STOCK PERFORMANCE PREDICTION USING MACHINE LEARNING

A report submitted to

RAMAIAH INSTITUTE OF TECHNOLOGY

Bengaluru

IS821 SENIOR PROJECT

as partial fulfillment of the requirement for

Bachelor of Engineering (B.E) in Information Science and Engineering

by

SUNNY WADHWANI (USN- 1MS14IS110)
SIDDARTH SINGHAL (USN- 1MS14IS108)
AGASTHYA H.D (USN- 1MS14IS142)

Under the guidance of

Dr Vijaya Kumar B P

(Professor and H.O.D, Department of ISE)

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY

May 2018

Department of Information Science and Engineering

Ramaiah Institute of Technology

Bengaluru – 54

RAMAIAH
Institute of Technology

CERTIFICATE

This is to certify that Sunny Wadhwani (USN- 1MS14IS110) , Siddarth Singhal (USN- 1MS14IS104) AND Agasthya H.D (USN- 1MS14IS142) (who were working for their IS821 SENIOR PROJECT under my guidance, have completed the work as per my satisfaction with the topic STOCK PERFORMANCE PREDICTION USING MACHINE LEARNING. To the best of my understanding the work to be submitted in dissertation does not contain any work, which has been previously carried out by others and submitted by the candidates for themselves for the award of any degree anywhere.

(Guide)
Dr. Vijaya Kumar B P
Professor & Head, Dept. of ISE

(Head of the Department)
Dr. Vijaya Kumar B P
Professor & Head, Dept. of ISE

(Examiner 1)

Name

Signature

(Examiner 2)

Department of Information Science and Engineering

Ramaiah Institute of Technology

Bengaluru - 54

DECLARATION

We hereby declare that the entire work embodied in this IS821 SENIOR PROJECT report has been carried out by us at Ramaiah Institute of Technology under the supervision of Dr. Vijaya Kumar B P(Professor & Head, Dept. of. ISE) This project report has not been submitted in part or full for the award of any diploma or degree of this or any other University.

SUNNY WADHWANI (USN- 1MS14IS110)

SIDDARTH SINGHAL (USN- 1MS14IS108)

AGASTHYA H.D (USN- 1MS14IS142)

Acknowledgements

We are very grateful to our guide Dr. Vijaya Kumar B.P, Professor and Head of Department, ISE Department for guiding us for the project through out each phase especially some aspects which requires more experienced opinion and deeper thought process. He has been our inspiration as we overcame all the obstacles in the completion of this project work. By his guidance we were pull out the project with good results.

We would like to thank our beloved principal Dr. N.V.R Naidu for his support and encouragement.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparations and completion of this study.

We would like to express sincere thanks to all the teaching and non-teaching faculty of ISE Department and my dear friends who helped in all the ways while preparing the Report.

Abstract

The share market is highly un-predictable and volatile in nature for stockbrokers to take decisions on investment. Many companies are using Machine learning and Artificial intelligence in order to predict trends in the stock market and invest wisely.

We have made an exploration of various Machine Learning approaches which can be taken to solve the problem. The project aims at prediction of whether to Buy/Sell stock market shares at given point in time in order to obtain maximum profit. We use various Machine learning algorithms to do supervised learning of historic stock prices based on which machine learning model is built.

Implementation are being carried out for the proposed prediction model through various machine learning algorithm using R language in R-studio framework to evaluate whether stock is high performing or low performing for different data sets of various companies stock.

This project uses various Machine Learning algorithms such Naive Bayesian, SVM Algorithm, Decision Tree in order to accurately predict the performance of the shares of the companies.

Table of Contents

Abstract 2

1. Introduction 2

1.1 Motivation 2

1.2 Scope 2

1.3 Objectives 3

1.4 Proposed Model 3

1.5 Organization of Report 3

2. Literature Review 6

3. System Analysis and Design 9

3.1. Requirement Analysis 9

3.1.1. Functional Requirements 9

3.1.2. Non-Functional Requirements 9

3.2. Minimum System Requirements 10

3.2.1. Software Requirements 10

3.3. Data Flow 10

3.4. System Design 12

3.5 Algorithms 13

3.5.1 Naïve Bayes classifier 13

3.5.2 Decision Tree Classifier 16

3.5.3 SVM classifier 17

4. Modeling and Implementation 24

4.1. Input Data 24

4.1.1 Price Difference 24

4.1.2 High Price 24

4.1.3 Low Price 24
4.1.4 Open Price 24
4.1.5 Closing Price 24
4.1.6 Volume 24
4.2 Modules used 25
4.2.1 Naïve Bayes Model 25
4.2.2 Decision Tree Model 32
4.2.3 Support Vector Machine Model 35
5. Testing and Results 39
5.1 Testing 39
5.1.1. Unit Testing 39
5.1.2. Performance testing 39
5.1 Result 40
6. Conclusion and Future Work 42
Bibliography 43

List of Figures

Fig. 3.1: Data Flow Diagram (Level 0)…… 11

Fig. 3.2: System Flow Diagram…… 12

Fig. 3.3: Classification of Stock Using Naive Bayes Flow Diagram…… 14

Fig. 3.4: Svm Flow Chart…….. 22

Fig.4.1: Naive Bayes-Accuracy VS Iterations Graph……... 30

Fig. 4.2: Naive Bayes-Time VS Iteration Graph ……. 31

Fig. 4.3: Decision Tree Classifier-1…….. 32

Fig. 4.4: Decision Tree Classifier-2…….. 33

Fig. 4.5: Decision Tree Classifier When Only Volume is considered…….. 34

Fig. 4.7: SVM-Accuracy VS Iterations Graph……. 35

Fig. 4.8: SVM-Time VS Iterations Graph…… 36

List of Tables

4.1: Training Data for Naive Bayes Example……. 25

4.2: Test Data for Naive Bayes Example…… 27

4.3: Confusion Matrix……. 28

4.4: Naive Bayes-Confusion Matrix…… 31

4.5: Decision Tree-Confusion Matrix……. 33

4.6: Decision Tree-Confusion Matrix When Volume is considered…… 34

4.7: SVM-Confusion Matrix…… 36

CHAPTER 1

Introduction

1. Introduction

Stock Market prediction and analysis is the act of trying to determine the future value of a company stock or other financial instrument traded on an exchange. Stock market is the important part of economy of the country and plays a vital role in the growth of the industry and commerce of the country that eventually affects the economy of the country. Both investors and industry are involved in stock market and wants to know whether some stock will rise or fall over certain period of time. The stock market is the primary source for any company to raise funds for business expansions. It is based on the concept of demand and supply. If the demand for a company's stock is higher, then the company share price increases and if the demand for company's stock is low then the company share price decrease.

1.1 Motivation

Share market is a very volatile space where prices change within a short span of time and share prices depends on various factors which cannot be predicted easily by stock brokers. Many investment banking companies are using Algorithmic trading to predict trends in stock prices for short-term as well as long term investments. We therefore considered this topic as we felt it very interesting to work on this domain given the complexity of the problem.

1.2 Scope

The project has a significant usage in predicting the trend of the future market. Even a layman can take an appropriate decision regarding investment.

Various companies, having their stocks in different share holdings, can influence their marketing strategies based on the knowledge of current and future behaviour of the stock market.

When project is applied on a web platform, stock brokers can use it as a means for various suggestions in the stock market.

1.3 Objectives

The main objective of the project includes the following:

- Consider various strategies for calculation of performance of stock prices.
- Use different Machine Learning models and algorithms to determine the accuracy of the each model and algorithm.
- Use the Machine learning models to determine the performance of stock prices in real time basis.

1.4 Proposed Model

The proposed model includes different machine learning algorithms that determine the accuracy of each algorithm against standard data.

Various Machine learning algorithm used are as follows:

- Naïve Bayes Classifier
- SVM Classifier
- Decision Tree Classifier

The model uses the data collected of various companies from the stock market to train and test the classifier for prediction.

1.5 Organization of Report

The Project report discusses about the algorithmic trading and various machine learning algorithms applied on stock market data and predicting the class of the stock.

The report has been broadly divided into six chapters as follows:
Chapter 1 deals with the introduction to the entire project.

Chapter 2 concentrates on the literature Survey done for the entire project which is presented to highlight the work that has been done until now.

Chapter 3 concentrates on system design and analysis, in this the system requirements are discussed, minimum system requirements , data flow diagram is discussed that how data is flown through input data and produces output data and system design is discussed how system is designed.

Chapter 4 discusses the various Modules used in the project and their importance and how effective the each modules are. The various modules discussed are Naïve Bayes classifier, Decision tree classifier, Svm classifier.

Chapter 5 consists of results and testing and shows the accuracy of all the models.

Chapter 6 discusses the conclusion and future work of the project.

Chapter 2

Literature Review

2. Literature Review

In the paper [1], researchers examine two separate algorithms and methodologies utilized to investigate Stock Market trends that is usage of linear regression and SVM and compare the accuracy of both.
For prediction concept of buckets is introduced where time domain was separated into
1-minute buckets and attempted to extract 8 identifiers to describe the price and volume change of that minute heuristically. Identifiers such as low, high, volume ,open, close are used to capture the trend of the data of a given minute by formulating the algorithms to predict the change in the closing price of each 1 minute bucket given information of the remaining seven identifiers (volume and price) prior to that minute.

In the paper [2], the project aims at predicting the short-term pricing trend of selected stocks and simulate the trading results with a simple strategy, to provide a key reference for improving algorithmic trading and better trading strategies. Using simple strategy of interday buying of share based on prediction of whether the stock price would increase or decrease in the next day based on past few days using the simplified model, researches managed to achieve a total return of 740.2% (for 731 days) corresponding to a daily return of 0.2294%, provided that the future stock prices are known.

In the paper [3], researchers have implemented short term model and long term prediction. In short term model, prediction of price is done for the next day. In long term prediction of stock prices, prediction of price is done over a period of time based on which purchase is done .Different set of features are used and ideal number of features is found out. Many machine learning models like Logistic Regression, Gaussian Discriminant Analysis, Quadratic Discriminant Analysis, and SVM are used where SVM performs best with 79% accuracy.

In the paper [4], authors studied prediction firm bankruptcy using neural networks and classical multiple discriminant analysis, where neural networks performed significantly better than multiple discriminant analysis. Min and Lee were doing prediction of bankruptcy using machine learning. They evaluated methods based on Support Vector Machine, multiple discriminant

analysis, logistic regression analysis, and three-layer fully connected back-propagation neural networks. Their results indicated that support vector machines outperformed other approaches

In the paper [5], author did a research where he tried to predict stock prices by using ensemble learning, composed of decision trees and artificial neural networks. He created dataset from Taiwanese stock market data, taking into account fundamental indexes, technical indexes, and macroeconomic indexes. The performance of Decision Tree + Artificial Neural Network trained on Taiwan stock exchange data showed F- score performance of 77%. Single algorithms showed F-score performance up to 67%

In the paper [6] ,the outcomes are great since it predicts a high level of the results for different stocks, and does not lose much precision when connected to an example from outside the preparation test. The model still has an incredible measure of opportunity to get better. The multivariate insights can be utilized to break down an organization's imaginable execution in the share trading system concerning worldwide monetary conditions and in addition its own money related execution the earlier year".

In the paper [7], the novel system called the SVM exhibit is proposed and associated in securities trade list gauging. For a given data, the money re lated data is deteriorated to a couple of trademark mode limits. By then to different areas, extraordinary SVMs are used which have various piece limits and learning parameters to get gauges. At last through mix differing region conjectures the cash relate d data envisioning is acquired through the financial data of China Stock Price.

Chapter 3

System Analysis and Design

3. System Analysis and Design

3.1. Requirement Analysis

After the extensive analysis in the system we are now familiar with the requirement that current system needs. Based on that we have categorized that in two categories that are function and non-functional requirements. These requirements are listed below:

3.1.1. Functional Requirements

Functional requirements are the requirements that needed to be added in the system to satisfy the needs of the users. Based on that the functional requirements that system must have are:

- The system should be able to able to classify the class of the stock correctly.
- The system should collect the right data or accurate data in order to have the right prediction of class in stock data in which it belong.
- The system should produce valid outputs
- The live data is collected by the alpha vantage's api from where you get data of past 10 years easily.

3.1.2. Non-Functional Requirements

Non-functional requirement is a description of features, characteristics and attribute of the system as well as any constraints that may limit the boundaries of the proposed system. The non-functional requirements are essentially based on the performance, information, control and security efficiency and services. Based on these the non-functional requirements are as follows:

- The system should provide better accuracy.
- The system should perform efficiently in short time
- The system should not take more time to give result or to produce output
- The system should consume less memory during the execution of the model
- The system should work smoothly.

3.2. Minimum System Requirements

3.2.1. Software Requirements

Operating system : Windows vista, 7, 8, 10/Ubuntu

Coding Language : R

Software Required : R-Studio

Ram Required : More than 32 MB

- A good Internet Connection is required so to crawl the data from the alpha vantage website.

3.3. Data Flow

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. Data Flow models are used to show how data flows through a sequence of processing steps. The data is transformed at each step before moving on to the next stage. These processing steps or transformations are program functions when Data Flow diagrams are used to document a software design. DFD diagram is composed of four elements, which are process, data flow, external entity and data store.

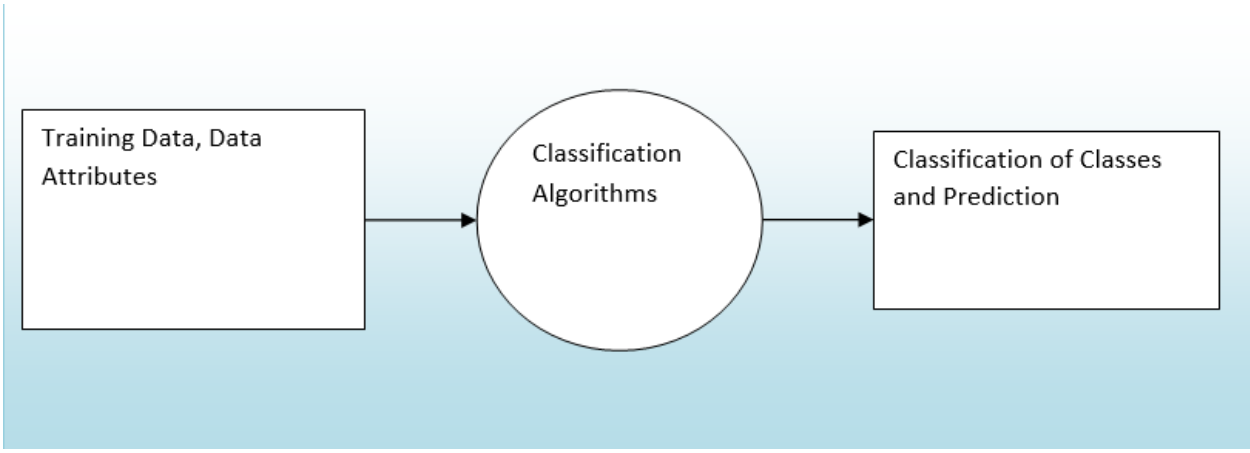

Fig. 3.1: DFD level 0

Fig.3.1 shows *DFD Level 0* shows the input namely training data and attributes. In next step the data is passed over to the classification algorithms and training of the model is done by the data available .After training is completed the prediction results are depicted and classification of class is done.

3.4. System Design

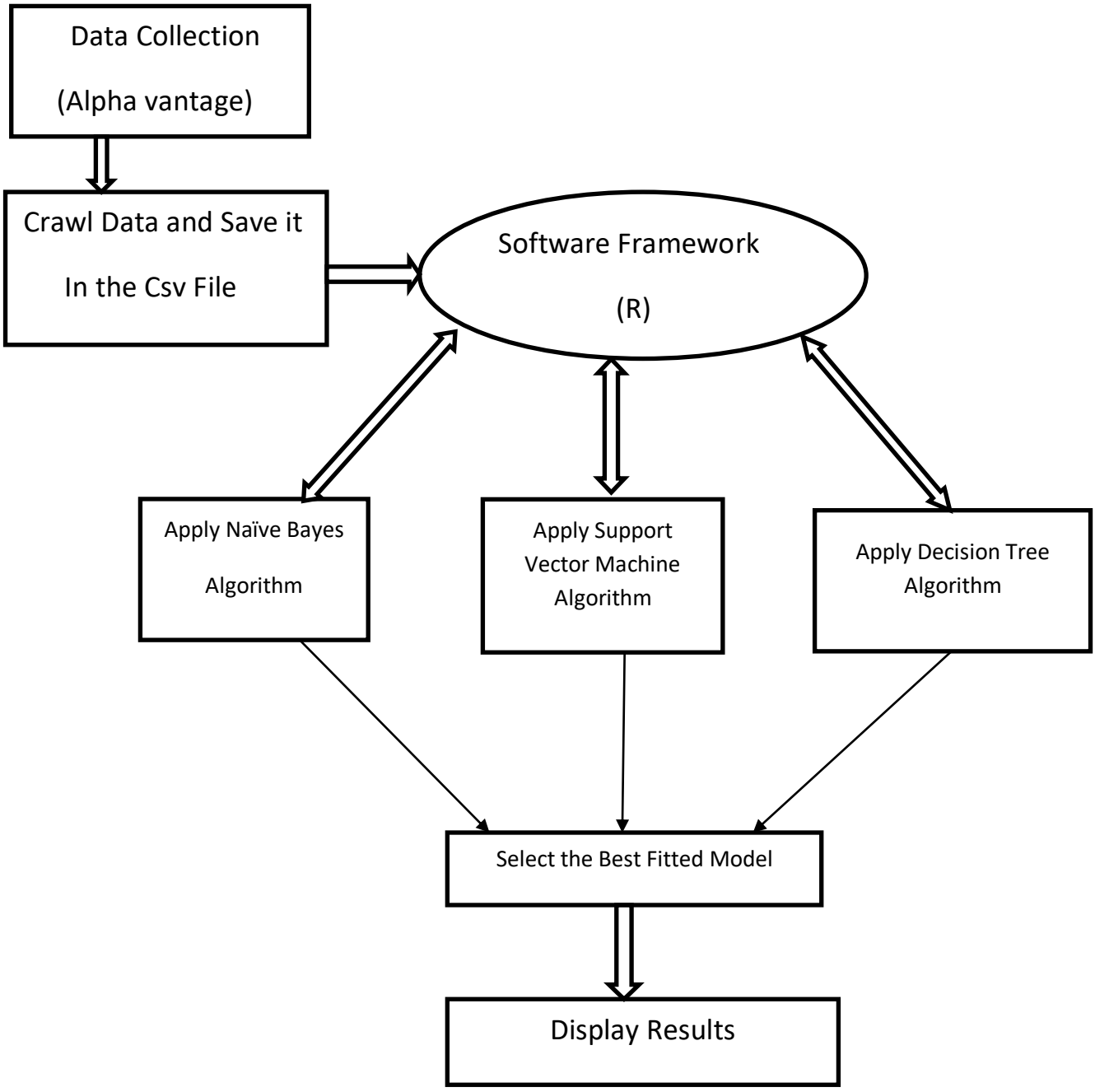

Fig. 3.2: System Flow Diagram

Fig. 3.2 shows *System flow diagram* shows the input namely training data and data attributes. The process contain the fo1llowing algorithms Naïve Bayes Classification, Support Vector Machine, Decision tree and finally Ranking of Categories Classes. The output is the classification class and ranking of categories.

3.5 Algorithms

We have used three different classifier algorithms in our project namely Naïve Bayes Classifier, Decision Tree Classifier, Svm Classifier

3.5.1 Naïve Bayes classifier

Naive Bayes is a kind of classifier which uses the Bayes Theorem. It predicts membership probabilities for each class such as the probability that given record or data point belongs to a particular class. The class with the highest probability is considered as the most likely class.

Naive Bayes classifier assumes that all the features are unrelated to each other. Presence or absence of a feature does not influence the presence or absence of any other feature or we can say Naïve Bayes classifier finds out the independent probabilities of each attribute belonging to the each of the class label.

The algorithm is described using the following steps and the steps will be applied for each of the companies. The algorithm has been described using two set classes namely Low performing and High performing stock and then attributes namely Volume, Low Price , High Price, Closing Price, Open Price .

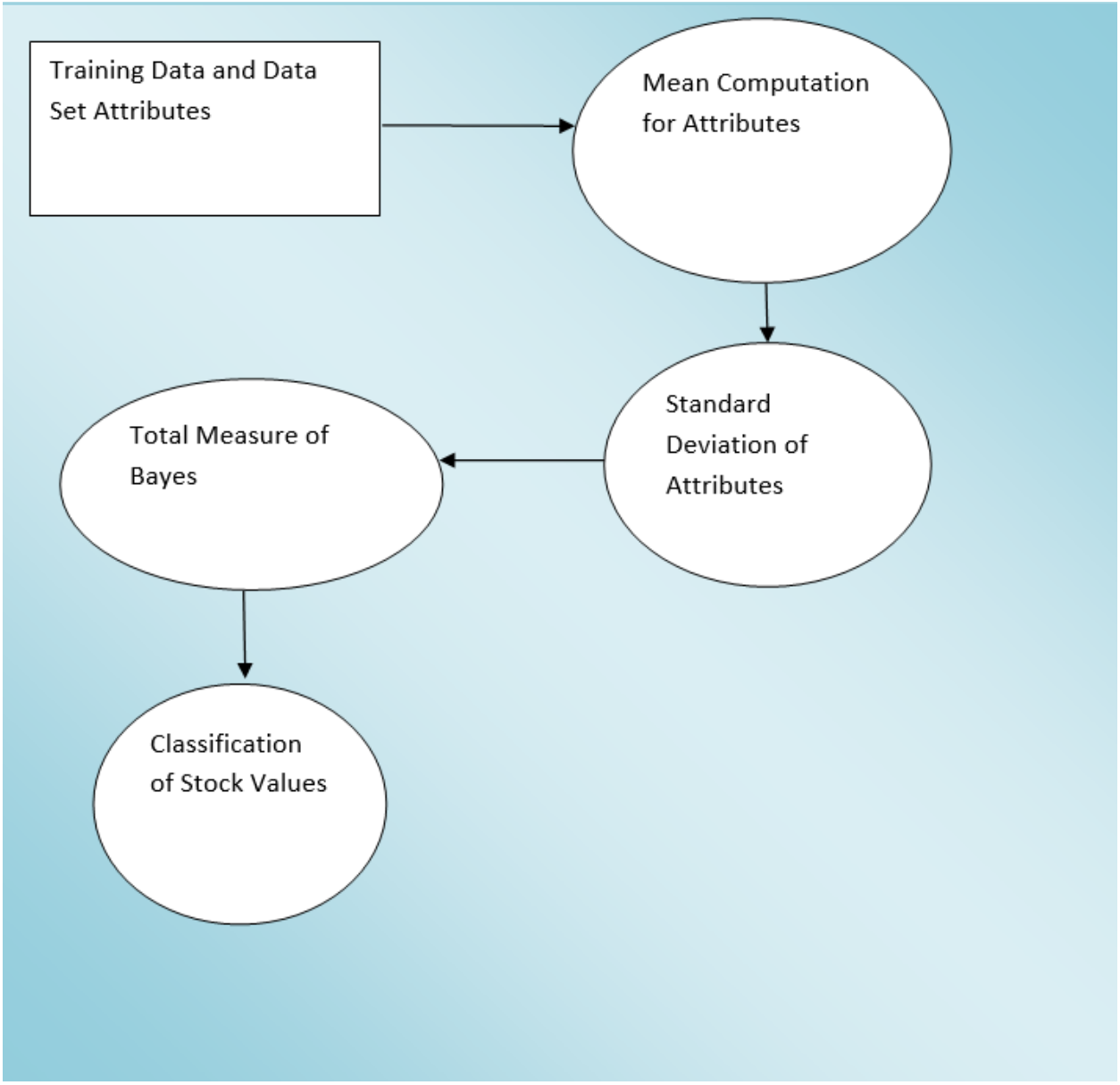

Fig. 3.3: Classification of stock data using Naïve Bayes Flow Diagram

Steps to perform Naïve Bayes classification of stock data:

1) Obtain the list of Volume, High Price, Low Price, Closing Price ,Open Price from the previous history data set for Low Performing

2) Compute the summation of list of list of Volume, High Price, Low Price, Closing Price ,Open Price for Low Performing
3) Compute the mean of list of Volume, High Price, Low Price, Closing Price ,Open Price taken all attributes separately for Low Performing
4) Compute the standard deviation of list of Volume, High Price, Low Price, Closing Price ,Open Price taken all attributes separately for Low Performing
5) Compute the individual Naive Bayes probability of Volume, High Price, Low Price, Closing Price ,Open Price take for Low Performing

In the step 5 the computation of probability is performed using the following equation

$$P_{attribute} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{(\mu-T)^2}{2*\sigma^2}\right)}$$

Where,

$\sigma = s\tan dard\ \ deviation$

$\mu = mean$

$T = current\ \ value\ of\ \ attribute$

6) Compute the total probability of Low Performing, it will be multiplication of each Low Performing probability of each attributes.

7) Obtain the list of list of Volume, High Price, Low Price, Closing Price, Open Price based on history data set for High Performing.
8) Compute the mean of Volume, High Price, Low Price, Closing Price, and Open Price, taken all attributes separately for High Performing.
9) Compute the standard deviation of list of Volume, High Price, Low Price, Closing Price, and Open Price taken all attributes separately for High Performing.

10) Compute the Naive Bayes probability Of Volume, High Price, Low Price, Closing Price, and Open Price for High Performing.

In step 10 the probability is calculated based on the given formulae

$$P_{attribute} = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{(\mu-T)^2}{2*\sigma^2}\right)}$$

Where,

$\sigma = s\tan dard\ \ deviation$

$\mu = mean$

$T = current\ \ value\ of\ \ attribute$

11) The total probability of High Performing Class is computed using multiplication of each independent probability of each attributes that belong to High Performing Class.
12) Now after computing all the probabilities the test data comes and we have to check whether it belongs to the Low Performing Class or High Performing Class.
13) The Test Data will contain all the 5 Attributes Namely Volume, Closing Price, Open Price, High Price and Low Price.
14) Compute The Probability of Test Data for Low Performing
15) Compute The Probability of Test Data for High Performing
16) The one with the greater probability will be the class for the Test Data

3.5.2 Decision Tree Classifier

Decision Tree Algorithm is a supervised classifier algorithm. It works like human where it makes decisions based on attributes of the data.

Decision Tree Algorithm uses the main deciding attribute at the root and split the data based on other attributes as well as main attribute. Initially all the data are assumed to be at root.

Recursively, the records are distributed by starting from the root.

Test data‘s test label is decided based on model built from training data.

Attributes selection can be done based on two computations namely information gain and Gini index.

Based on these values attributes are placed in tree hierarchy. Higher position at root is given for lower values of Gini index.

By using information gain as a criterion, we try to estimate the information contained by each attribute.
To measure the randomness or uncertainty of a random variable X is defined by Entropy**.**

In case of binary classification problem with only two classes.

- If all examples are positive or all are negative then entropy will be zero i.e, low.
- Suppose that 1/2 of the records are of +ve class and 1/2 are of -ve class then entropy is one i.e., high.

Information gain can be calculated

$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) \log p(x).$$

Where p(x) is the probability of the class in the resulting split.

Gini Index is measure of often a randomly chosen element would be incorrectly captured suppose they were randomly labelled. It means an attribute with lower Gini index is more preferable.

$$Gini\ Index = 1 - \sum_j p_j^2$$

Where pj is probability of data items classified in class j.

3.5.3 SVM classifier

SVM algorithm is based on statistical learning theory. SVM can be used for both classification and regression task. In classification we try to find an optimal hyperplane that separates two classes. In order to find an optimal hyperplane,

We need to minimize the normal of the vector w, which defines the separating hyperplane. This is equivalent to maximizing the margin between two classes.

Finding the biggest margin, is the same thing as finding the optimal hyperplane**.** Steps to find out the margin are as follow

1. You have a dataset
2. select two hyperplanes which separate the data with no points between them
3. maximize their distance (the margin)

The region bounded by the two hyperplanes will be the biggest possible margin.

So we will go through the algorithm step by step

Step 1: You have a dataset **D** and you want to classify it

Most of the time your data will be composed of n vectors xi.

Each xi will also be associated with a value yi indicating if the element belongs to the class (+1) or not (-1).

Note that yi can only have two possible values -1 or +1

So your dataset D is the set of n couples of element (xi,yi) The more formal definition of an initial dataset in set theory is :

$D=\{ (xi,yi) \mid xi \in R , yi \in \{-1,1\}\}$

Step 2: You need to select two hyperplanes separating the data with no points between them

Any hyperplane can be written as the set of points x satisfying $w \cdot x+b=0$.

We can select two others hyperplanes H1 and H2 which also separate the data and have the following equations:

$W \cdot x +b=1$

And

$w \cdot x+ b=-1$

Now we want to be sure that they have no points between them.

We won't select any hyperplane, we will only select those who meet the two following **constraints**:

For each vector xi either:

$w \cdot xi+b \geq 1$for xi having the class1

or

$w \cdot x_i + b \leq -1$ for x_i having the class -1

Combining both constraints

We start with equation

for x_i having the class -1

$w \cdot x_i + b \leq -1$

And multiply both sides by y_i (which is always -1 in this equation)

$Y_i (w \cdot x_i + b) \geq y_i(-1)$

Which means equation

$Y_i (w \cdot x_i + b) \geq 1$ for x_i having the class -1

as $y_i = 1$ it doesn't change the sign of the inequation.

$y_i(w \cdot x_i + b) \geq 1$ for x_i having the class1

We combine equations

$y_i(w \cdot x_i + b) \geq 1$ for all $1 \leq i \leq n$

We now have a unique constraint instead of two equations, but they are mathematically equivalent. So their effect is the same (there will be no points between the two hyperplanes).

Step 3: Maximize the distance between the two hyperplanes

Let:

- H_0 be the hyperplane having the equation $w \cdot x + b = -1$
- H_1 be the hyperplane having the equation $w \cdot x + b = 1$
- x_0 be a point in the hyperplane $H_0 H_0$.

We will call m the perpendicular distance from x_0 to the hyperplane H_1 . By definition, m is what we are used to call **the margin**.

As x_0 is in H_0, m is the distance between hyperplanes H_0 and H_1.

Let's define $u = \frac{w}{||w||}$ the unit vector of w .As it is a unit vector ‖u‖=1and it has the same direction as w so it is also perpendicular to the hyperplane. If we multiply u by m we get the vector k=mu and:

1. ‖k‖=m‖
2. k is perpendicular to H1 (because it has the same direction as u)

$k=mu=\frac{m*w}{||w||}$

the fact that z0 is in H1 means that

w·z0+b=1

We can replace z0 by x0+k because that is how we constructed it.

w·(x0+k)+b=1

We can now replace k using equation

w·(x0+m $\frac{w}{||w||}$) +b=1

We now expand equation

w·x0+ (mw·$\frac{w}{||w||}$)+b=1

The dot product of a vector with itself is the square of its norm so :

w·x0+m $\frac{||w||^2}{||w||}$ +b=1

w·x0+m‖w‖+b=1

w·x0+b=1−m‖w‖

As x0 is in H0 then w·x0+b=−1

−1=1−m‖w‖

m*‖w‖=2

$m=\frac{2}{||w||}$

This give us the following optimization problem:

Minimize in (w,b)

$\|w\|$

Subject to $y_i (w \cdot x_i + b) \geq 1$

For any $i=1 \ldots n$

Solving this problem is like solving and equation. Once we have solved it, we will have found the couple (w,b) for which $\|w\|$ is the smallest possible and the constraints we fixed are met. Which means we will have the equation of the optimal hyperplane.

The flow chart of Svm algorithm is given below:

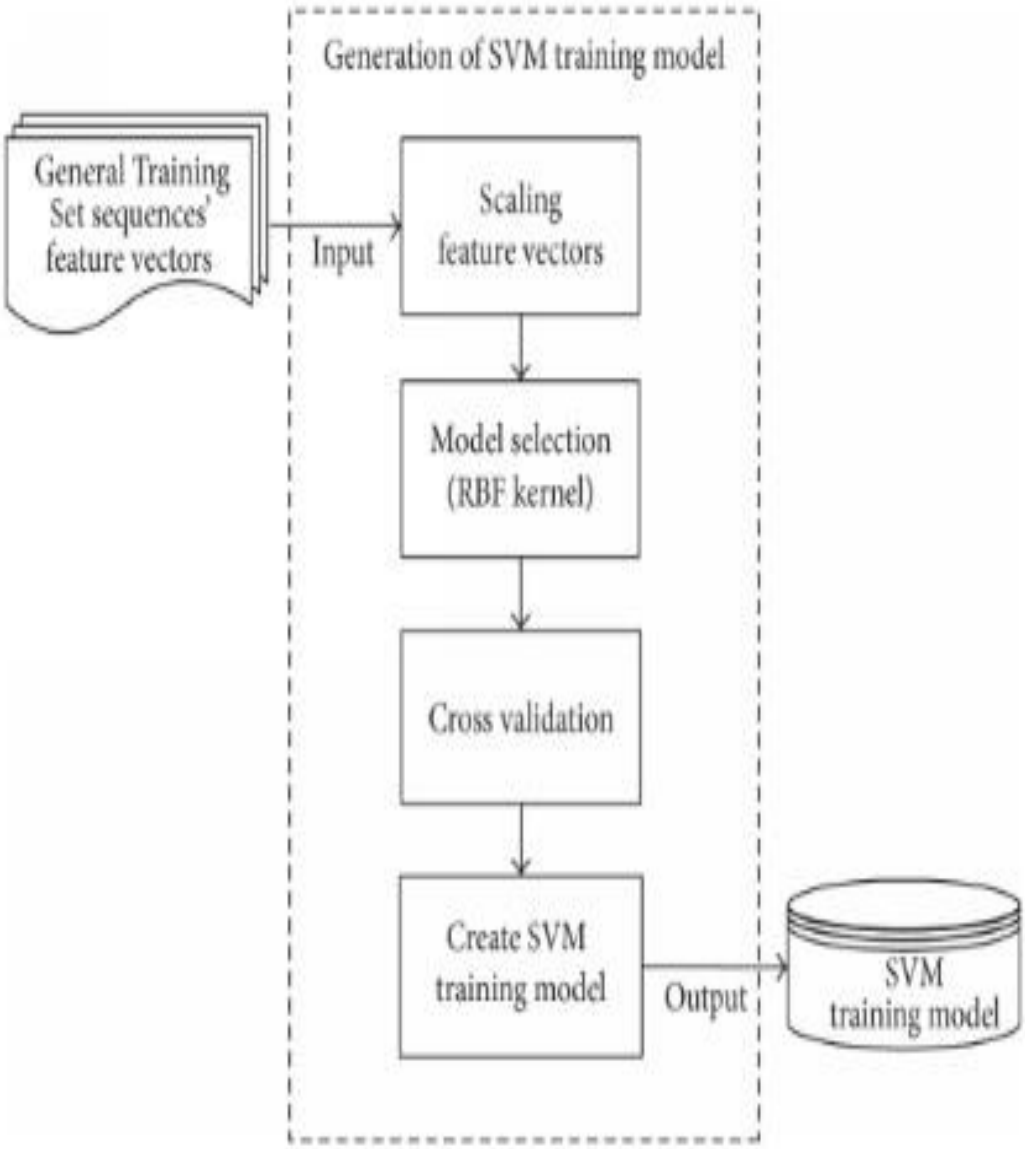

Fig. 3.4: SVM Flow chart

Chapter 4

Modeling and Implementation

4. Modeling and Implementation

4.1. Input Data

Here is the brief description to all the input data used for all the algorithms. The input data consists of mainly six attributes which are High Price, Low Price, Open Price, Close Price, and Volume.

4.1.1 Price Difference

It is the difference between the closing price and the opening price of the share value .this difference specifies weather share price is increasing, decreasing or constant.

4.1.2 High Price

It is the highest value of the share price of the company reached on that particular day.

4.1.3 Low Price

It is the lowest value of the share price of the company reached on that particular day.

4.1.4 Open Price

Open price is the price at the start of the day at which stock is traded during a regular session. For NSE/BSE regular trading sessions runs from Monday to Friday starting from 9 am in the morning to 3.30 in afternoon

4.1.5 Closing Price

Closing price is the price at the end of the day at which stock is traded during a regular session, it is the price of company's share at 3.30 pm.

4.1.6 Volume

Volume is the total numbers of shares traded in that particular day for a particular company.

4.2 Modules used

For building the project we have made three different modules and checked the results of all the modules and compared the results and depicted the best module which can be used for prediction. The comparison between the modules is done on the basis of the performance, accuracy, time taken by each of the algorithm. The modules used are Naïve Bayes, Support Vector Machine and Decision Tree.

4.2.1 Naïve Bayes Model

Now Let us understand this Model by taking the small example of stock data by considering the dataset:

Table 4.1: Training Data for Naïve Bayes

Date	Open	Close	High	Low	Volume	Class
12-04-2018	92.43	93.58	94.16	92.43	26758879	Low Performing
11-04-2018	92.01	91.86	93.29	91.48	24872110	Low Performing
10-04-2018	92.39	92.88	93.28	91.64	26939883	Low Performing
09-04-2018	91.04	90.77	93.17	90.62	31533943	High Performing
06-04-2018	91.49	90.23	92.46	89.48	38026000	High Performing
05-04-2018	92.435	92.38	93.065	91.4	29771881	Low Performing
04-04-2018	87.85	92.33	92.76	87.73	35559956	High Performing
03-04-2018	89.575	89.71	90.05	87.89	37213837	High Performing

1) List of Open for Low Performing-{92.43,92.01,92.39,92.435}
2) List Of Close for Low Performing-{93.58,91.86,92.88,92.38}
3) List Of High for Low Performing-{94.16,93.29,93.28,93.065}
4) List Of Low for Low Performing-{92.43,91.48,91.64,91.4}
5) List of Volume for Low Performing-{26758879, 24872110, 26939883, 29771881}

Now calculation of Mean is to be done for all attributes that Belong to the Low Performing

$$\bar{x} = \frac{\sum x}{N}$$

$\sum x =$ the sum of x

$N =$ number of data

Mean of Open for Low Performing= $\frac{92.43+92.01+92.39+92.435}{4}$

Mean of open for Low Performing=92.316

Mean of Close for Low Performing=92.675

Mean of High for Low Performing=93.4487

Mean of Low for Low Performing=91.735

Mean of Volume for Low Performing=27085688.25

$$\mathrm{SD} = \sqrt{\frac{\sum |x - \bar{x}|^2}{n}}$$

Standard Deviation of open for Low Performing=0.2051

Standard Deviation of Close for Low Performing=0.733

Standard Deviation of High for Low Performing=0.4853

Standard Deviation of Low for Low Performing=0.4723

Standard Deviation of Volume for Low Performing=2020199.95

List of Open for High Performing-{91.04, 91.49, 87.85, 89.575}

List of Close for High Performing-{90.77, 90.23, 92.33, 89.71}

List of High for High Performing-{93.17, 92.46, 92.76, 90.05}

List of Low for High Performing-{90.62, 89.48, 87.73, 87.89}

List of Volume for High Performing-{31533943, 38026000, 35559956, 37213837}

Mean of open for High Performing=89.98

Mean of Close for High Performing=90.76

Mean of High for High Performing=92.11

Mean of Low for High Performing=88.93

Mean of Volume for High Performing=35583434

Standard Deviation of open for High Performing=1.64

Standard Deviation of Close for High Performing=1.326

Standard Deviation of High for High Performing=1.40

Standard Deviation of Low for High Performing=1.376

Now let's take a Single Data point and predict the class for that data point:

Table 4.2 Test Data

Open	Close	High	Low	Volume
93	92	94.	91	30000000

By putting the values in the Bayesian formulae we get all the values of probability

P(open=93/low performing)=0.0072

P(close=92/low performing)=0.3717

P(high=94/low performing)=0.45

P(low=91/low performing)=0.253

P(data/low performing)=0.007 *0.37*0.45*0.253=0.00029

P(open=93/high performing)=0.044

P(close=92/high performing)=0.18

P(high=94/high performing)=0.11

P(low=91/high performing)=0.09

P(data/high performing)=0.044*0.18*0.11*0.09=0.29*0.86*0.38*0.78*0.35=0.000078

P(data/low performing) > P(data/high performing)

Hence we can say that this dataset belongs to the Low Performing Class.

Once the algorithm is run after all preprocessing of the data then the accuracy of the model is calculated .The Model is trained by taking the 80% of dataset and by remaining 20% the testing is done on the model and find out the classification output of 20% dataset. Once the Classification Output is predicted by the Model then it is compared with the original class from which it belong and weather Naïve Bayes have produced the right output or not ,based on that the accuracy of the model is calculated.

$$Accuracy = \frac{\text{Number of right predictions}}{Total\ predictions} \times 100$$

Accuracy can also be predicted by the help of the confusion matrix where confusion matrix is given in the table 4.3.

Table 4.3: confusion matrix

		Predicted class	
		P	*N*
Actual Class	*P*	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

$$Accuracy = \frac{TP + TN}{\mathrm{T} + \mathrm{N}}$$

- In our model we founded the accuracy of Naïve Bayes to be varying from 67% to 77 % because the data taken during the training and testing the model was randomly taken and on an average the accuracy of the model is 72%.
- Time taken to run the Naïve Bayes algorithm depends upon the how huge the dataset is if the dataset is large it will take more time to compute and train the model.in our model we have taken the data of stocks of past 5 years so for that the total time taken for the execution of the model is approximately 0.275 seconds. Time can be measured by tic and toc methods.
- Total memory taken by the algorithm to be executed is 144kb.

Accuracy is calculated by taking the average of 10 iterations performed by changing the test dataset and is plotted on the graph.

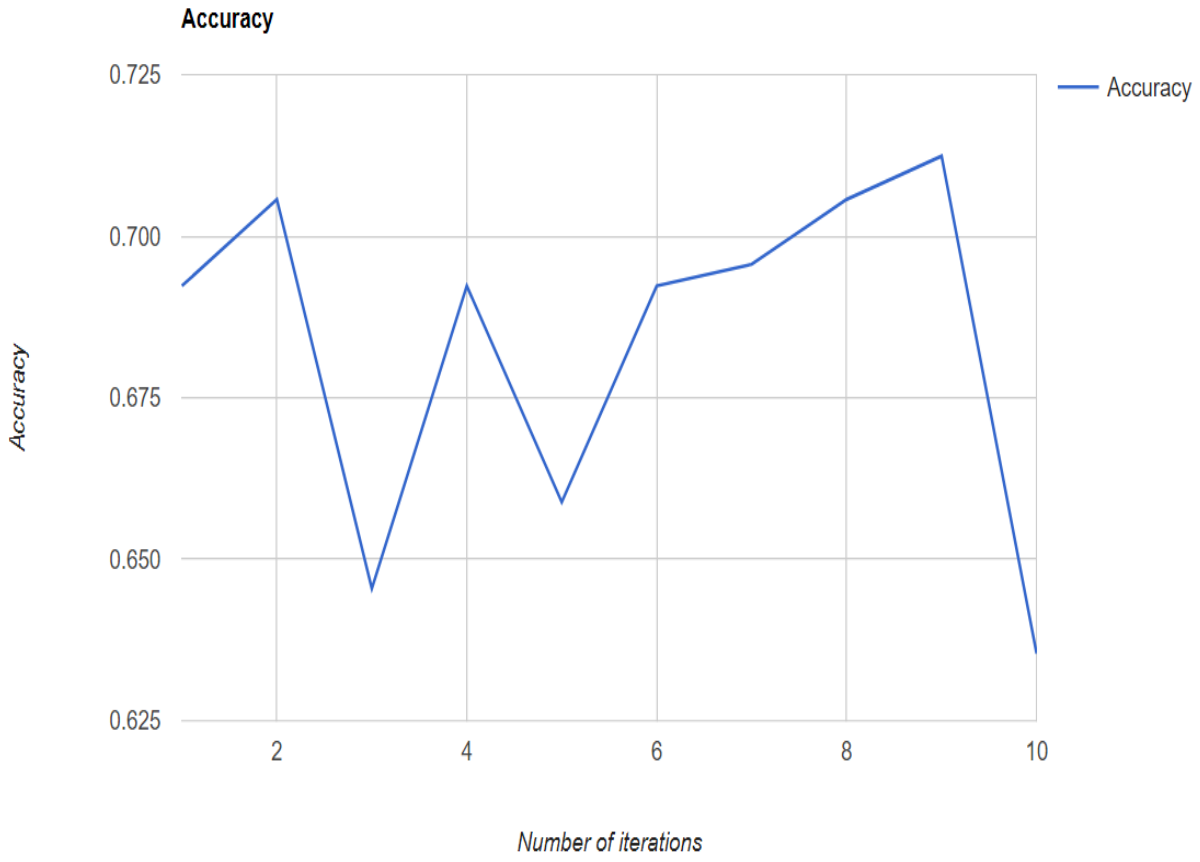

Fig. 4.1: Accuracy vs Iterations Graph

Time graph is also plotted for each of the iterations how much time did the algorithm takes to execute. On an average the algorithm takes 0.275 secs to execute.

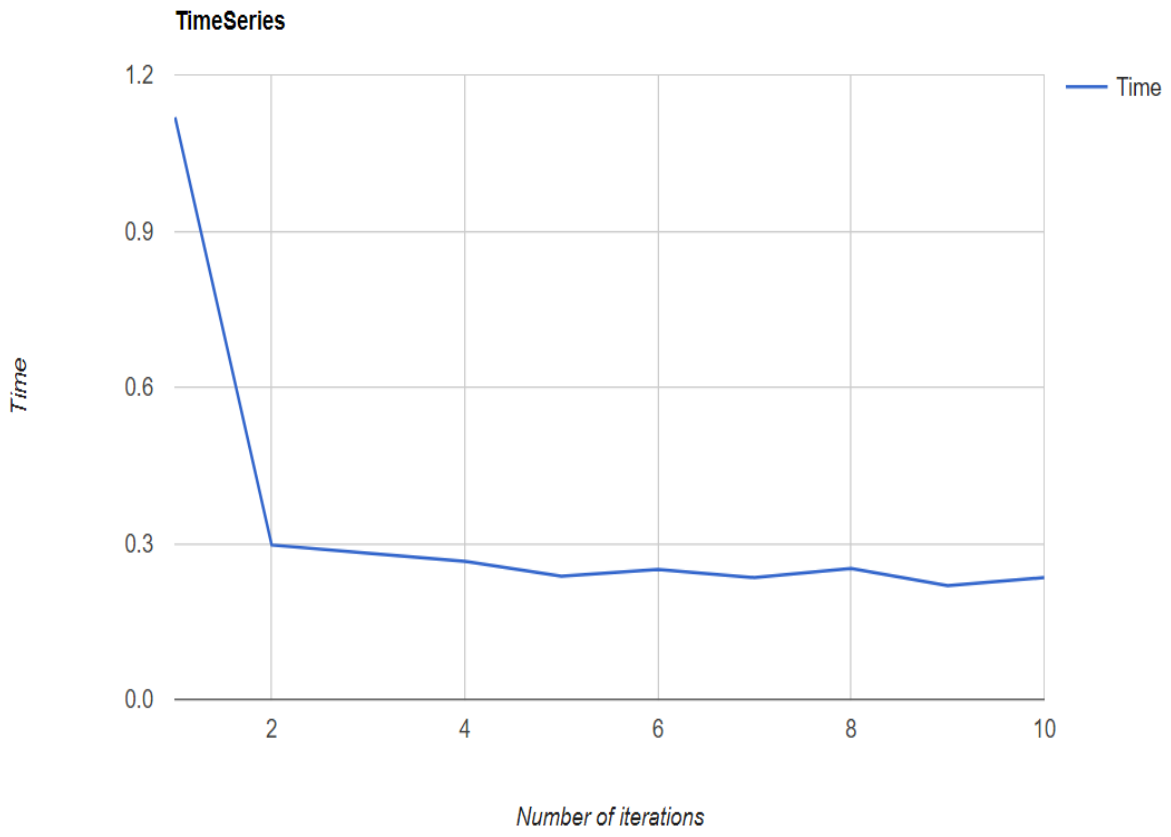

Fig. 4.2: Time taken vs Iterations Graph

Table 4.4: Naïve Bayes Confusion Matrix

	High Performing	Low Performing
High Performing	11	46
Low Performing	43	199

Accuracy= (11+199)/299

Accuracy= 70%

4.2.2 Decision Tree Model

Decision Tree Classifier is used to build a decision tree based on training data based on high, low, volume, open, close feature values. Decision Tree classifier is built and visualized using the rpart and rpart.plot package.

We input day-wise data of right size mostly three years or five years and partition the data into 70 percent training data and 30 percent test data.

The training data is used to build the model. We have used relationship among the features like close-open, high-low, high-close, and low-open to get better models.

The Decision tree is built based on these features and the model tree that is built depends on the size of training data. The split used to build is Gini index by default, we can also use information gain to split the data.

Since volume is the most variable feature among others, it usually used to compute the performance by the decision tree classifier

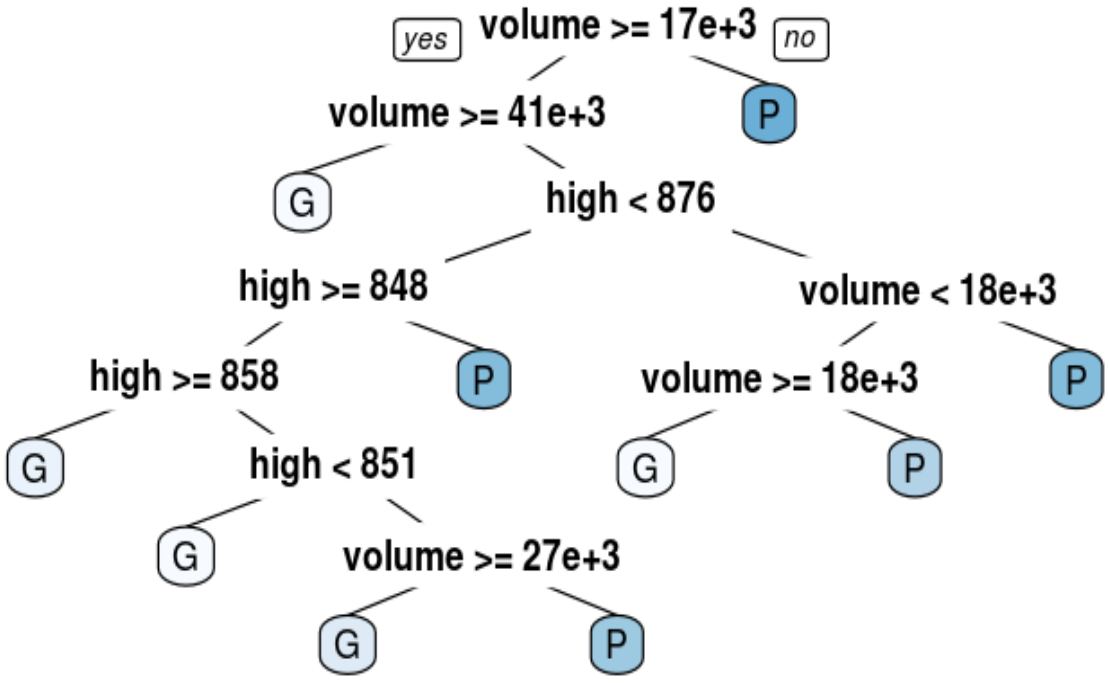

Fig. 4.3 Decision Tree Classifier

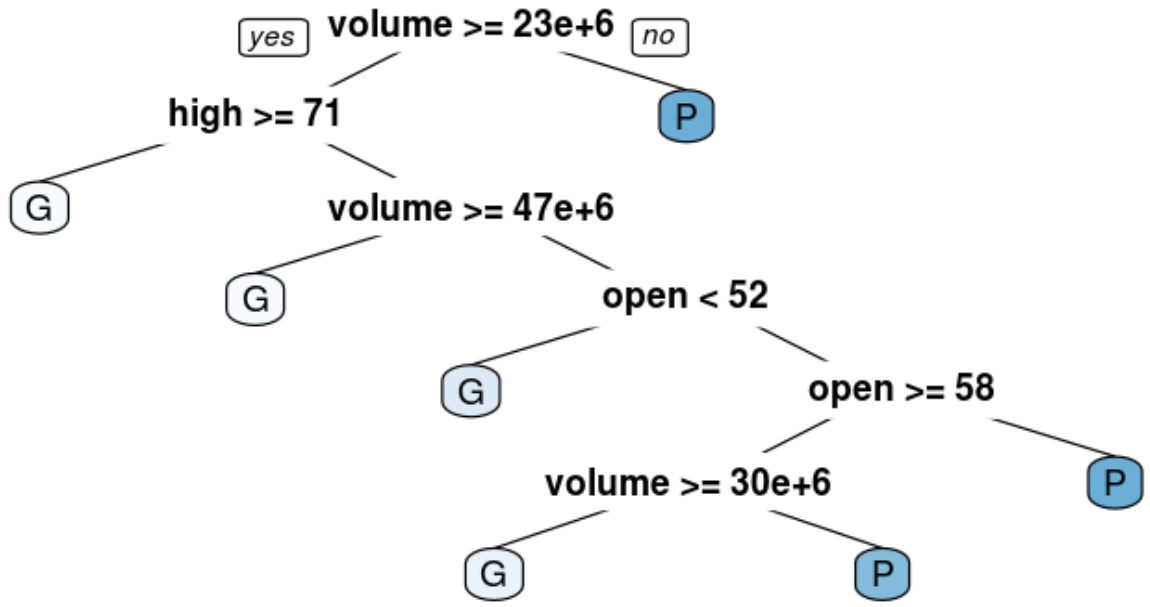

Fig. 4.4: Decision tree classifier2

Where class p refers to the Low Performing and class G represents the High Performing

Table 4.5: Confusion Matrix Decision Tree

	High Performing	Low Performing
High Performing	34	10
Low Performing	18	81

$$Accuracy = \frac{34+81}{143} * 100$$

Accuracy=80.41

When we compute accuracy for three years we get accuracy of 75-85% accuracy on test data.

This decision tree has average accuracy of 80 percent.

After analysis of the model when want to use the model on real-time data we understood that since we are applying the model on completely new data so dependency of the model on high, low, open, close might reduce the accuracy so we decide to use only volume as a feature to get better results.

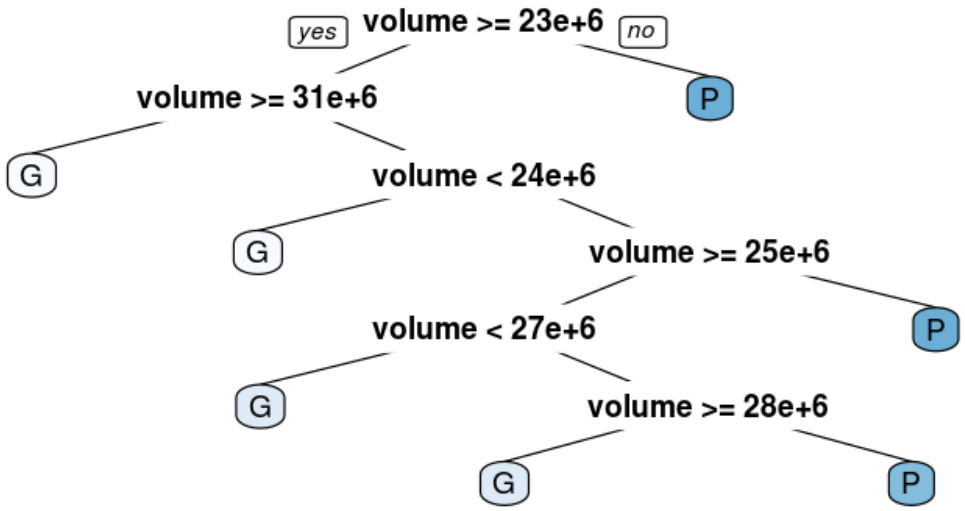

Fig 4.5 : Decision Tree when only volume is considered as factor

Table 4.6 : Confusion Matrix when Volume is considered

	High Performing	Low Performing
High Performing	33	26
Low Performing	15	73

$$Accuracy = \frac{33+73}{147} * 100$$

Accuracy=72.1%

This gives us accuracy of 70-75 percent on test data but it predicts the performance accurately for real-time stock data and the user can decide to buy/sell based on this for next week.

We try avoid over fit the tree model by using smaller data set usually for three-five years as usually simpler and wider tree gives better results.

4.2.3 Support Vector Machine Model

By applying the svm model on the stock data the accuracy of model is approximately 83.5%.

The graph of the accuracy is plotted vs the total iterations done.

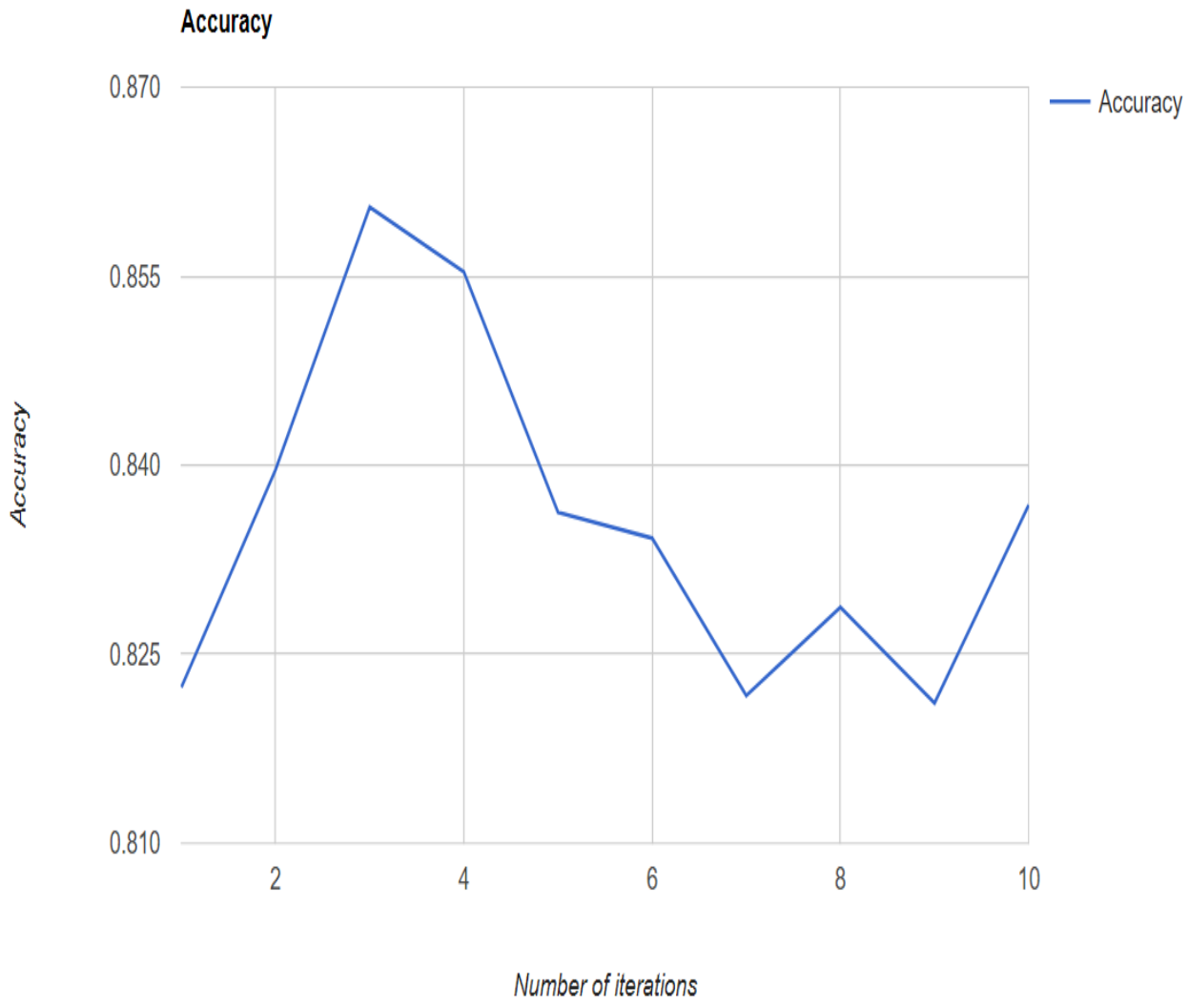

Fig 4.6 : Accuracy vs Iteration graph for SVM.

The time taken for running the algorithm varies and is approximately 0.3 sec .the plot of the time taken is plotted vs total no of iterations

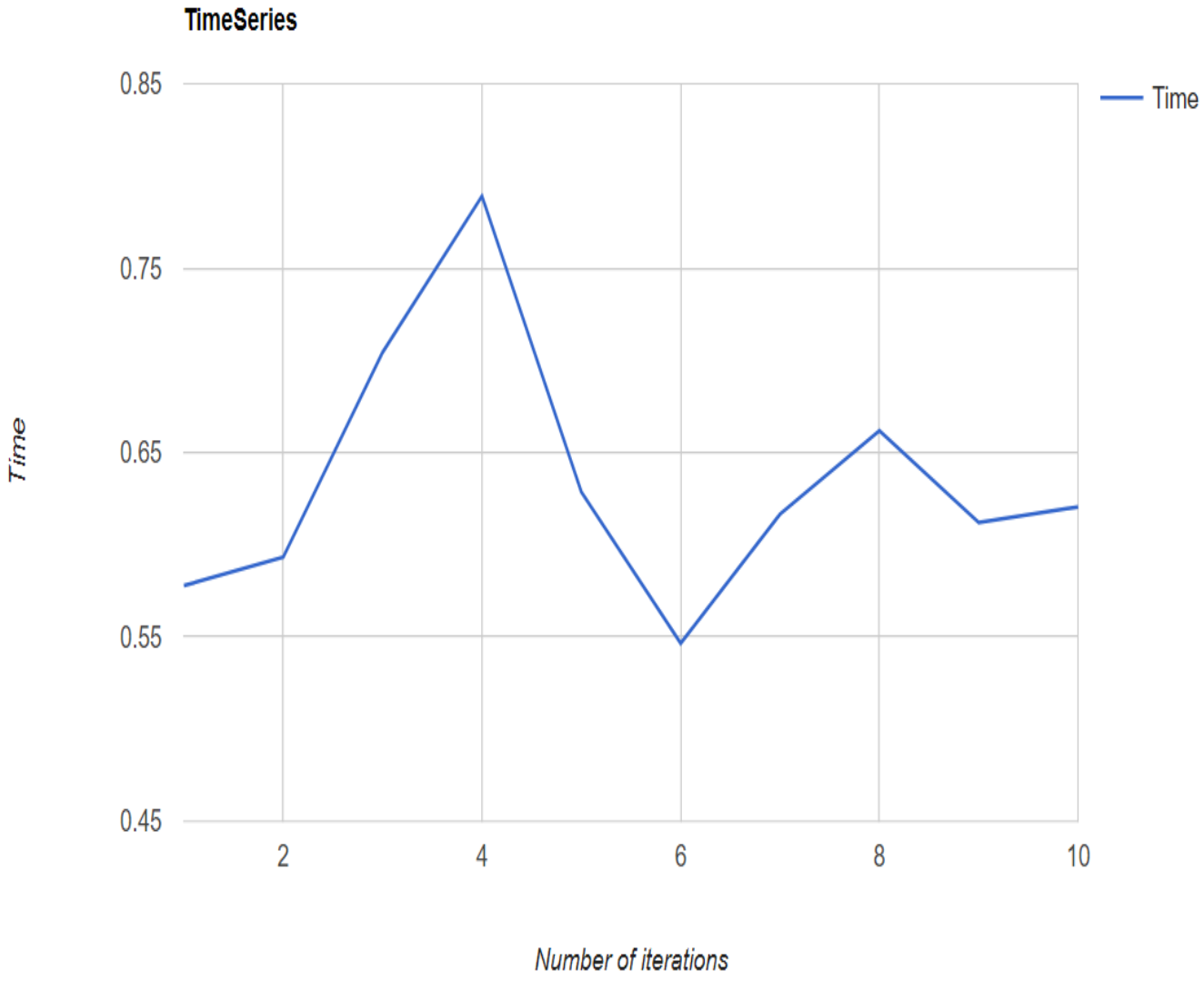

Fig 4.7 : Time vs iterations graph

Table 4.7 : Confusion Matrix for svm test data prediction

	High Performing	Low Performing
High Performing	15	4
Low Performing	68	286

$$Accuracy = \frac{15+286}{15+4+68+286}$$

Accuracy= 80.61 %

Chapter 5

Testing and Results

5. Testing and Results

5.1 Testing

Software testing is an action to check whether the real outcomes coordinate the normal outcomes and to ensure that the software system is defect free. Software testing additionally distinguishes blunders, gaps or missing requirements in contrary to the actual requirements. The testing's done in this projects are unit testing and performance testing

5.1.1. Unit Testing

Unit testing deals with testing a unit as a whole. This would test the interaction of many functions but tests within one unit. The exact scope of a unit is left to interpretation. In this project, the three modules as a whole constitutes into one application. The testing results done on the whole application are same as testing done on individual modules. We tested our all the three modules with the test data and observed that the accuracy of all three models was quite appreciable. For different companies the accuracy variation was on average 5% to 10% for the applied models.

The accuracy for naïve Bayes varies from 68% to 76% for different companies' data, which is quite good. On application of other models that is decision tree, the average accuracy was 72% and maximum accuracy touches to 79%, while for support vector machine accuracy varies from 80 to 85%.

Our models doesn't deflect much on taking different companies data.

5.1.2. Performance testing

Where the performance requirements, if any, are checked. These may include amount of main memory and/or secondary storage it requires and the demands made of the operating when

running with normal limits or the response time, system perform well after taking the huge data also the system should be able to predict with the high accuracy. The time system was taking to train the model for 5 years of data varies from 0.4 sec for Naïve Bayes, 0.2 sec for decision tree and 0.65 sec for support vector machine.

System memory consumption was linearly dependent on data, while the time was not linearly dependent, it took less than linear dependency.

5.1 Result

In this project, we have performed Support vector machine, Decision tree and Naive Bayes as Classifier algorithm to predict the label of our stock whether it is low performing or high performing where low performance says whether we should buy the present stock and sell it after 30 days. After performing all algorithm's we found that on performing naive Bayes to classify the class, the average accuracy of the naive Bayes comes close to 72% and the average time took by it was 0.275 sec for the data over the period of 5 years.

On performing Decision tree to classify the class, the average accuracy comes close to 76% and the average time took by it was 0.2 sec for the data over the period of 5 years.

On performing Support vector machine algorithm to classify the class, the average accuracy comes close to 83.5% and the average time took by it was 0.3 sec for the data over the period of 5 years. So if we look at the accuracy the SVM outperform both Decision tree and Naive Bayes whereas if we look at the time taken by the algorithms the Decision tree outperform both Naive Bayes and SVM.But Overall we can say SVM is better than both decision tree and naive Bayes because time difference is not much of both algorithm but the accuracy difference is greater.

So Support vector machine is better both than naive Bayes and Decision tree for classification purpose.

Chapter 6

Conclusion and Future Work

6. Conclusion and Future Work

The project has a significant usage in predicting the trend of the future market. Various companies, having their stocks in different share holdings, can influence their marketing strategies based on the knowledge of current and future behaviour of the stock market.
This study presents a proposal to use different classifiers on the historical prices of the stocks to predict buy or sell recommendations in the stock market. The three classifiers which we utilized as a part of our undertaking foresee distinctive level of level of precision.

After applying all three models, we came to the conclusion that Support vector machine is better than both decision tree and naive Bayes in terms of accuracy and in terms of time taken by individual model, decision tree is fastest of all three models used.

Concerning the future work, there is still enormous space for enhancing the proposed models, such improvement can be attained by adding additional variables, particularly those showing aspects of the company not related to profitability or earnings-share relationships. Sentimental Analysis can also be used along with the previously used factors as news about the particular company has great impact on their respective stock prices.

Bibliography

[1]“Automated Stock Trading Using Machine Learning Algorithms” by Tianxin Dai (Computer Science Department, Stanford University), Arpan Shah (Computer Science Department, Stanford University), Hongxia Zhong (Computer Science Department, Stanford University)

[2] “ALGORITHMIC TRADING USING MACHINE LEARNING T ECHNIQUES” by Chenxu Shao (Department of Management Science and Engineering, Stanford University), Zheming Zheng(Department of Management Science and Engineering, Stanford University).

[3]”Machine Learning in Stock Price Trend Forecasting” Yuqing Dai(Masters in Financial Mathematics, Stanford University), Yuning Zhang(Masters in Management Science and Engineering, Stanford University)

[4]R. Wilson and R. Sharda, "Bankruptcy prediction using neural networks", Decision Support Systems, vol. 11, no. 5, pp. 545-557, 1994.

[5]Atsalakis, G. S., Dimitrakakis, E. M., & Zopounidis, C. D. (2011). Elliott wave theory and neuro-fuzzy systems, in stock market prediction: The WASP system. Expert Systems with Applications, 38, 9196–920

[6]’’Stock Market Prediction” Radu Iacomin Faculty of Automatic Control and Computers (University POLITEHNICA of Bucharest, Romania)

[7] ’’Stocks Market Prediction Using Support Vector Machine’’ by Zhen Hu(College of Electronics and Information Engineering, Nanjing University of Technology, Nanjing, China),Jie Zhu(Marketing Department of Business School, Sun Yat-sen University, Guangzhou, China) and Ken Tse(Department of Computer Science, University of California, Los Angeles, USA)