted'],df_knn['Actual'])))
plotDataframe(df_knn,"KNN Regresson")
```

Listing 8.5: Code snippet for KNN regressor

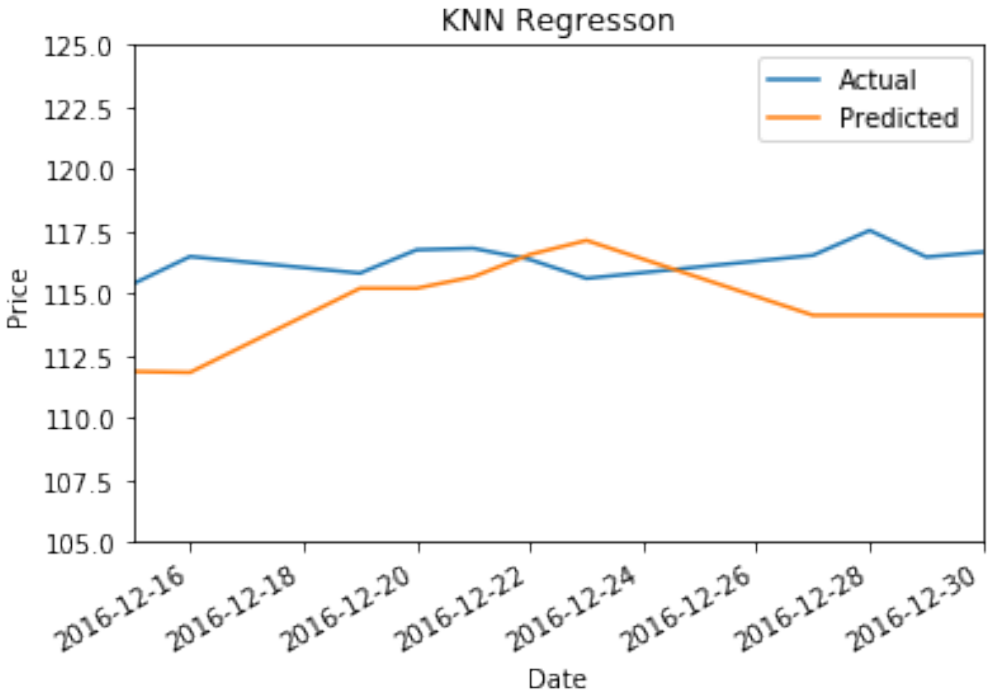

Figure 8.9: KNN REGRESSION GRAPH

Error: Margin of error ranges from **0.172755224** to **-4.002748742**.

8.2.6 Deep Neural Networks

Neural Networks are very effective to identify hidden features which other algorithms miss out. In this project we implemented Deep Neural Network which contains multiple hidden layers with Input and Output layers and activation function as Rectified Linear Unit(ReLU). The Deep Learning model performed extremely well and higher accuracy was achieved in comparison to other Machine Learning models. The reason for higher accuracy for Deep Learning is the ability of Deep Neural Network to capture the relations between the price and the features affecting it. DNN captures the relation between different seasons in a year and how the price of stock gets affected during a particular season or month.
The Actual Data in the table is from the test data set corresponding to a particular date. The Predicted data is the value predicted by Deep Learning algorithm on test data. Percentage Change represents the percentage of absolute difference between predicted data and actual data.

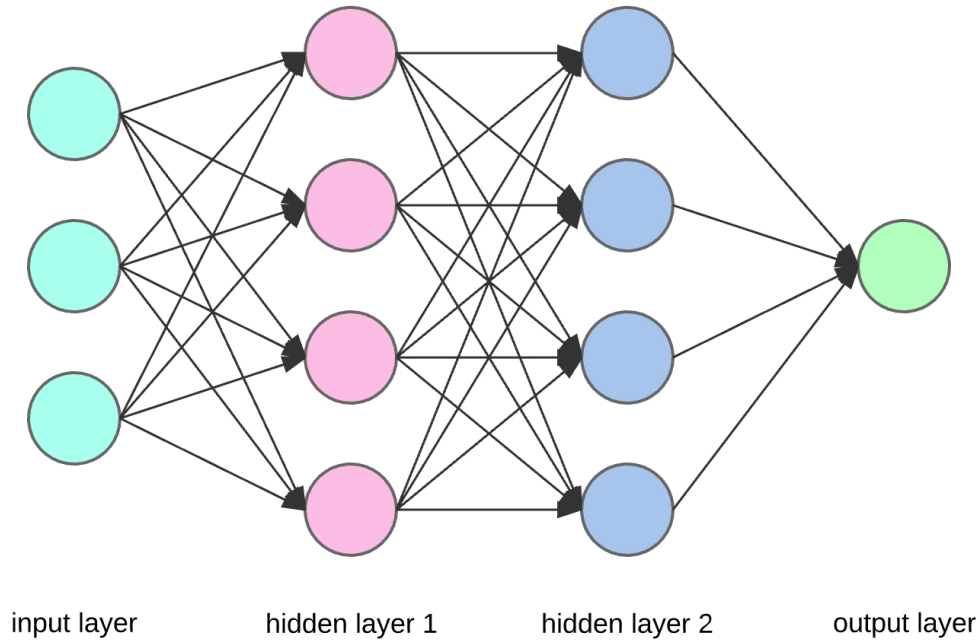

Figure 8.10: DEEP NEURAL NETWORKS STRUCTURE

```
from keras.models import Sequential
from keras.layers import Dense
model1 = Sequential()
model1.add(Dense(100,activation='relu',input_dim=X_train.shape[1]))
model1.add(Dense(100,activation='relu'))
model1.add(Dense(100,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(50,activation='relu'))
model1.add(Dense(100,activation='relu'))
model1.add(Dense(100,activation='relu'))
model1.add(Dense(1))
model1.compile(optimizer='adam',loss='mse')
model1.fit(X_train, y_train, epochs=500, batch_size=25)
```

Listing 8.6: Code snippet for deep neural networks

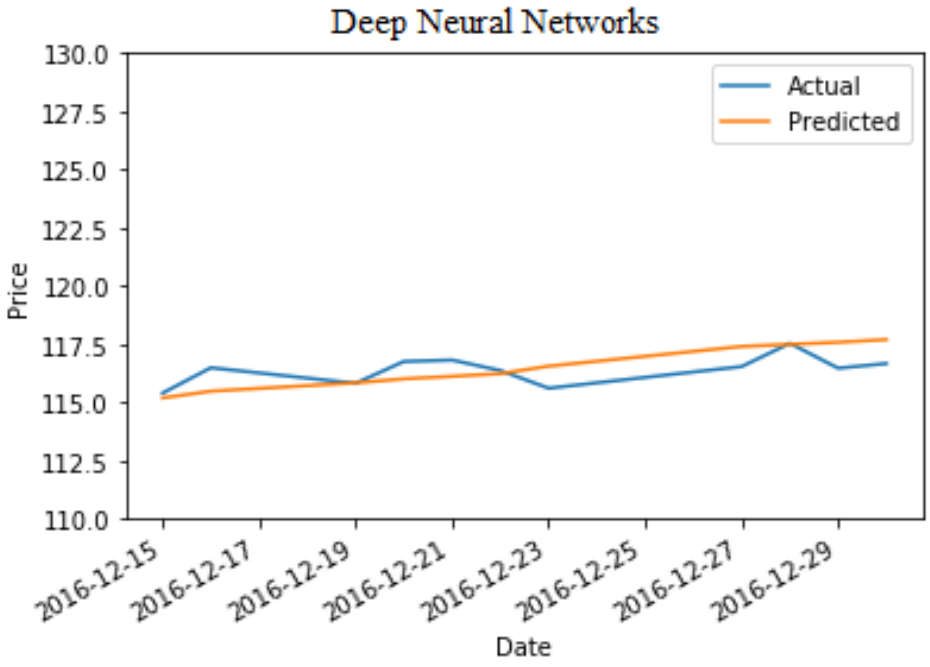

Figure 8.11: DEEP NEURAL NETWORKS GRAPH

Error: Margin of error ranges from **0.091120758** to **0.780186894**.

Chapter 9

Screenshots of Project

9.1 LINEAR REGRESSION WORK FLOW

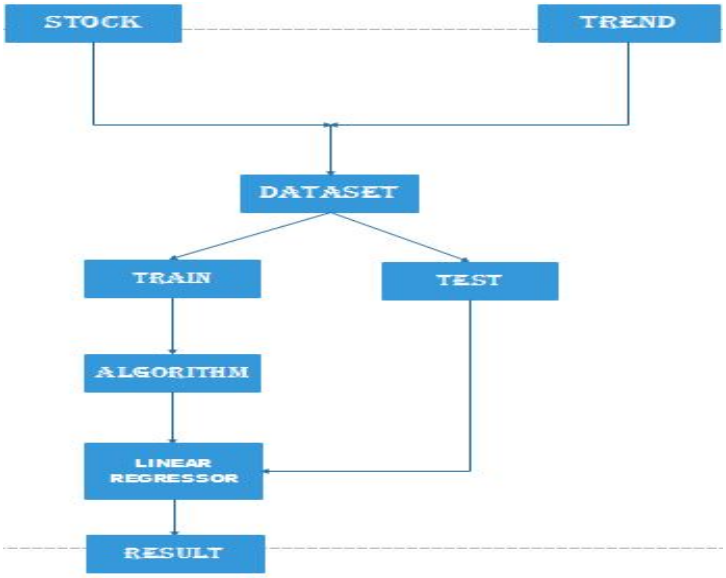

Figure 9.1: LINEAR REGRESSION WORK FLOW

9.2 SGD WORK FLOW

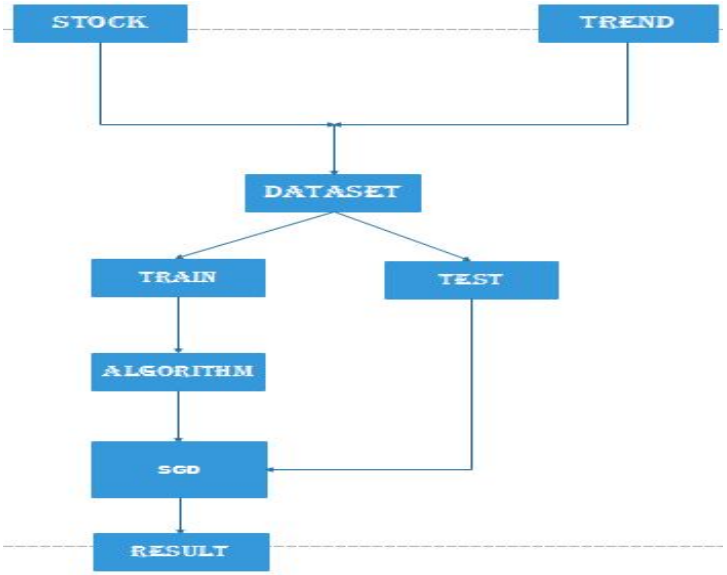

Figure 9.2: SGD WORK FLOW

9.3 DECISION TREE WORK FLOW

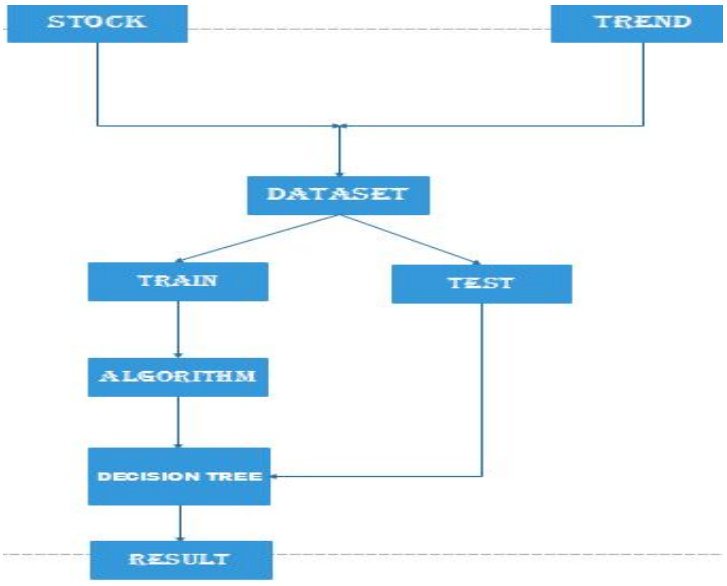

Figure 9.3: DECISION TREE WORK FLOW

9.4 RANDOM FOREST WORK FLOW

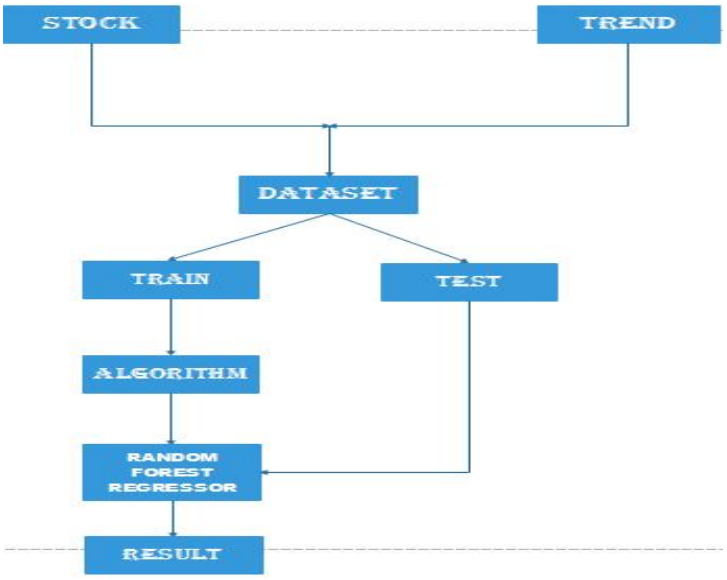

Figure 9.4: RANDOM FOREST WORK FLOW

9.5 KNN REGRESSOR WORK FLOW

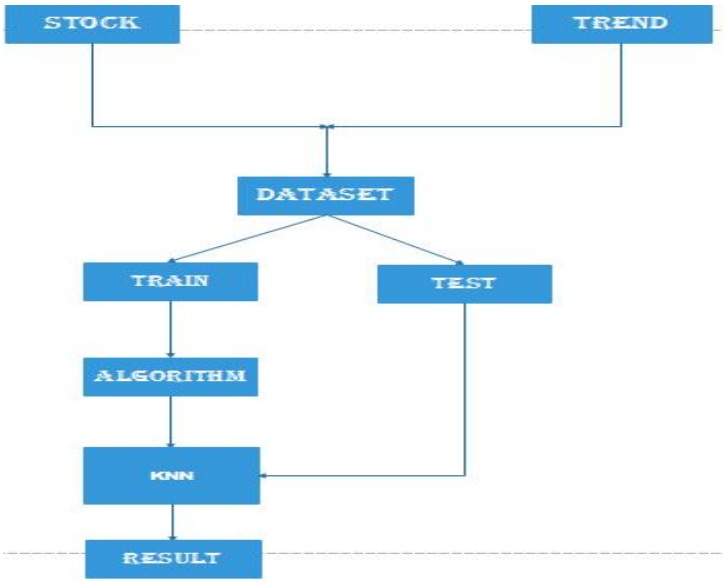

Figure 9.5: KNN REGRESSOR WORK FLOW

9.6 DEEP NEURAL NETWORKS

9.6.1 DEEP NEURAL NETWORKS WORK FLOW

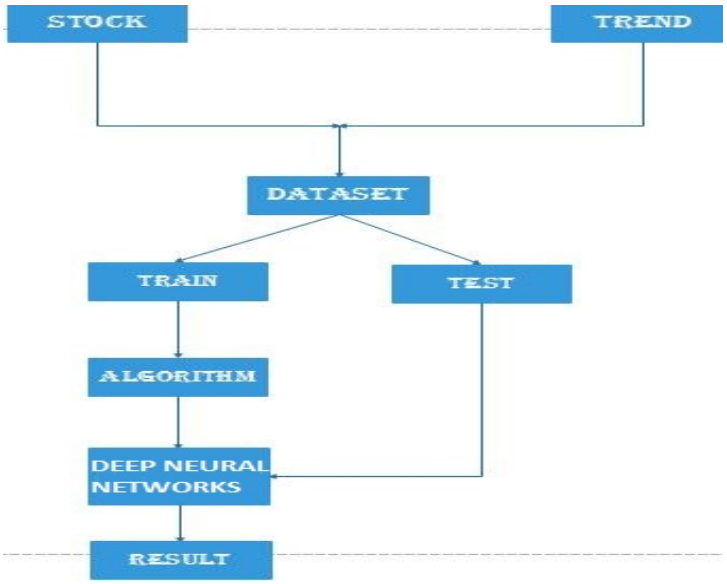

Figure 9.6: DEEP NEURAL NETWORKS WORK FLOW

9.6.2 INTO THE DNN STRUCTURE

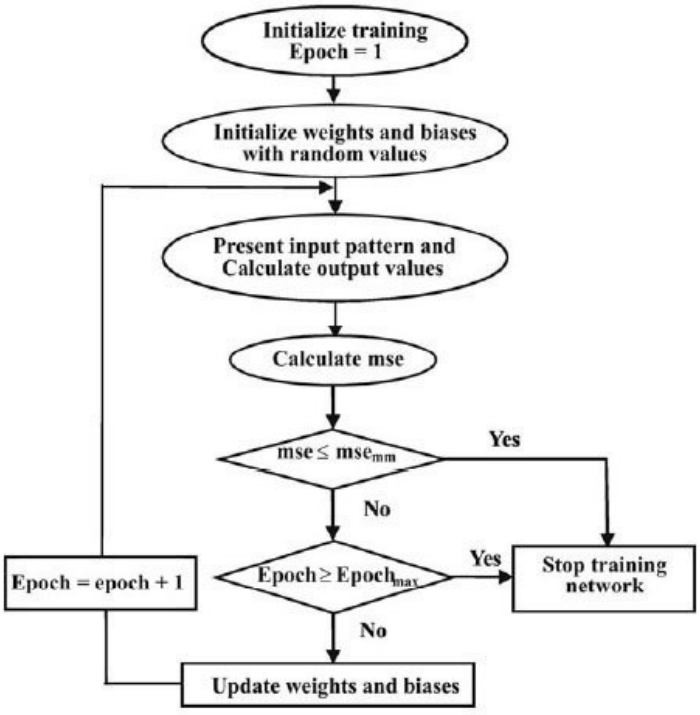

Figure 9.7: INTO THE DNN STRUCTURE

Chapter 10

Comparative Analysis

We have implemented 5 Machine Learning Regression algorithms namely, Linear Regression, SGD, Decision Tree, Random Forest, KNN along with Deep Neural Networks. We have pre-processed the data and used it as input to our algorithms. The results or outputs have been obtained and we have conducted a detailed study of the results to identify the best algorithm for stock market prediction.

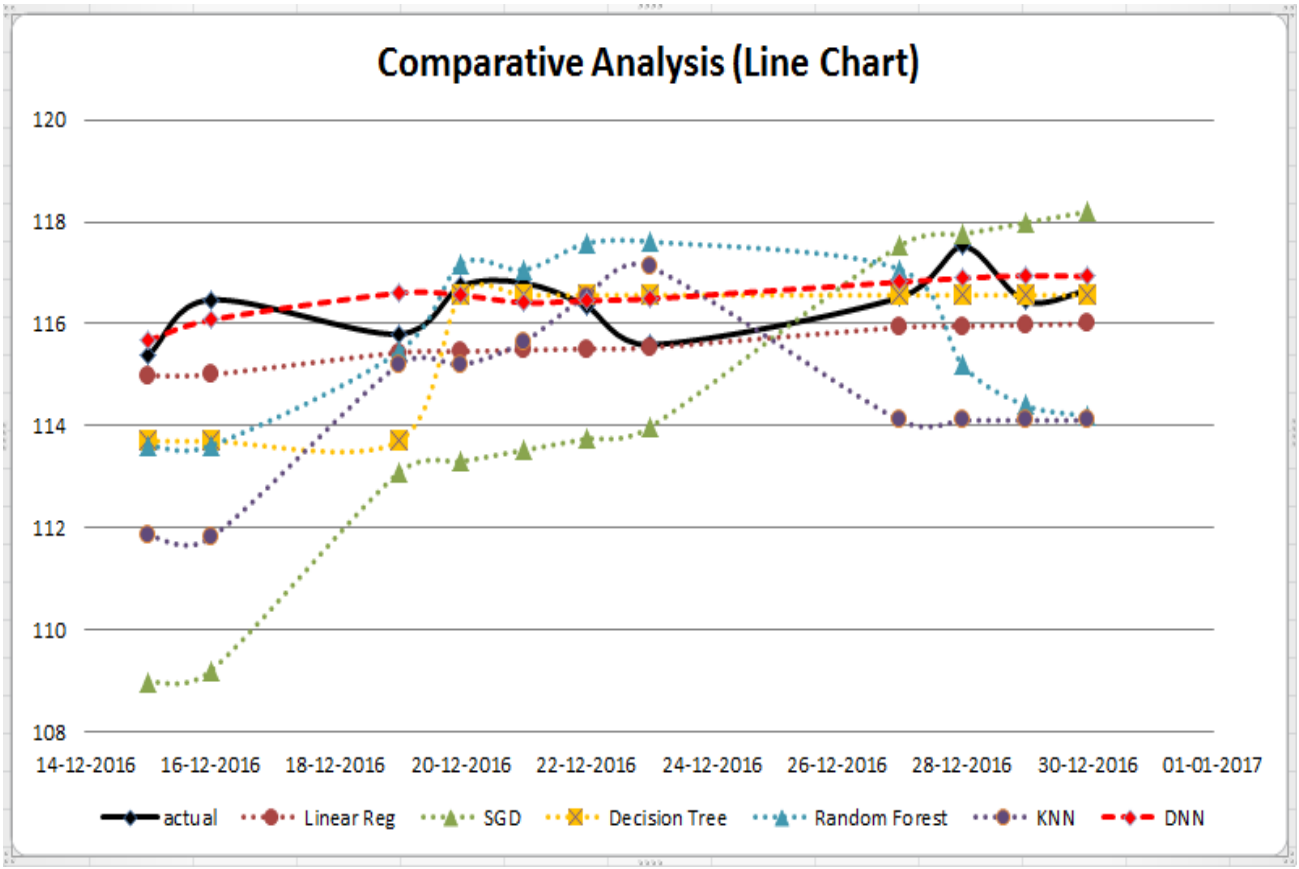

Figure 10.1: COMPARATIVE ANALYSIS (Line Graph)

As we can see in the line chart, the black line represents the actual values while the predicted values are indicated by other colors and designs. We have highlighted the predicted values obtained from Deep Neural Networks as it is the best result obtained among the implemented algorithms.

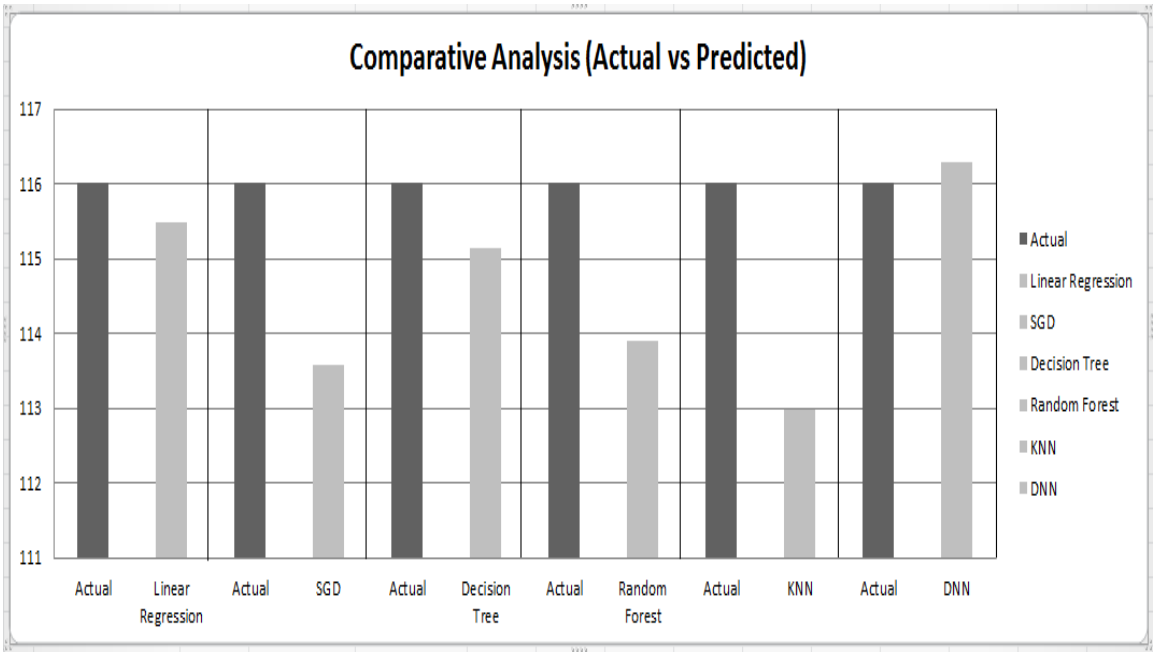

Figure 10.2: ACTUAL VS PREDICTED VALUES

The bar chart helps us to identify what the actual values are and how close the average predicted value of each of the algorithm is to the actual values. The average actual value across the selected dates is 116.0149995. The average predicted value from Linear Regression is obtained as 115.4923986. The results obtained from Stochastic Gradient Descent is slightly poor as the average obtained predicted value is found to be 113.5873033. Decision Tree Regression is quite close to the actual value. The average obtained predicted value is 115.1306656. In Random Forest Regression, the average obtained predicted value is 113.8976502. This is better than Stochastic Gradient Descent. 112.975499 is the average obtained predicted value in K-Nearest Neighbours Algorithm.

Deep Neural Networks have surpassed all the machine learning algorithms we have implemented. It has an average obtained predicted value of 116.3003655. Error percentage is the lowest at 0.002459734.
We can thus conclude from the results that Deep Neural Networks is best suited for stock market prediction.

Chapter 11

Conclusion and Future Scope

11.1 CONCLUSION

Working with Machine Learning on stock market analysis and prediction, we have come to the conclusion that there is a huge scope of work in this eld .As this eld is comparatively new, there are endless possibilities. The algorithms we have so far implemented on our dataset are:

- Linear Regressor
- SGD Regressor
- Decision Tree Regressor
- Random Forest Regressor
- KNN Regressor
- Deep Neural Networks

The main aim of this project is to understand and analyse the future state of the market. Having knowledge of the market beforehand will help people avoid losses. Our machine learning model can pave the way for a better future. Among the machine learning algorithms, Linear Regressor provided the best result. This result was surpassed while implementing the Deep Neural Networks. The error is calculated in each of the algorithm using the formula:

$$ErrorPercentage : ((PredictedValue - ActualValue)/ActualValue) * 100$$

The average error among all the implemented machine learning models is -1.07083 while the error in the deep neural networks is 0.114519. We can conclude from the obtained results that deep neural networks algorithm is most suitable for stock price prediction and can be widely used in the near future.

11.2 FUTURE SCOPE

Proposed methodology in real time :In this project, we have used both train and test data that were previously known to us. In future, we aim to test our model on real time data. For a start, we can experiment on small scale markets. This will help us to understand how our model is performing in real time.

Sentiment Analysis Of News Headlines: Statements by Influencers affects the market and thus stock price may change. It has been observed whenever there is a negative news about the organisation the price of stock falls and vice versa. Machine Learning has changed the way we analyse sentiment from text. Using Machine Learning it can be said whether the news is negative or positive and thus can be used to further enhance the prediction of price in advance.

References

[1] Rohit Choudhry., Kumkum Garg. A Hybrid Machine Learning System for Stock Market Forecasting.

[2] George S. Atsalakis., Kimon P. Valavanis. Forecasting stock market short-term trends using a neuro-fuzzy based methodology.

[3] Yakup Kara., Melek Acar Boyacioglu., mer Kaan Baykan. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange.

[4] Min-Ling Zhang., Zhi-Hua Zhou. ML-KNN:A lazy learning approach to multi-label learning

[5] Erkam Guresen ., Gulgun Kayakutlu., Tugrul U. Daim. Using artificial neural network models in stock market index prediction.

[6] Zan Huang., Hsinchun Chen., Chia-Jung Hsu., Wun-Hwa Chen., Soushan Wu. Credit rating analysis with support vector machines and neural networks: a market comparative study.

[7] Wei Huang., Yoshiteru Nakamori., Shou-Yang Wang. Forecasting stock market movement direction with support vector machine.

[8] Steven J. Jordan., Andrew J. Vivian., Mark E. Wohar. Forecasting returns: New European evidence.

[9] Birgul Egeli. Stock Market Prediction Using Artificial Neural Networks.

[10] Ding., Yue Zhang., Ting Liu., Junwen Duan. Deep Learning for Event-Driven Stock Prediction.

[11] Md. Rafiul Hassan., Baikunth Nath., Michael Kirley. A fusion model of HMM, ANN and GA for stock market forecasting.

[12] Kyoung-jae Kim., Ingoo Han. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index.

[13] Kai Chen., Yi Zhou., Fangyan Dai. A LSTM-based method for stock returns prediction : A case study of China stock market.

[14] Walid Chkili., Shawkat Hammoudeh., Duc Khuong Nguyen., (2014). Volatility forecasting and risk management for commodity markets in the presence of asymmetry and long memory.

[15] Takashi Kimoto and Kazuo Asakawala., Morio Yoda and Masakazu Takeoka. Stock Market Prediction System with Modular Neural Networks.

[16] Ryo Akita., Akira Yoshihara., Takashi Matsubara., Kuniaki Uehara. Deep Learning for Stock Prediction using Numerical and Textual Information.

[17] Richard Thaler., Werner F. M. De Bondt. Does the Stock Market Overreact?

[18] Akira Yoshihara., Kazuki Fujikawa., Kazuhiro Seki., Kuniaki Uehara. Predicting Stock Market Trends by Recurrent Deep Neural Networks.

[19] George S. Atsalakis., Kimon P. Valavanis. Surveying stock market forecasting techniques Part II: Soft computing methods.

[20] B.Wuthrich., V. Cho., S. Leung., D. Permunetilleke., K. Sankaran., J. Zhang., W. Lam. Daily Stock Market Forecast from Textual Web Data.

[21] `https://stackoverflow.com`

[22] `https://in.udemy.com`

[23] `https://in.udacity.com`