**STOCK MARKET PREDICTION**

Project Report Submitted

In Partial Fulfillment of the Requirements

For the Degree Of

**BACHELOR OF ENGINEERING**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**Syed Ahmed Rafae (1604-17-733-020)**

**Syed Osman Hussain (1604-17-733-030)**

**Mohd Shakeeb (1604-17-733-036)**

****

**COMPUTER SCIENCE AND ENGINEERING DEPARTMENT**

**MUFFAKHAM JAH COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Affiliated to Osmania University)**

**Mount Pleasant, 8-2-249, Road No. 3, Banjara Hills, Hyderabad-34**

**CERTIFICATE**

This is to certify that the project dissertation titled “**STOCK MARKET PREDICTION***”* being submitted by

1. SYED AHMED RAFAE (1604-17-733-020)

2. SYED OSMAN HUSSAIN (1604-15-733-030)

3. MOHD SHAKEEB (1604-17-733-036)

in Partial Fulfillment of the requirements for the award of the degree of BACHELOR OF ENGINEERING IN COMPUTER SCIENCE AND ENGINEERING in MUFFAKHAM JAH COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyderabad for the academic year 2020-21 is the bonafide work carried out by them. The results embodied in this report have not been submitted to any other University or Institute for the award of any degree or diploma.

**Signatures**:

Internal Project Guide

(Mr. Mohd Abdul Raheem) (Dr. A.A. Moiz Qyser) (Associate Professor)  (Head CSED)

External Examiner

**DECLARATION**

This is to certify that the work reported in the major project entitled “**STOCK MARKET PREDICTION**” is a record of the bonafide work done by us in the Department of Computer Science and Engineering, Muffakham Jah College of Engineering and Technology, Osmania University. The results embodied in this report are based on the project work done entirely by us and not copied from any other source.

1. SYED AHMED RAFAE (1604-17-733-020)

2. SYED OSMAN HUSSAIN (1604-17-733-030)

3. MOHD SHAKEEB (1604-17-733-036)

**ACKNOWLEDGEMENT**

Our hearts are filled with gratitude to the Almighty for empowering us with courage, wisdom and strength to complete this project successfully. We give him all the glory, honor and praise.

We thank our Parents for having sacrificed a lot in their lives to impart the best education to us and make us promising professionals for tomorrow.

We would like to express our sincere gratitude and indebtness to our project supervisor **Mr. Mohd Abdul Raheem** for his valuable suggestions and interest throughout the course of this project.

We are happy to express our profound sense of gratitude and indebtedness to **Prof. Dr. Ahmed Abdul Moiz Qyser,** Head of the Computer Science and Engineering Department, for his valuable and intellectual suggestions apart from educate guidance constant encouragement right throughout our work and making successful.

With a great sense of pleasure and privilege, we extend our gratitude to **Prof. Dr. Syed Shabbeer Ahmed,** Associated Head of Computer Science and Engineering Department, project in-charge, offered valuable suggestions was a pre-requisite to carry out support in every step.

We are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

**SYED AHMED RAFAE**

**SYED OSMAN HUSSAIN**

**MOHD SHAKEEB**

**ABSTRACT**

The prediction of a stock market direction may serve as an early recommendation system for short-term investors and as an early financial distress warning system for long-term shareholders. Forecasting accuracy is the most important factor in selecting any forecasting methods. Research efforts in improving the accuracy of forecasting models are increasing since the last decade. The appropriate stock selections those are suitable for investment is a very difficult task. The key factor for each investor is to earn maximum profits on their investments.

Because of dependency on various factors, the stock prices are dynamic, highly noisy, and nonlinear time series data. The stock market prediction has always caught the attention of many analysts and researchers. Predicting stock prices is a challenging problem in itself because of the number of variables which are involved. In the short term, the market behaves like a voting machine but in the longer term, it acts like a weighing machine and hence there is scope for predicting the market movements for a longer timeframe. Application of machine learning techniques and other algorithms for stock price analysis and forecasting is an area that shows great promise. In this project, we first provide a concise review of stock markets, We then focus on some of the research achievements in stock analysis and prediction. We discuss technical, fundamental, short- and long-term approaches used for stock.

This project aims to shed light on the process of web scraping, emphasizing its importance in the new ’Big Data’ era with an illustrative application of such methods in financial markets. The work essentially focuses on different scraping methodologies that can be used to obtain large quantities of heterogeneous data in real-time. Automatization of data extraction systems is one of the main objectives pursued in this work, immediately followed by the development of a framework for predictive modeling. applying neural networks and deep learning methods to the data obtained through web scraping.

Anyone with nary an idea on the stock market will have a tool that may assist them in the quicksand that is day trading. The economy of our country is not a toy for huge corporations to fiddle with. It is the livelihood of many and retribution against the corporations is due with the first step being the mob short-selling of GameStop by Redditors. These methods are applied on 5 years of data retrieved from Yahoo Finance. The results will be used to analyze the stock prices and their prediction in depth in future research efforts.

iv



TABLE OF CONTENTS

TITLE.......................................................................................................................................................................... 1

CERIFICATE............................................................................................................................................................ 2

DECLARATION ...................................................................................................................................................... 3

ACKNOWLEDGEMENT ...................................................................................................................................... 4

ABSTRACT....................................................................................................................................................... 5

1

INTRODUCTION ........................................................................................................................................ 8

OBJECTIVE ........................................................................................................................................ 8

WHAT IS THE PROBLEM? ................................................................................................................ 8

WHY THIS IS A PROJECT RELATED TO THIS CLASS? ...................................................................... 8

WHY OTHER APPROACH IS NO GOOD? .......................................................................................... 8

WHY YOU THINK YOUR APPROACH IS BETTER? ............................................................................. 9

STATEMENT OF THE PROBLEM ....................................................................................................... 9

AREA OR SCOPE OF INVESTIGATION ............................................................................................ 10

1.2

1.3

1.4

1.5

1.6

1.7

2

THEORETICAL BASES AND LITERATURE REVIEW ................................................................................. 11

2.1

uiien

TABLE OF CONTENTS

TITLE.......................................................................................................................................................................... 1

CERIFICATE............................................................................................................................................................ 2

DECLARATION ...................................................................................................................................................... 3

ACKNOWLEDGEMENT ...................................................................................................................................... 4

ABSTRACT....................................................................................................................................................... 5

1. INTRODUCTION
   1. OBJECTIVE
   2. WHAT IS THE PROBLEM
   3. WHY IS THE SOLUTION RELATED TO THIS CLASS
   4. WHY OTHER APPRAOCH IS NOT ADOPTABLE
   5. WHY IS THIS SOLUTION BETTER
   6. STATEMENT OF PROBLEM
   7. AREA OF INVESTIGATION

DEFINITION OF THE PROBLEM ...................................................................................................... 11

THEORETICAL BACKGROUND OF THE PROBLEM .......................................................................... 11

RELATED RESEARCH TO SOLVE THE PROBLEM ............................................................................. 11

ADVANTAGE/DISADVANTAGE OF THOSE RESEARCH ................................................................... 12

OUR SOLUTION TO SOLVE THIS PROBLEM ................................................................................... 12

WHY OUR SOLUTION IS DIFFERENT FROM OTHERS? .................................................................. 12

WHY OUR SOLUTION IS BETTER? ................................................................................................. 12

2.2

2.3

2.4

2.5

2.6

2.7

3

4

HYPOTHESIS ........................................................................................................................................... 13

3.1

POSITIVE/NEGATIVE HYPOTHESIS ................................................................................................ 13

METHODOLOGY ..................................................................................................................................... 14

4.1

4.2

4.2.1

HOW TO COLLECT INPUT DATA?................................................................................................... 14

HOW TO SOLVE THE PROBLEM? ................................................................................................... 14

ALGORITHM DESIGN......................................................................................................16

USING DECISION TREES ...............................................................................................................16

USING SUPPORT VECTOR MACHINES ..............................................................................................17

4.2.2

R (PROGRAMMING LANGUAGE) [4] –.............................................................................................18

4.2.3 TOOLS USED ...............................................................................................................18

LANGUAGE USED..........................................................................................................18

RSTUDIO DESKTOP [5] –..............................................................................................................18



4.3

4.4

HOW TO GENERATE OUTPUT? ...................................................................................................... 18

HOW TO PROVE CORRECTNESS? .................................................................................................. 19

5

METHODOLOGY ..................................................................................................................................... 22

5.1 CODE .............................................................................................................................................. 22

5.1.1

5.1.2

5.2

DATA COLLECTION CODE....................................................................................................22

LSTM IMPLEMENTATION CODE........................................................................................25

DESIGN DOCUMENT AND FLOWCHART........................................................................................ 28

5.2.1 Design Document........................................................................................................................... 28

METHODS USED FOR INDICATORS...................................................................................................28

METHOD USED FOR DECISION TREE................................................................................................28

METHOD USED FOR PRUNING DECISION TREE ...................................................................................28

METHOD USED FOR PREDICTING THE OUTPUT ...................................................................................28

METHODS USED FOR EVALUATING THE MODEL..................................................................................28

METHOD USED FOR SVM.............................................................................................................29

METHOD USED FOR PREDICTING THE OUTPUT ...................................................................................29

METHODS USED FOR EVALUATING THE MODEL..................................................................................29

5.2.2 Flowchart........................................................................................................................................ 30

DATA ANALYSIS AND DISCUSSION ....................................................................................................... 31

6

6.1

6.2

6.3

6.4

6.5

OUTPUT GENERATION .................................................................................................................. 31

OUTPUT ANALYSIS ........................................................................................................................ 31

COMPARE OUTPUT AGAINST HYPOTHESIS ................................................................................. 33

STATISTIC REGRESSION................................................................................................................. 33

DISCUSSION ................................................................................................................................... 33

7

CONCLUSION AND RECOMMENDATIONS ............................................................................................. 34

7.1

7.2

SUMMARY AND CONCLUSION ....................................................................................................... 34

RECOMMENDATIONS FOR FUTURE STUDIES ............................................................................... 34

8

9

BIBLIOGRAPHY ....................................................................................................................................... 35

APPENDICES ........................................................................................................................................... 36

9.1

9.2

9.2.1

9.2.2

9.3

PROGRAM FLOWCHART ................................................................................................................ 36

PROGRAM SOURCE CODE AND DOCUMENTATION ..................................................................... 37

DECISION TREE IMPLEMENTATION CODE...........................................................................37

SVM IMPLEMENTATION CODE........................................................................................40

INPUT/OUTPUT LISTING ............................................................................................................... 43

4 | P a g e



Input 43

Output43

LIST OF FIGURES

Figure 1 : Steps to collect Input Data......................................................................................................... 14

Figure 2 : Steps to generate output........................................................................................................... 19

5 | P a g e

List of Tables

Table 1: Confusion Matrix.......................................................................................................................... 19

Table 2: Effect of Indicators on prediction accuracy ................................................................................. 32



6 | P a g e

+++



1 INTRODUCTION

1.1 OBJECTIVE

In the past decades, there is an increasing interest in predicting markets among economists,

policymakers, academics and market makers. The objective of the proposed work is to implement the supervised learning algorithms to predict the stock price.

A prediction system will be implemented using LSTM. The experimental objective will be to compare the forecasting ability of LSTM. We will test and evaluate the system with test data to find their prediction accuracy

The technical objectives will be implemented in python. The system must be able to access a list of historical prices. It must calculate the estimated price of stock based on the historical data. It

must also provide an instantaneous visualization of the market index.

1.2 WHAT IS THE PROBLEM?

Investors are familiar with the saying, “buy low, sell high” but this does not provide enough

context to make proper investment decisions. Before an investor invests in any stock, he needs

to be aware how the stock market behaves. Investing in a good stock but at a bad time can

have disastrous results, while investment in a mediocre stock at the right time can bear profits.

Financial investors of today are facing this problem of trading as they do not properly

understand as to which stocks to buy or which stocks to sell in order to get optimum profits.

Predicting long term value of the stock is relatively easy than predicting on day-to-day basis as

the stocks fluctuate rapidly every hour based on world events.

1.3 WHY THIS IS A PROJECT RELATED TO THIS CLASS?

The solution to this problem demands the use of tools and technologies related to the field of

data mining, pattern recognition, machine learning and data prediction. The application will

predict the stock prices for the next trading day. The requirements and the functionality of this

application correlates it to the class.

1.4 WHY OTHER APPROACH IS NOT GOOD?

The other approach makes use of Arima method [1]. Arima method have the following

drawbacks:

8 | P a g e



1.4.1 Short Term Prediction

ARIMA yields better results in forecasting short term predictions.

1.4.2. Univariate Data

Neural Networks are based on gradient descent method to find the local extreme value and

they have a tendency to get stuck on the local minima and maxima and therefore it is difficult

to find global minima and maxima. In the approach previously discussed, the author has used

pattern matching to overcome this problem.

1.5 WHY YOU THINK YOUR APPROACH IS BETTER?

The proposed approach makes use of Long Short Term Memory (LSTM).

The benefit of using Neural Network over Decision trees are:

1. They are easy to program.

2. The top nodes in the tree will give the information about what data affects the

prediction.

3. Trees are interpretable and provide visual representation of data.

4. Performs faster than Neural Networks after training.

5. LSTM has strong founding theory.

6. Global optimum guaranteed.

7. Requires less memory to store the predictive model.

8. Yield more readable results and a geometrical interpretation.

1.6 STATEMENT OF THE PROBLEM

Financial analysts investing in stock market usually are not aware of the stock market behavior.

They are facing the problem of trading as they do not properly understand which stocks to buy

or which stocks to sell in order to get more profits. In today’s world, all the information

pertaining to stock market is available. Analyzing all this information individually or manually is

tremendously difficult. As such, automation of the process is required. This is where Data

mining techniques help.

Understanding that analysis of numerical time series gives close results, intelligent investors use

machine learning techniques in predicting the stock market behavior. This will allow financial

analysts to foresee the behavior of the stock that they are interested in and thus act

accordingly.

The input to our system will be historical data from Yahoo Finance. Appropriate data would be

applied to find the stock price trends. Hence the prediction model will notify the up or down of

the stock price movement for the next trading day and investors can act upon it so as to

maximize their chances of gaining a profit. The entire system would be implemented in

9 | P a g e



Python language using open source libraries. Hence it will effectively be a zero cost

system.

1.7 AREA OR SCOPE OF INVESTIGATION

This project requires investigation in the following areas:

Stock Market

Investigating trends in stock market and factors affecting the stock prices.

Data mining techniques

Investigating the available tools and techniques for data mining and then selecting those that

are best fit to solve the problem.

10 | P a g e



2 THEORETICAL BASES AND LITERATURE REVIEW

2.1 DEFINITION OF THE PROBLEM

Stock market attracts thousands of investors’ hearts from all around the world. The risk and

profit of it has great charm and every investor wants to book profit from that. People use

various methods to predict market volatility, such as K-line diagram analysis method, Point Data

Diagram, Moving Average Convergence Divergence, even coin tossing, fortune telling, and so

on.

Now, all the financial data is stored digitally and is easily accessible. Availability of this huge

amount of financial data in digital media creates appropriate conditions for a data mining

research. The important problem in this area is to make effective use of the available data.

2.2 THEORETICAL BACKGROUND OF THE PROBLEM

Stock market is highly volatile. At the most fundamental level, it is said that supply and demand

in the market determines stock price. But, it does not follow any fixed pattern and is also

affected by a large number of highly varying factors

The investors on the Wall Street are split in two largest factions of adherents; those who

believe the market cannot be predicted and those who believe the market can be beat.

2.3 RELATED RESEARCH TO SOLVE THE PROBLEM

Recently, a lot of interesting work has been done in the area of applying Machine Learning

Algorithms for analyzing price patterns and predicting stock price. Most stock traders nowadays

depend on Intelligent Trading Systems which help them in predicting prices based on various

situations and conditions.

Recent researches uses input data from various sources and multiple forms. Some systems use

historical stock data, some use financial news articles, some use expert reviews while some use

a hybrid system which takes multiple inputs to predict the market.

Also, a wide range of machine learning algorithms are available that can be used to design the

system. These systems have different approaches to solve the problem. Some systems perform

mathematical analysis on historic data for prediction while some perform sentiment analysis on

financial news articles and expert reviews for prediction.

However, because of the volatility of the stock market, no system has a perfect or accurate

prediction.

11 | P a g e



2.4 ADVANTAGE/DISADVANTAGE OF THOSE RESEARCH

Advantages

The research helps a lot of new investors in deciding when to buy or sell a particular stock. It

also helps in understanding the sentiments of experienced financial analysts and financial news

data more quickly than doing the same manually.

Disadvantages

Current research makes use of neural networks which have the drawback of slow convergence rate and

local optimum. To overcome the problem of slow convergence the author uses a pattern matching

algorithm to select the input data to train the network which is an increased overhead [2].

2.5 OUR SOLUTION TO SOLVE THIS PROBLEM

We will implement the system using Recurrent Neural Network. In that we will use the LSTM method to predict results in a long term pattern with a decent accuracy

We will train both the systems using 75% of 4 years of historic data and then test our model to

check which systems yields better output using the remaining 25% of historic data.

2.6 WHY OUR SOLUTION IS DIFFERENT FROM OTHERS?

Our solution uses a different algorithm and different technique to perform the prediction. We

are using LSTM with C type classification and Radial Basis Function(RBF)

kernel.

2.7 WHY OUR SOLUTION IS BETTER?

It uses LSTM which have better performance than ARIMA and Prophet.

Moreover, LSTM takes away the burden of matching the present price pattern with

historic patterns and also it trains faster than a NN and has a lower computational cost.

NEURAL NETWORKS

NN CONT

12 | P a g e

NN CONT



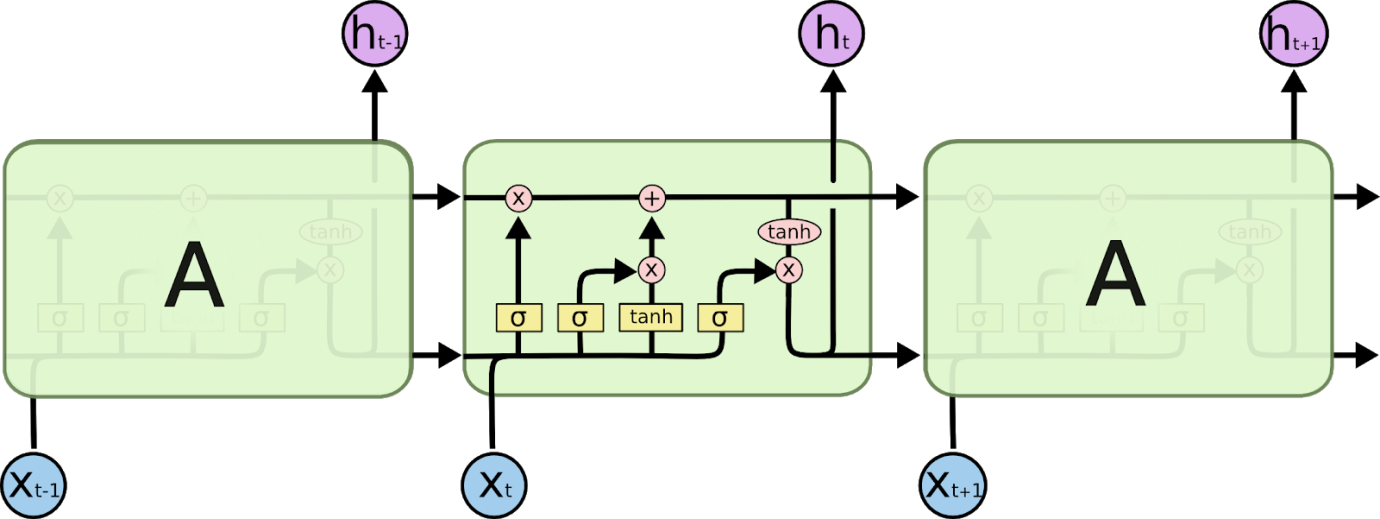
3 LSTM

Long Short Term Memory networks – usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies.

LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, not something they struggle to learn!

All recurrent neural networks have the form of a chain of repeating modules of neural networks. In standard RNNs, this repeating module will have a very simple structure.

LSTMs also have this chain-like structure, but the repeating module has a different structure. Instead of having a single neural network layer, there are four, interacting in a very special way.



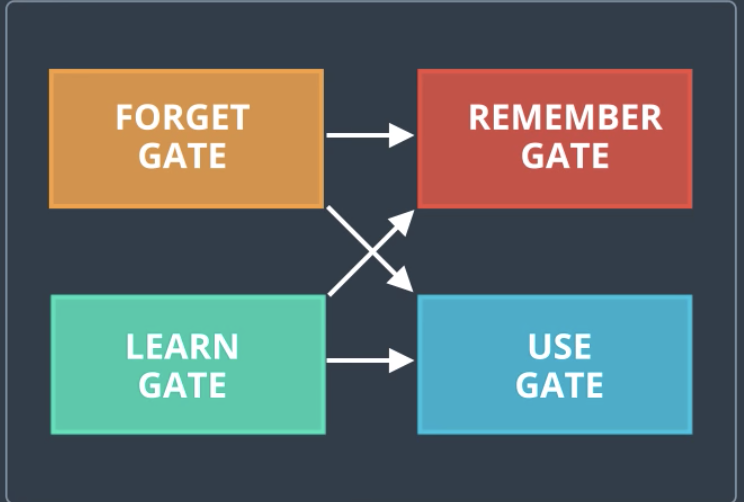
In the above diagram, each line carries an entire vector, from the output of one node to the inputs of others. The pink circles represent pointwise operations, like vector addition, while the yellow boxes are learned neural network layers. Lines merging denote concatenation, while a line forking denotes its content being copied and the copies going to different locations.

13 | P a g e



An LSTM has four “gates”: forget, remember, learn and use(or output)

It also has three inputs: long-term memory, short-term memory, and E. (E is some training example/new data)



**Learn Gate**

This gate combines existing Short-term memory (STM) and some input “E” , multiplies by a matrix (W) and adds b. Then squishes this all into a tanh function.



This combination gives us “N”.

Then it ignores some of the short-term memory, by multiplying the combined result by an “ignore factor” .

The ignore factor (I) is calculated by combining STM and E, with a new set of W(weights) and b(biases)



Once we have N and I, we multiply them together, and that’s the result of the learn gate.

We have “learned” our new information (E).

14 | P a g e

# Forget Gate

Forget gate is the gate you use to dump out all the unnecessary long term information. Kind of like when you study for a big exam, and the next day you forget everything. That’s the power of the forget gate.

Basically, the long-term memory (LTM) gets multiplied by a **forget factor (f).** This factor will make some of the long-term information be “forgotten”

The forget factor is this:



It is computed by taking the short-term memory, and input (E), multiplying them by some weights and biases and squishing them into a sigmoid function.

This function (f) gets multiplied by LTM — and boom, we’re left with LTM that we need.

# Remember Gate

This gate takes the information from the forget gate and adds it to the information from the learn gate, to compute the new long term memory.

Rember gate = Learn gate output + Forget gate output

# Use Gate

Use gate takes the LTM from the forget gate, and STM + E from the learn gate and uses them to come up with a new short term memory or an output (same thing).

For example, if we were trying to classify images, the output would be the network classification.

It takes the output of the learn gate, and applies a sigmoid function, so the equation looks like this:



Then, the gate multiplies V x U, to get the new short-term memory!

15 | P a g e

hfhfghfh

4 METHODOLOGY

4.1 HOW TO COLLECT INPUT DATA?

Input data is taken from Yahoo Finance using following steps:

1. For our project, we are considering S&P 500 Companies. The list of companies in S&P

500 can be obtained from Wikipedia [3].

2. Use stock’s ticker symbol from step a to get data from Yahoo Finance.

3. System will take last 4 years’ stock data of the company using pandas package in python.

4. Further we divide the data into two parts, training data and testing data, where 70% of

the data will be used for training and 30% of the data will be used for testing.

For our project,

we are

considering S&P

500 Companies.

The list of

• 70% of the data will

be used for training

•30% of the data will

be used for tesng

System will take

last 4 years stock

data of the

Use stock’s cker

symbol from

step a to get

data from Yahoo

Finance

company using

pandas

companies in

S&P 500 can

be obtained

package in R

from Wikipedia

Figure 1 : Steps to collect Input Data

4.2 HOW TO SOLVE THE PROBLEM?

To solve the problem, we will follow below steps -

1. Fetch the data of a stock from Yahoo Finance of last 4 years.

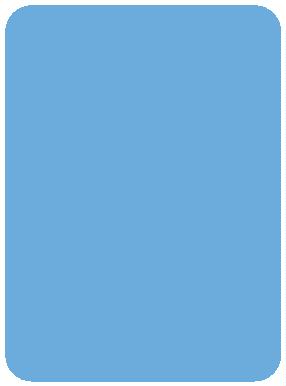
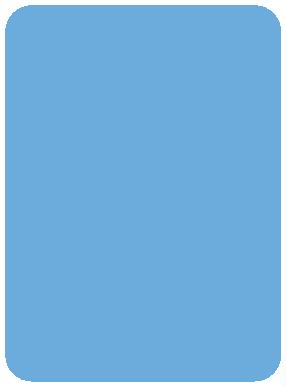
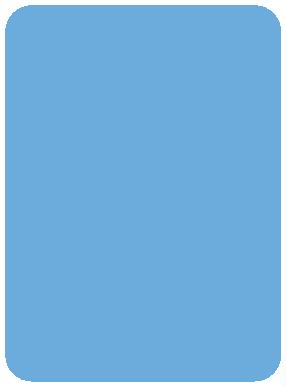
2. Calculate the values of technical indicators RSI, EMA, MACD, SMI, etc.

3. Train the model using these indicators and training data.

4. Test the model using testing data.

5. Evaluate our system using various evaluation techniques.

14 | P a g e



15 | P a g e

4.5 Terminologiesused

Given below is a brief summary of the various terminologies

relating to our proposed stock prediction system:

**1. Training set :** subsection of the original data that is

used to train the neural network model for

predicting the output values

**2. Test set** : part of the original data that is used to

make predictions of the output value, which are

then compared with the actual values to evaluate

the performance of the model

**3. Validation set** : portion of the original data that is

used to tune the parameters of the neural network

model

**5. Batch size** : number of samples that must be

processed by the model before updatingtheweights

of the parameters

**6. Epoch** : a complete pass through the given dataset

by the training algorithm

**7. Dropout**: a technique where randomly selected

neurons are ignored during training i.e., they are

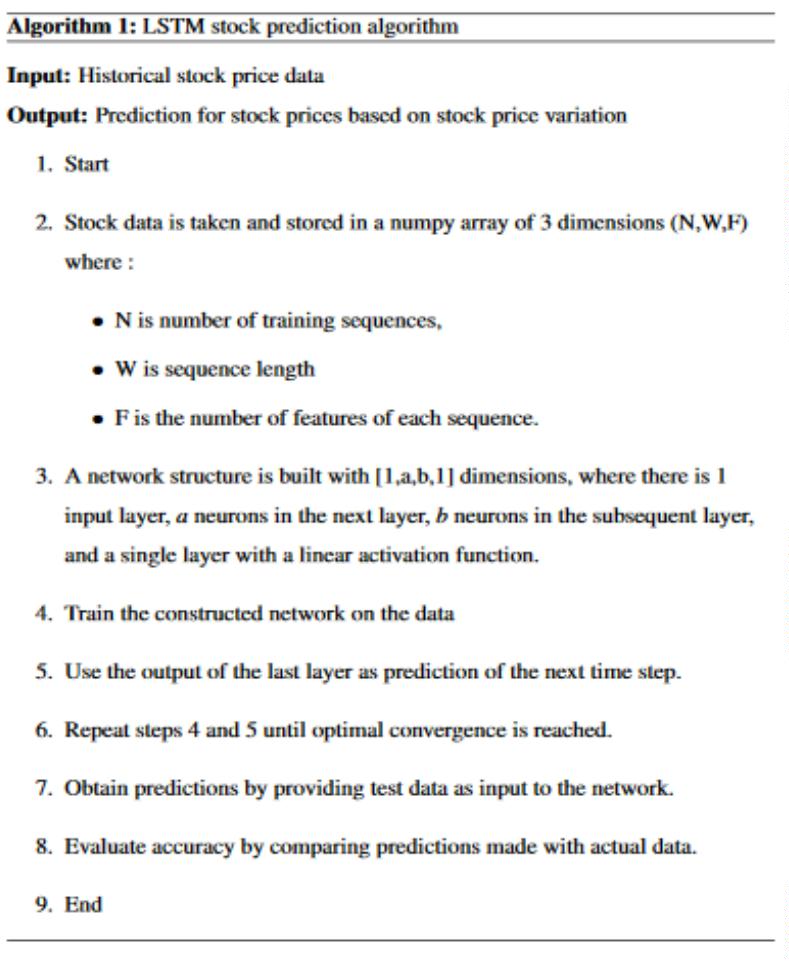
“dropped out” randomly. Thus, their contributionto

the activation of downstream neurons is temporally

removed on the forward pass, and any weight

updates are not applied to the neuron on the

backward pass.





•

•

•

16 | P a g e

5. OVERALL SYSTEM ARCHITECTURE

5.1 Obtaining dataset and preprocessing

Benchmark stock market data (for end-of-day prices of

various ticker symbols i.e., companies) was obtained from

two primary sources: Yahoo Finance and Google Finance.

These two websites offer URL-based APIs from which

historical stock data for various companies can be obtained

for various companies by simplyspecifyingsomeparameters

in the URL.

The obtained data contained five features:

1. Date: of the observation

2. Opening price: of the stock

3. High: highest intra-day price reached by the stock

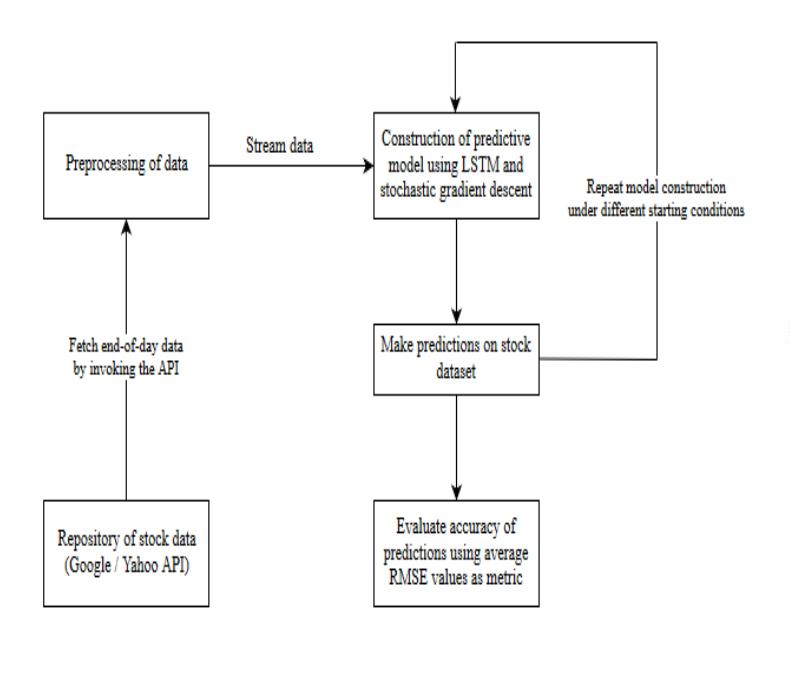
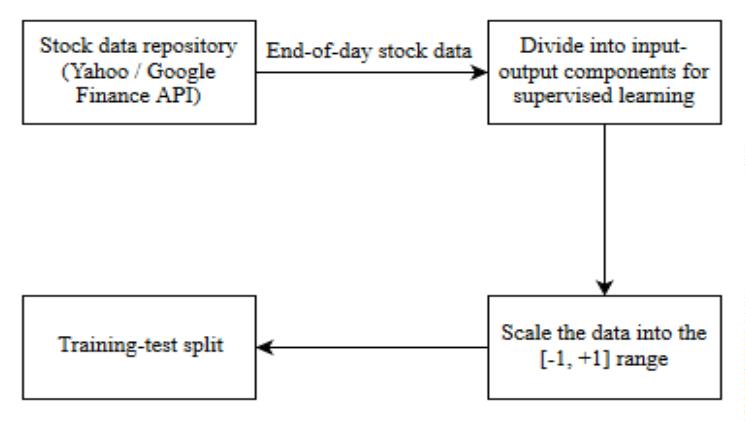
4. Low: lowest intra-day price reached by the stock

5. Volume: number of shares or contracts bought and

sold in the market during the day

6. OpenInt i.e., Open Interest: how many futures

contracts are currently outstanding in the market





5.2 Construction of prediction model

**Fig – 7:** Recurrent Neural Network structure for stock price prediction

5.3 Predictions and accuracy

**Fig – 8:** Prediction of end-of-day stock prices

Once the LSTM model is fit to the training data, it can be used

to predict the end-of-day stock price of an arbitrary stock.

**2.** Dynamic – a complex, more accurate approach

where the model is refit for each time step of the test

data as new observations are made available.

**1.** Static – a simple, less accurate method where the

model is fit on all the training data. Each new time

step is then predicted one at a time from test data.

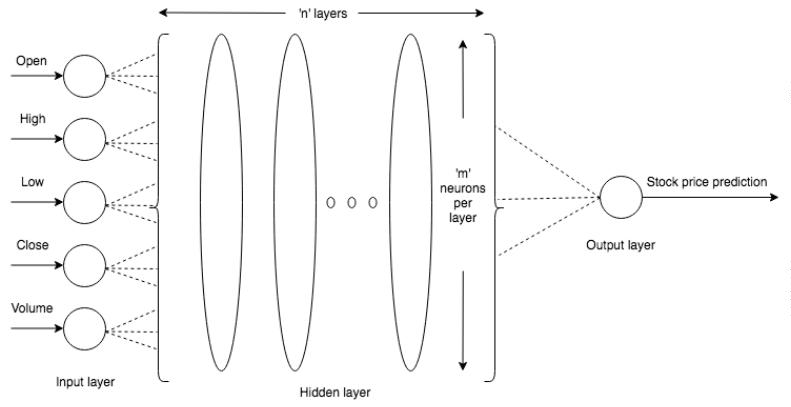
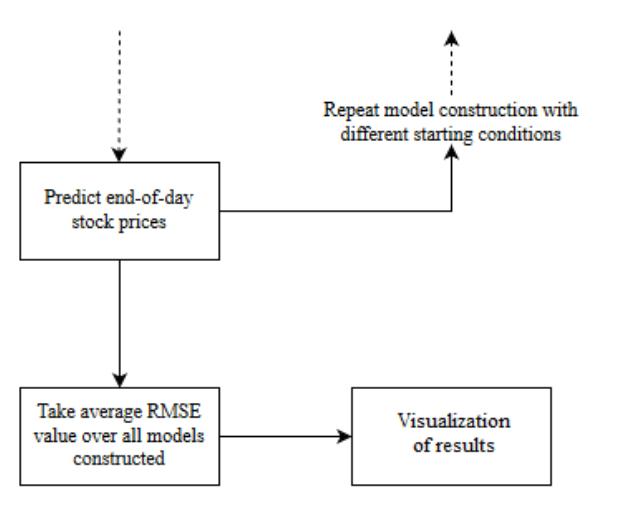
The accuracy of the prediction model can then be estimated

robustly using the RMSE (Root Mean Squared Error) metric.

This is due to the fact that neural networks in general

(including LSTM) tend to give different results with different

starting conditions on the same data.





4.2.1 ALGORITHM DESIGN

4.2.2 LANGUAGE USED

PYTHON (Programming Language) [4] –

Ra programming language and software environment for statistical computing and graphics

supported by the R Foundation for Statistical Computing. The R language is widely used among

statisticians and data miners for developing statistical software and data analysis. Polls, surveys of data

miners, and studies of scholarly literature databases show that R's popularity has increased substantially

in recent years.

Packages used -

• Panda data reader

• Dash

• Scikit learn

• Numpy

• Pandas

• Matplotlib

Microsoft Excel

4.3 HOW TO GENERATE OUTPUT?

Perform following steps to generate output:

a)

b)

c)

d)

Using pandas data reader library, get the last 4 years’ data.

Provide the data to the system.

Train the system.

System will predict the output.

18 | P a g e

VS code desc



Provide the

data to the

system.

System will

predict the

output.

Train the

System

Figure 2 : Steps to generate output

4.4 HOW TO PROVE CORRECTNESS?

Once we acquire a dataset, we intend to divide it into two subsets:

Training set is a subset of the dataset used to build predictive models.

Test set or unseen examples is a subset of the dataset to assess the likely future performance of

a model. If a model fit to the training set much better than it fits the test set, over fitting is

probably the cause.

Confusion Matrix

A confusion matrix shows the number of correct and incorrect predictions made by the

classification model compared to the actual outcomes (target value) in the data. The matrix is

NxN, where N is the number of target values (classes). Performance of such models is

commonly evaluated using the data in the matrix. The following table displays a 2x2 confusion

matrix for two classes (Positive and Negative).

Actual

Positive

Negative

Positive

True Positive

False Negative

False Positive

True Negative

Prediction

Negative

Accuracy (ACC) = (Σ True

positive + Σ True negative) /

Σ Total population

True positive rate (TPR) False positive rate (FPR)

= Σ True positive / Σ

Condition positive

= Σ False positive / Σ

Condition negative

Table 1: Confusion Matrix

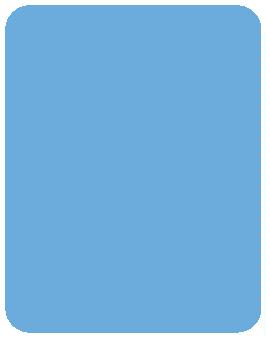
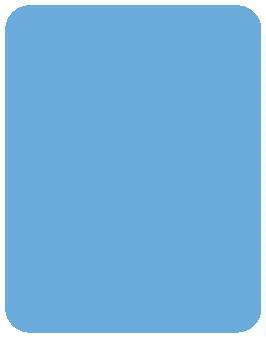
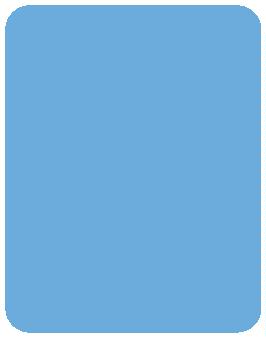
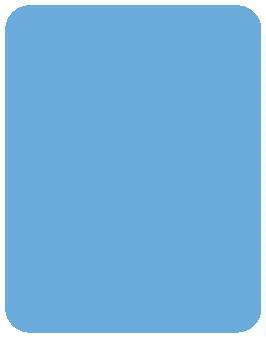
Accuracy: The proportion of the total number of predictions that were correct.

Positive Predictive Value or Precision: the proportion of positive cases that were correctly

identified.

19 | P a g e

Using pandas data reader library get the last 4 years’ data.



Negative Predictive Value: the proportion of negative cases that were correctly identified.

Sensitivity or Recall: the proportion of actual positive cases which are correctly identified.

Specificity: The proportion of actual negative cases which are correctly identified.

ROC Chart

The ROC chart is similar to the gain or lift charts in that they provide a means of comparison

between classification models. The ROC chart shows false positive rate (1-specificity) on X-axis,

the probability of target=1 when its true value is 0, against true positive rate (sensitivity) on Y-

axis, the probability of target=1 when its true value is 1. Ideally, the curve will climb quickly

toward the top-left meaning the model correctly predicted the cases. The diagonal red line is

for a random model (ROC101).

Figure 3: Receiver Operating Curve

Area Under the Curve (AUC)

Area under ROC curve is often used as a measure of quality of the classification models. A

random classifier has an area under the curve of 0.5, while AUC for a perfect classifier is equal

to 1. In practice, most of the classification models have an AUC between 0.5 and 1.

An area under the ROC curve of 0.8, for example, means that a randomly selected case from

the group with the target equals 1 has a score larger than that for a randomly chosen case from

the group with the target equals 0 in 80% of the time. When a classifier cannot distinguish

between the two groups, the area will be equal to 0.5 (the ROC curve will coincide with the

diagonal). When there is a perfect separation of the two groups, i.e., no overlapping of the

distributions, the area under the ROC curve reaches to 1 (the ROC curve will reach the upper

left corner of the plot).

20 | P a g e

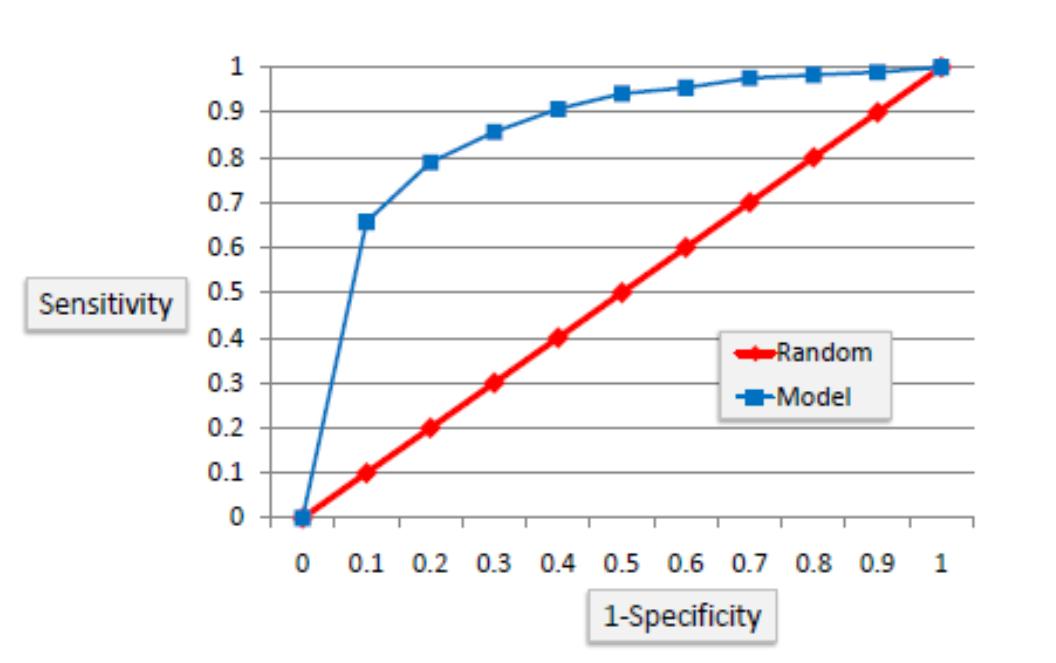


Figure 4: Area Under Curve

RMS Error

The regression line predicts the average y value associated with a given x value. Note that is

also necessary to get a measure of the spread of the y values around that average. To do this,

we use the root-mean-square error (RMS error).

To construct the RMS error, we first need to determine the residuals. Residuals are the

difference between the actual values and the predicted values. We denoted them by (p - a),

where a is the observed value for the ith observation and p is the predicted value.

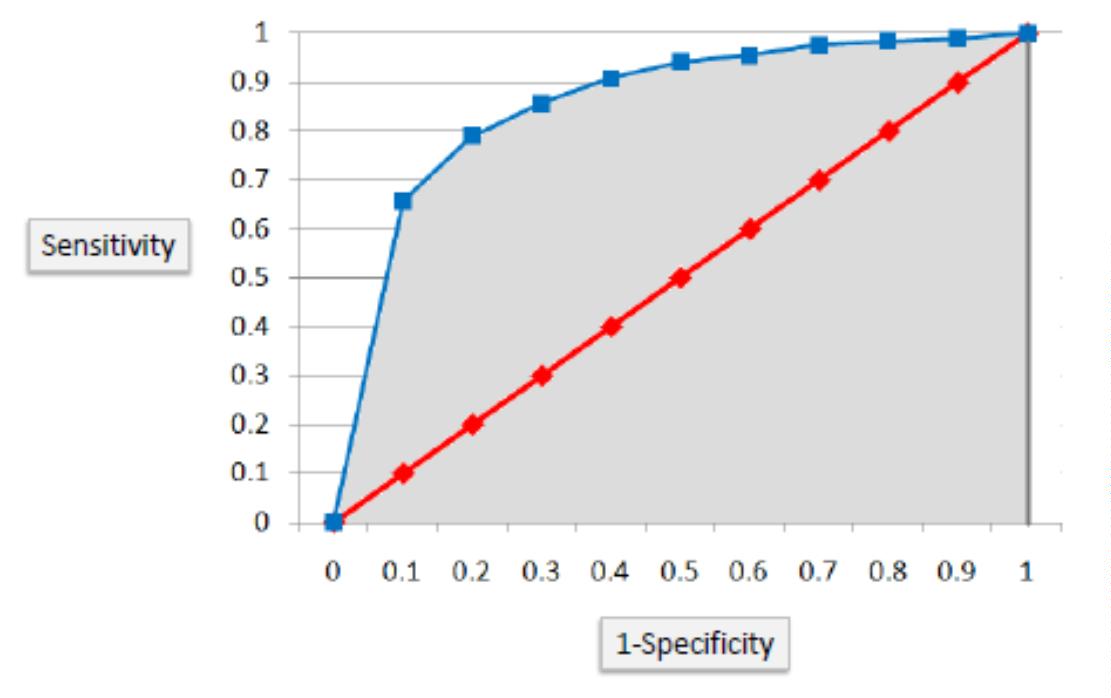
They can be positive or negative as the predicted value under or overestimates the actual

value. Squaring the residuals, averaging the squares, and taking the square root gives us the

RMS error. You then use the RMS error as a measure of the spread of the y values about the

predicted y value.

21 | P a g e



5 METHODOLOGY

5.1 CODE

5.1.1 Decision Tree Implementation Code

library(quantmod)

library(lubridate)

library(e1071)

library(rpart)

library(rpart.plot)

library(ROCR)

options(warn = -1)

tryCatch({

print('--------------------------------------------------------------------

-----')

SYM <- 'FB'

print('--------------------------------------------------------------------

-----')

print(paste('Predicting the output for', SYM, sep = ' '))

trainPerc <- 0.75

#Percent of data to be used as Training Data and remaining will be used as

Testing data

date <- as.Date(Sys.Date() - 1)

endDate <- date#as.Date("2016-01-01")

d <- as.POSIXlt(endDate)

d$year <- d$year - 2

#To take last 2 years of data

startDate <- as.Date(d)

STOCK <- getSymbols(

SYM,

env = NULL,

src = "yahoo",

from = startDate,

to = endDate

)

RSI <- RSI(Op(STOCK), n = 3)

#Calculate a 3-period relative strength index (RSI) off the open price

EMA <- EMA(Op(STOCK), n = 5)

#Calculate a 5-period exponential moving average (EMA)

EMAcross <- Op(STOCK) - EMA

#Let us explore the difference between the open price and our 5-period EMA

MACD <- MACD(Op(STOCK),

fast = 12,

slow = 26,

signal = 9)

#Calculate a MACD with standard parameters

MACD <- MACD[, 2]

22 | P a g e



#Grab just the signal line to use as our indicator.

SMI <- SMI(

Op(STOCK),

n = 13,

slow = 25,

fast = 2,

signal = 9

)

#Stochastic Oscillator with standard parameters

SMI <- SMI[, 1]

#Grab just the oscillator to use as our indicator

WPR <- WPR(Cl(STOCK), n = 14)

WPR <- WPR[, 1]

#Williams %R with standard parameters

ADX <- ADX(STOCK, n = 14)

ADX <- ADX[, 1]

#Average Directional Index with standard parameters

CCI <- CCI(Cl(STOCK), n = 14)

CCI <- CCI[, 1]

#Commodity Channel Index with standard parameters

CMO <- CMO(Cl(STOCK), n = 14)

CMO <- CMO[, 1]

#Collateralized Mortgage Obligation with standard parameters

ROC <- ROC(Cl(STOCK), n = 2)

ROC <- ROC[, 1]

#Price Rate Of Change with standard parameters

PriceChange <- Cl(STOCK) - Op(STOCK)

#Calculate the difference between the close price and open price

Class <- ifelse(PriceChange > 0, "UP", "DOWN")

#Create a binary classification variable, the variable we are trying to pre

dict.

DataSet <-

data.frame(Class, RSI, EMAcross, MACD, SMI, WPR, ADX, CCI, CMO, ROC)

#Create our data set

colnames(DataSet) <-

c("Class",

"RSI",

"EMAcross",

"MACD",

"SMI",

"WPR",

"ADX",

"CCI",

"CMO",

"ROC")

#Name the columns

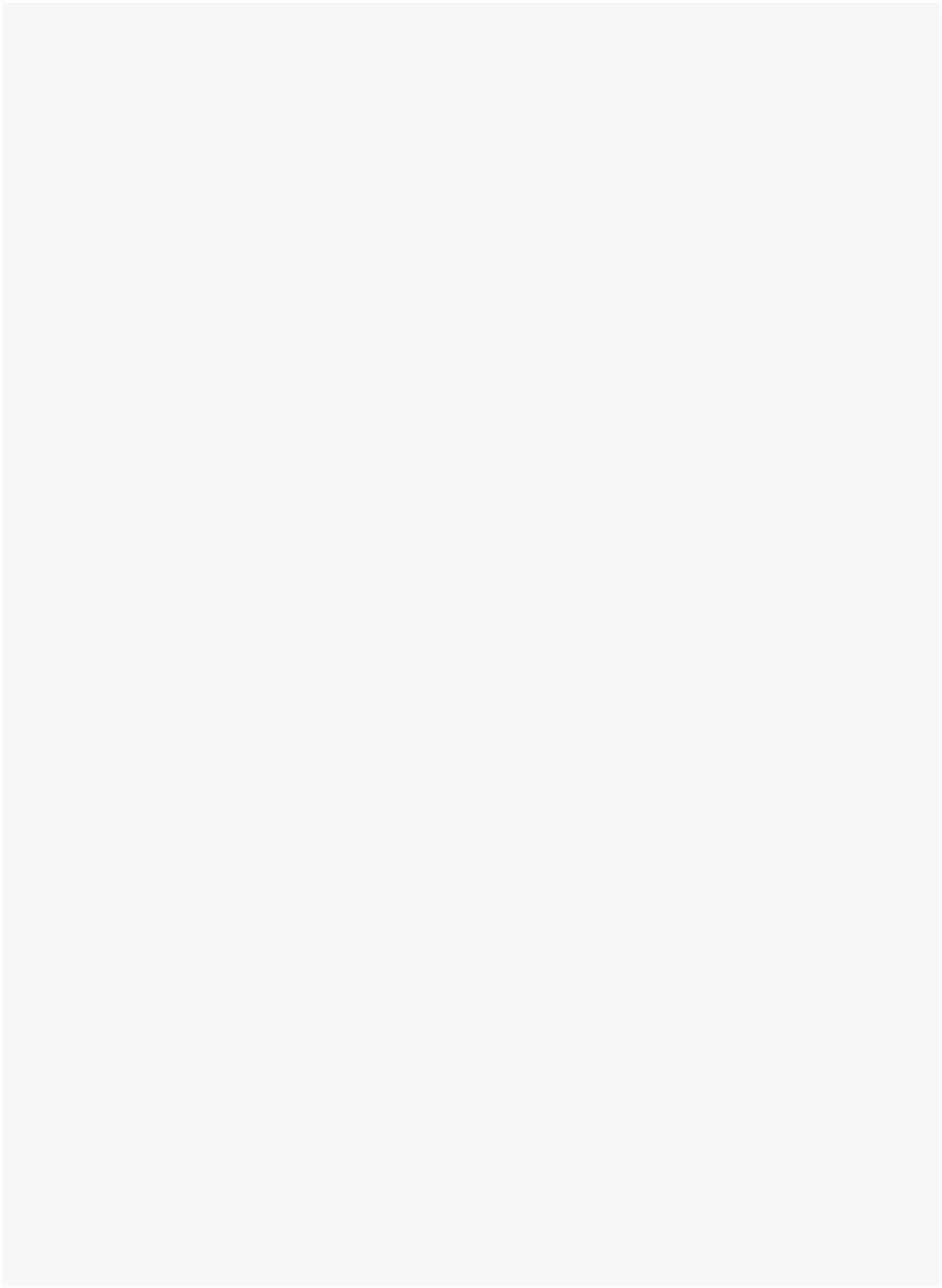
#DataSet <- DataSet[-c(1:33),]

#Get rid of the data where the indicators are being calculated

TrainingSet <- DataSet[1:floor(nrow(DataSet) \* trainPerc), ]

#Use 2/3 of the data to build the tree

23 | P a g e



TestSet <-

DataSet[(floor(nrow(DataSet) \* trainPerc) + 1):nrow(DataSet), ]

#And leave out 1/3 data to test our strategy

DecisionTree <-

rpart(

Class ~ RSI + EMAcross + WPR + ADX + CMO + CCI + ROC,

data = TrainingSet,

na.action = na.omit,

cp = .001

)

#Specifying the indicators to we want to use to predict the class and contr

olling the growth of the tree by setting the minimum amount of information ga

ined (cp) needed to justify a split.

prp(DecisionTree, type = 2, extra = 8)

#Plotting tool with a couple parameters to make it look good.

fit <- printcp(DecisionTree)

#Shows the minimal cp for each trees of each size.

mincp <- fit[which.min(fit[, 'xerror']), 'CP']

#Get the lowest cross-validated error (xerror)

plotcp(DecisionTree, upper = "splits")

#plots the average geometric mean for trees of each size.

PrunedDecisionTree <- prune(DecisionTree, cp = mincp)

#Selecting the complexity parameter (cp) that has the lowest cross-validate

d error (xerror)

t <- prp(PrunedDecisionTree, type = 2, extra = 8)

confmat <-

table(

predict(PrunedDecisionTree, TestSet, type = "class"),

TestSet[, 1],

dnn = list('predicted', 'actual')

)

#Building confusion matrix

print(confmat)

acc <-

(confmat[1, "DOWN"] + confmat[2, "UP"]) \* 100 / (confmat[2, "DOWN"] + con

fmat[1, "UP"] + confmat[1, "DOWN"] + confmat[2, "UP"])

#Calculating accuracy

xy <-

paste('Decision Tree : Considering the output for', SYM, sep = ' ')

yz <-

paste('Accuracy =',

acc,

sep = ' ')

print(xy)

print(yz)

predout <- data.frame(predict(PrunedDecisionTree, TestSet))

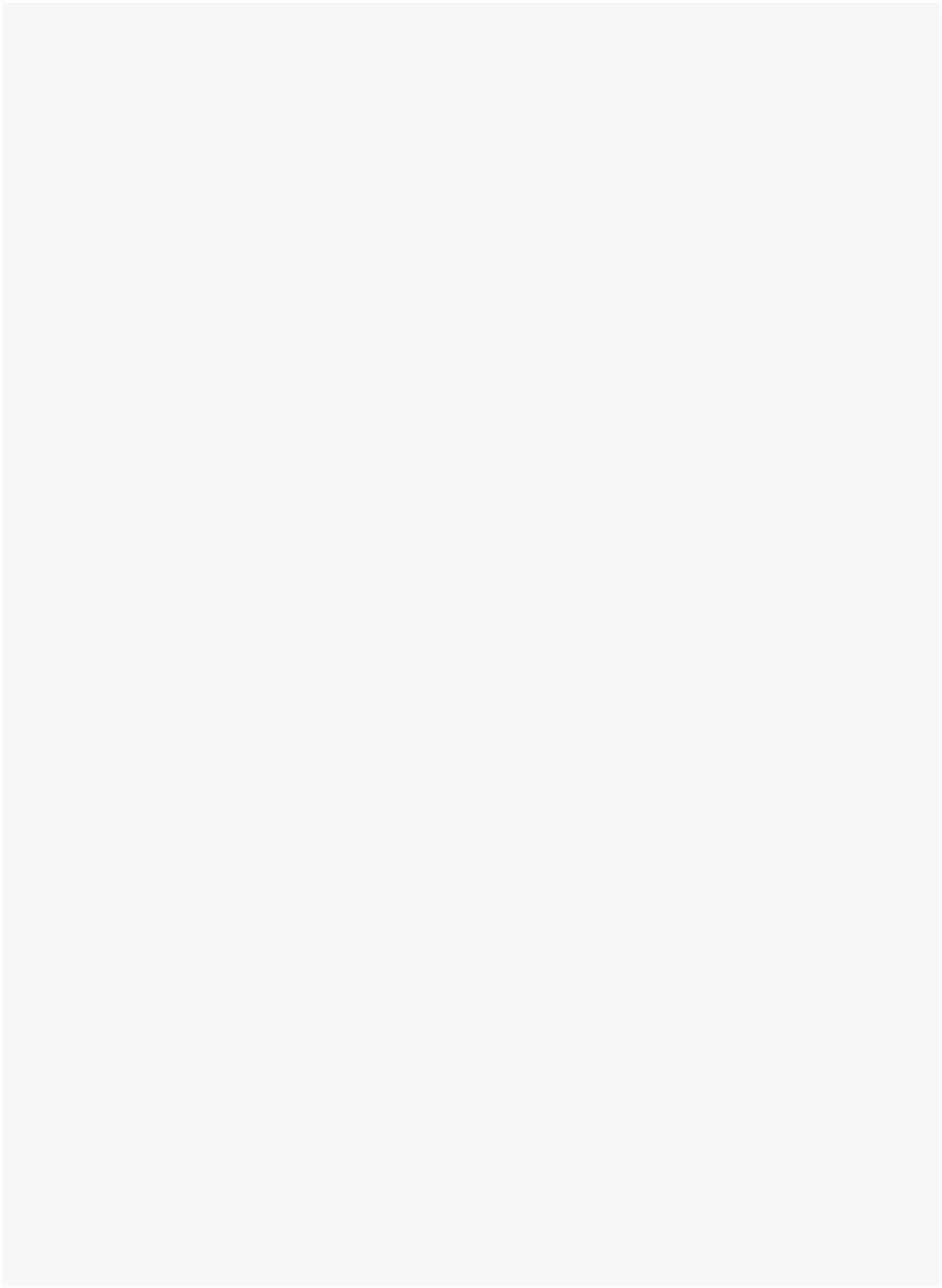
predval <- predout['UP'] - predout['DOWN']

predclass <- ifelse(predout['UP'] >= predout['DOWN'], 1, 0)

predds <- data.frame(predclass, TestSet$Class)

colnames(predds) <- c("pred", "truth")

24 | P a g e



predds[, 2] <- ifelse(predds[, 2] == 'UP', 1, 0)

pred <- prediction(predds$pred, predds$truth)

perf = performance(pred, measure = "tpr", x.measure = "fpr")

auc.perf = performance(pred, measure = 'auc')

#Calculating the AUC

rmse.perf = performance(pred, measure = 'rmse')

#Calculating the RMSE

RMSE <- paste('RMSE =', rmse.perf@y.values, sep = ' ')

AUC <- paste('AUC =', auc.perf@y.values, sep = ' ')

print(AUC)

print(RMSE)

plot(perf, col = 1:10)

abline(a = 0, b = 1, col = "red")

#Plotting ROC curve

print('--------------------------------------------------------------------

-----')

}, error = function(e) {

print(e)

})

5.1.2 SVM Implementation Code

library(quantmod)

library(lubridate)

library(e1071)

library(rpart)

library(rpart.plot)

library(ROCR)

options(warn = -1)

tryCatch({

SYM <- 'FB'

print('--------------------------------------------------------------------

-----')

print(paste('Predicting the output for', SYM, sep = ' '))

trainPerc <- 0.75

#Percent of data to be used as Training Data and remaining will be used as

Testing data

date <- as.Date(Sys.Date() - 1)

endDate <- date#as.Date("2016-01-01")

d <- as.POSIXlt(endDate)

d$year <- d$year - 2

#To take last 2 years of data

startDate <- as.Date(d)

STOCK <- getSymbols(

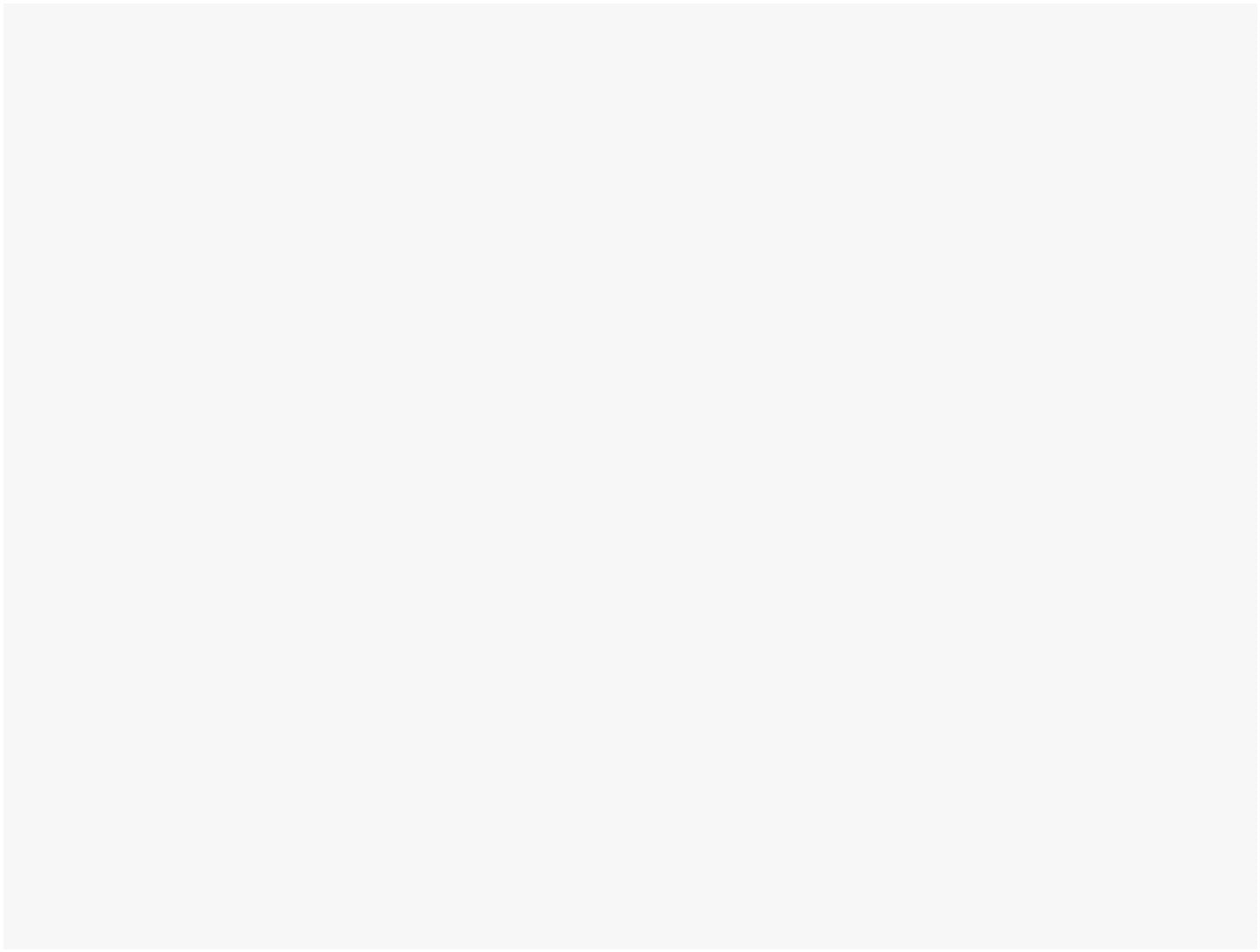
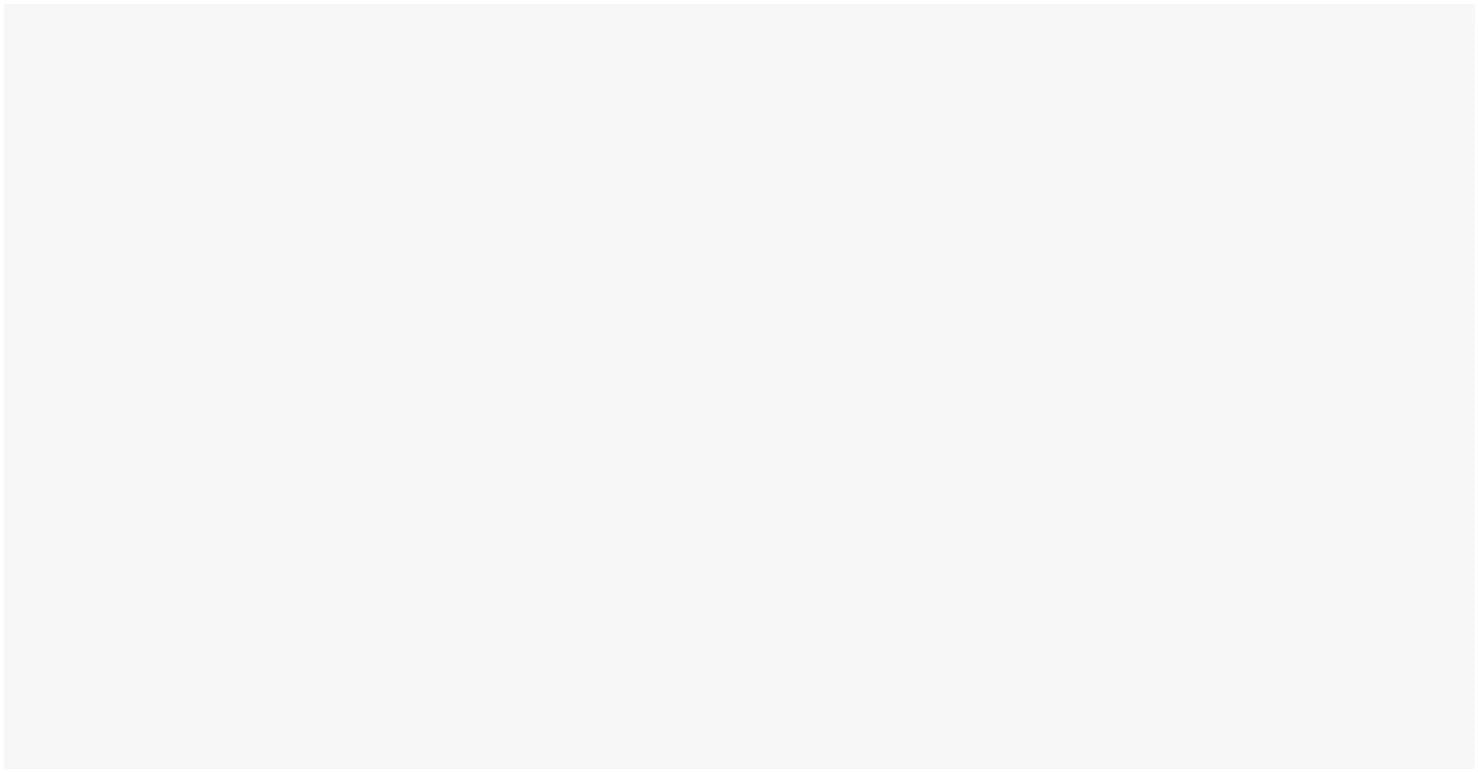
SYM,

env = NULL,

src = "yahoo",

from = startDate,

25 | P a g e



to = endDate

)

RSI <- RSI(Op(STOCK), n = 3)

#Calculate a 3-period relative strength index (RSI) off the open price

EMA5 <- EMA(Op(STOCK), n = 5)

#Calculate a 5-period exponential moving average (EMA)

EMAcross <- Op(STOCK) - EMA5

#Let us explore the difference between the open price and our 5-period EMA

MACD <- MACD(Op(STOCK),

fast = 12,

slow = 26,

signal = 9)

#Calculate a MACD with standard parameters

MACD <- MACD[, 2]

#Grab just the signal line to use as our indicator.

SMI <- SMI(

Op(STOCK),

n = 13,

slow = 25,

fast = 2,

signal = 9

)

#Stochastic Oscillator with standard parameters

SMI <- SMI[, 1]

#Grab just the oscillator to use as our indicator

WPR <- WPR(Cl(STOCK), n = 14)

WPR <- WPR[, 1]

#Williams %R with standard parameters

ADX <- ADX(STOCK, n = 14)

ADX <- ADX[, 1]

#Average Directional Index with standard parameters

CCI <- CCI(Cl(STOCK), n = 14)

CCI <- CCI[, 1]

#Commodity Channel Index with standard parameters

CMO <- CMO(Cl(STOCK), n = 14)

CMO <- CMO[, 1]

#Collateralized Mortgage Obligation with standard parameters

ROC <- ROC(Cl(STOCK), n = 2)

ROC <- ROC[, 1]

#Price Rate Of Change with standard parameters

PriceChange <- Cl(STOCK) - Op(STOCK)

#Calculate the difference between the close price and open price

Class <- ifelse(PriceChange > 0, 'UP', 'DOWN')

#Create a binary classification variable, the variable we are trying to pre

dict.

DataSet <-

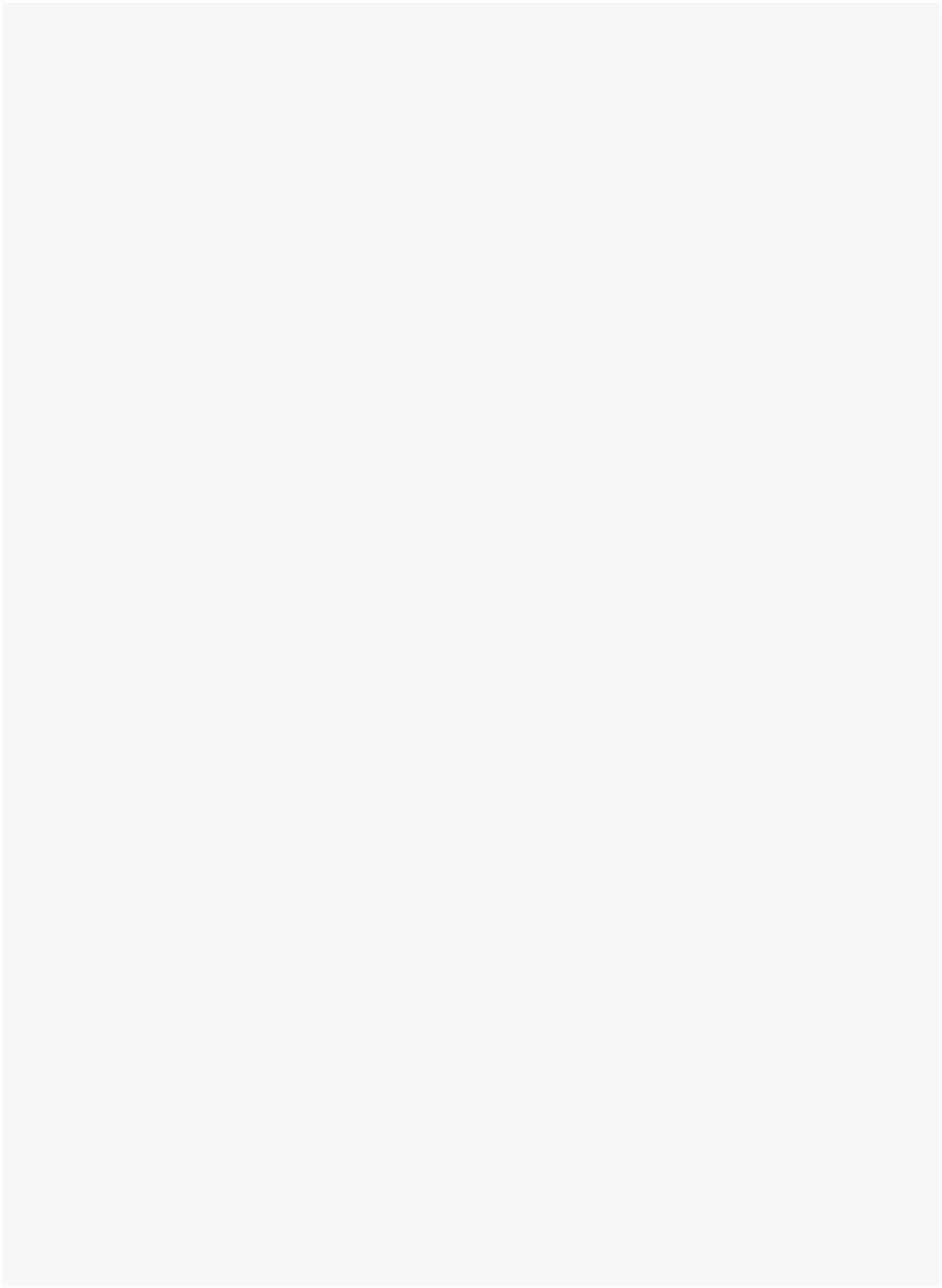
data.frame(Class, RSI, EMAcross, MACD, SMI, WPR, ADX, CCI, CMO, ROC)

#Create our data set

colnames(DataSet) <-

c("Class",

26 | P a g e



"RSI",

"EMAcross",

"MACD",

"SMI",

"WPR",

"ADX",

"CCI",

"CMO",

"ROC")

#Name the columns

#DataSet <- DataSet[-c(1:33), ]

#Get rid of the data where the indicators are being calculated

TrainingSet <- DataSet[1:floor(nrow(DataSet) \* trainPerc), ]

#Use 2/3 of the data to build the tree

TestSet <-

DataSet[(floor(nrow(DataSet) \* trainPerc) + 1):nrow(DataSet), ]

#And leave out 1/3 data to test our strategy

SVM <-

svm(

Class ~ RSI + EMAcross + WPR + ADX + CMO + CCI + ROC,

data = TrainingSet,

kernel = "radial",

type = "C-classification",

na.action = na.omit,

cost = 1,

gamma = 1 / 5

)

#Specifying the indicators to we want to use to predict the class.

print(SVM)

confmat <-

table(predict(SVM, TestSet, type = "class"),

TestSet[, 1],

dnn = list('predicted', 'actual'))

#Building confusion matrix

print(confmat)

acc <-

(confmat[1, "DOWN"] + confmat[2, "UP"]) \* 100 / (confmat[2, "DOWN"] + con

fmat[1, "UP"] + confmat[1, "DOWN"] + confmat[2, "UP"])

#Calculating accuracy

xy <- paste('SVM : Considering the output for', SYM, sep = ' ')

yz <-

paste('Accuracy =',

acc,

sep = ' ')

print(xy)

print(yz)

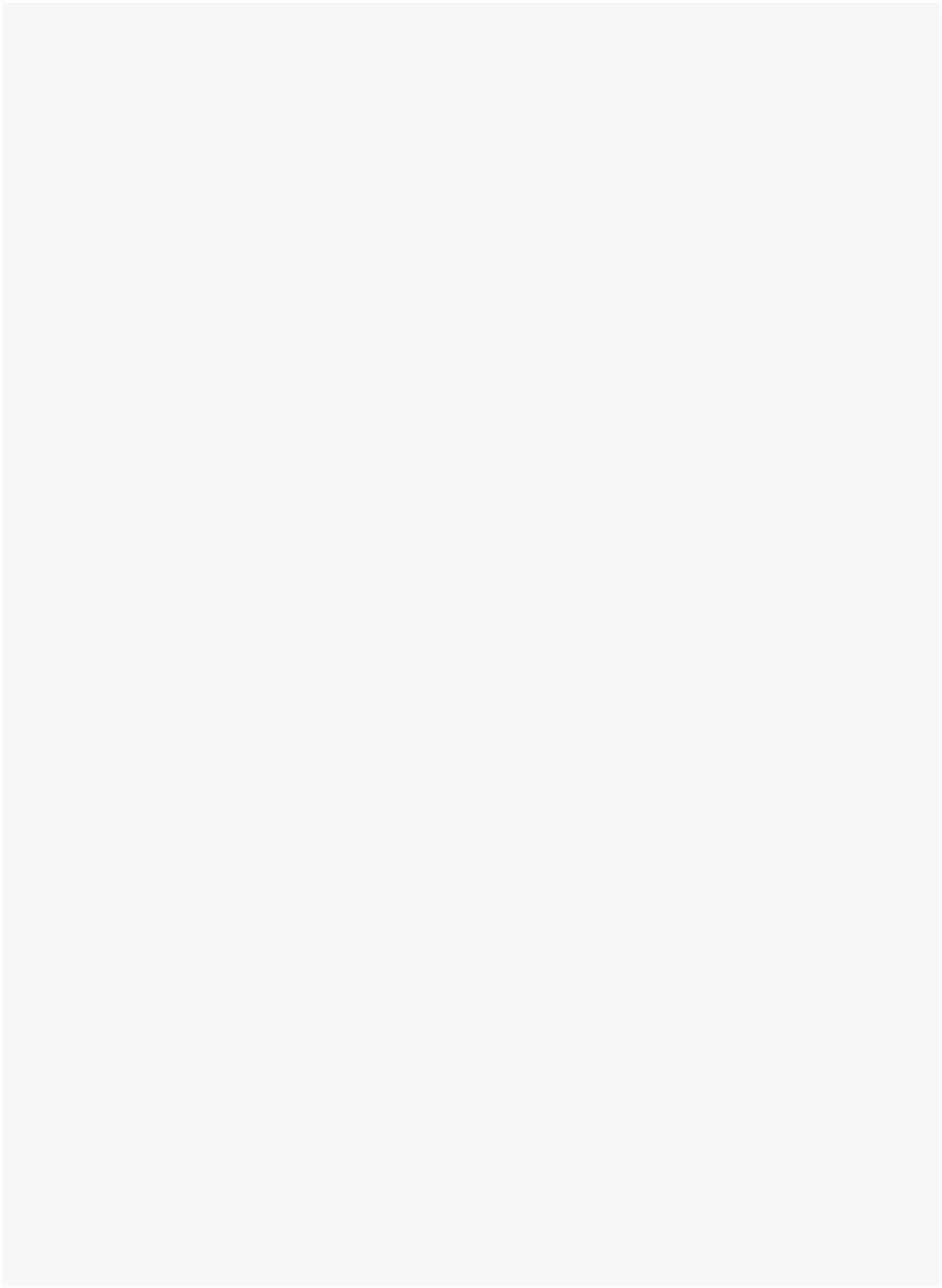
predds <- data.frame(predict(SVM, TestSet), TestSet$Class)

colnames(predds) <- c("pred", "truth")

predds[, 1] <- ifelse(predds[, 1] == 'UP', 1, 0)

predds[, 2] <- ifelse(predds[, 2] == 'UP', 1, 0)

27 | P a g e



pred <- prediction(predds$pred, predds$truth)

perf = performance(pred, measure = "tpr", x.measure = "fpr")

auc.perf = performance(pred, measure = 'auc', col = "red")

#Calculating the AUC

rmse.perf = performance(pred, measure = 'rmse')

#Calculating the RMSE

RMSE <- paste('RMSE =', rmse.perf@y.values, sep = ' ')

AUC <- paste('AUC =', auc.perf@y.values, sep = ' ')

print(AUC)

print(RMSE)

plot(perf, col = 1:10)

abline(a = 0, b = 1, col = "red")

#Plotting ROC curve

print('--------------------------------------------------------------------

-----')

}, error = function(e) {

print(e)

})

5.2 DESIGN DOCUMENT AND FLOWCHART

5.2.1 Design Document

Methods used for indicators

RSI(): To calculate RSI

EMA (): To calculate EMA

EMAcross = Open price - EMA

WPR(): To calculate WPR

CCI(): To calculate CCI

CMO(): To calculate CMO

ADX(): To calculate ADX

ROC(): To calculate ROC

Method used for Decision Tree

rpart(): To prepare decision tree model

Method used for pruning Decision Tree

prune():

To prune the decision tree

Method used for predicting the output

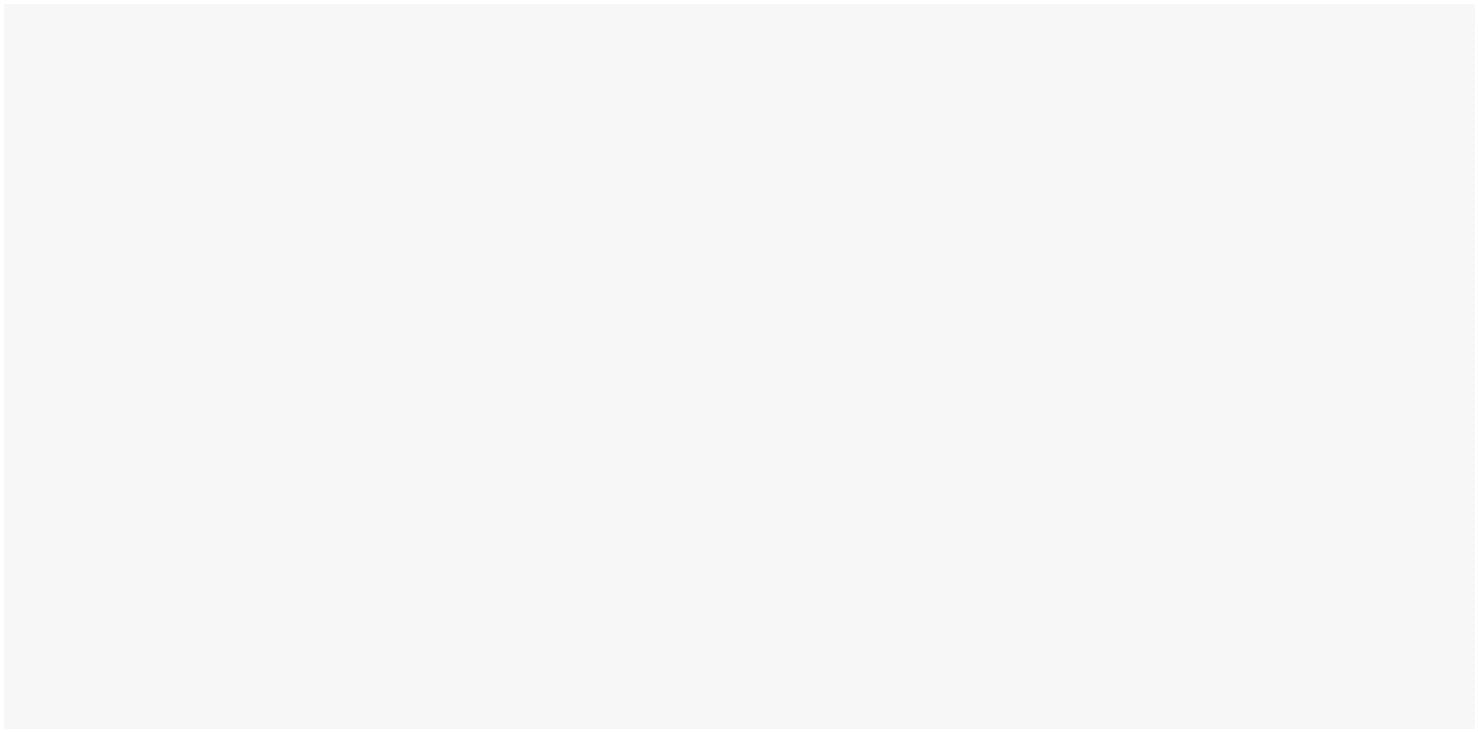
predict():

To predict the output

Methods used for evaluating the model

performance(): To calculate data for ROC graph, AUC, and RMSE

28 | P a g e



plot(): To plot ROC graph

Method used for SVM

svm(): To prepare decision tree model

Method used for predicting the output

predict():

To predict the output

Methods used for evaluating the model

performance(): To calculate data for ROC graph, AUC, and RMSE

plot(): To plot ROC graph

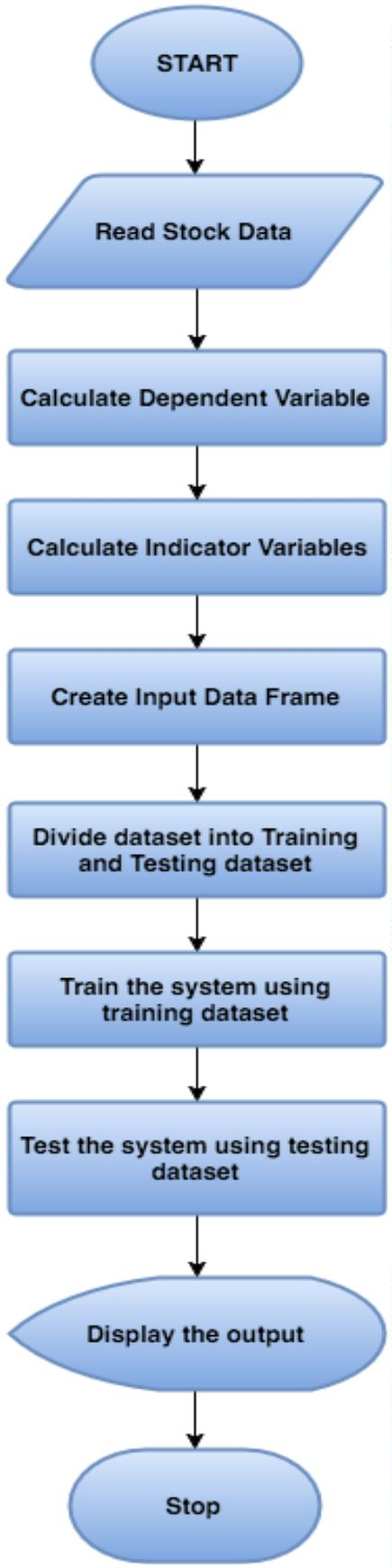
29 | P a g e



5.2.2 Flowchart

Figure 5: Program Flowchart

30 | P a g e



6 DATA ANALYSIS AND DISCUSSION

6.1 OUTPUT GENERATION

The application reads the input and applies the prediction algorithm to it to generate the

output. The output consists of confusion matrix, accuracy, ROC curve, area under curve and root

mean square error. This output is generated for all the selected stocks.

6.2 OUTPUT ANALYSIS

The close analysis of the output of Decision trees and SVM algorithm reveals that the SVM gives

better results than decision trees. Table 2 displays a comparison of the output of both the

algorithms.

Stock Name Parameters

Decision Tree

48.81

60.62

SVM

46.45

59.84

Apple

RSI

WPR

ADX

CMO

CCI

ROC

55.11

58.26

59.05

65.35

50.39

72.44

0.525

50.39

53.54

62.99

63.77

50.39

80.31

0.444

EMA Cross

Combined Accuracy

RMSE

Microsoft

RSI

WPR

ADX

CMO

CCI

ROC

57.48

57.48

51.96

60.62

61.41

69.29

47.24

81.10

0.435

49.60

60.62

49.60

57.48

64.56

66.92

49.60

82.67

0.416

EMA Cross

Combined Accuracy

RMSE

IBM

RSI

52.75

62.99

51.96

51.18

51.96

66.92

51.96

59.05

WPR

ADX

CMO

CCI

ROC

61.41

70.86

51.96

76.37

0.486

62.99

71.65

51.18

85.03

0.387

EMA Cross

Combined Accuracy

RMSE

31 | P a g e



General

Motors

RSI

53.54

53.54

WPR

ADX

CMO

CCI

58.26

49.60

62.99

59.05

70.07

48.81

74.01

0.509

66.14

48.81

61.41

66.14

70.07

48.81

82.67

0.416

ROC

EMA Cross

Combined Accuracy

RMSE

General

Electric

RSI

51.96

66.92

40.15

49.6

40.94

66.92

39.37

66.92

66.92

WPR

ADX

CMO

CCI

62.20

ROC

65.35

44.88

73.22

0.517

64.56

40.94

82.67

0.416

EMA Cross

Combined Accuracy

RMSE

Facebook

RSI

WPR

ADX

CMO

CCI

ROC

58.26

65.35

54.33

56.69

63.77

67.71

58.18

78.74

0.461

48.81

62.99

51.18

55.9

65.35

70.07

58.26

87.4

EMA Cross

Combined Accuracy

RMSE

0.355

Google

RSI

WPR

ADX

CMO

CCI

ROC

47.24

60.62

50.39

56.69

64.56

70.86

48.03

80.31

0.444

48.03

66.14

48.81

56.69

63.77

67.7

48.03

81.88

0.425

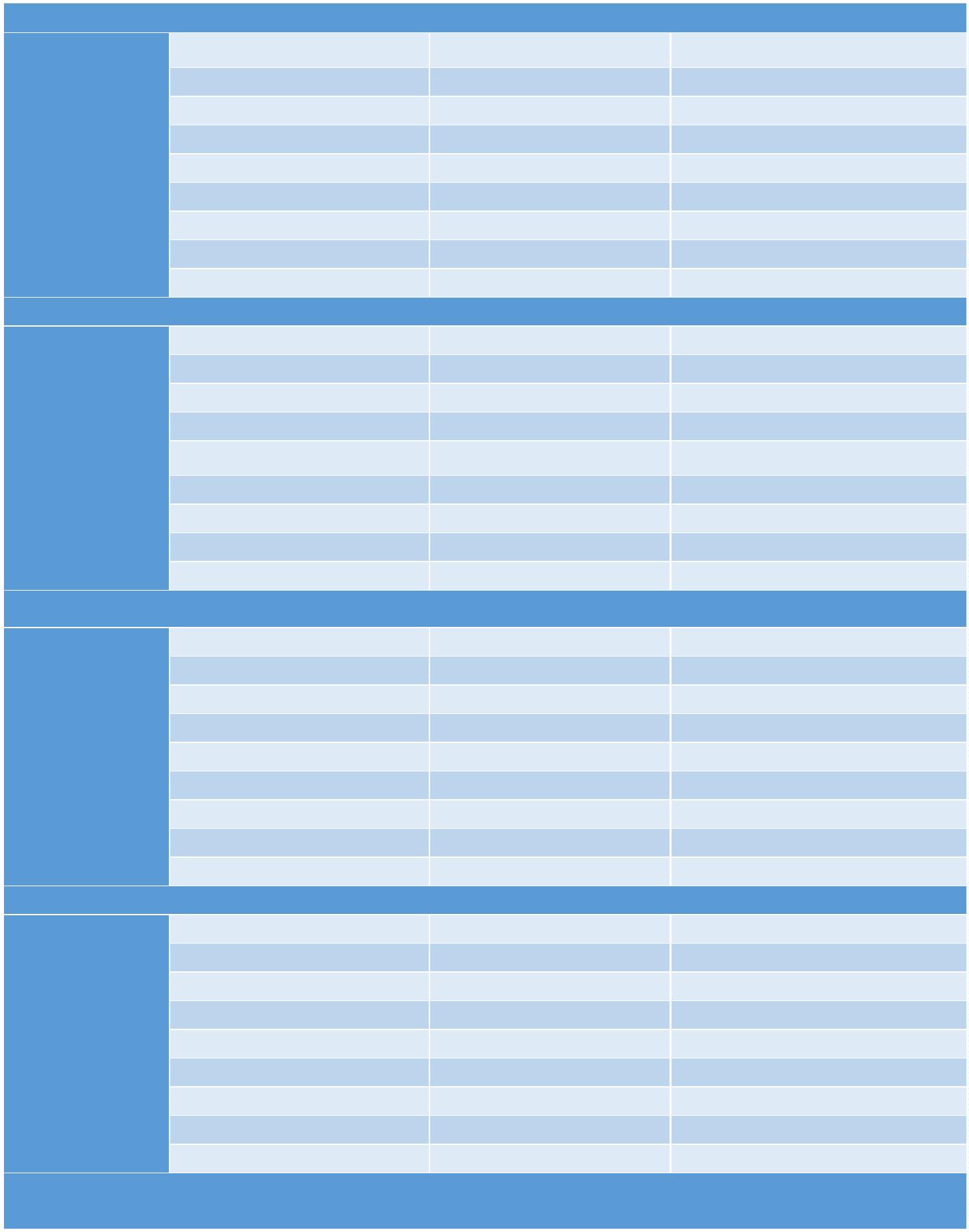
EMA Cross

Combined Accuracy

RMSE

Table 2: Effect of Indicators on prediction accuracy

32 | P a g e



6.3 COMPARE OUTPUT AGAINST HYPOTHESIS

The prediction accuracy depends upon the choice of indicator variables. We tried multiple

indicator variables and their permutations and selected the permutation which gave best

result. With the chosen combination of indicator variables, we were able to get the maximum

accuracy of 87.4% for Facebook stock.

6.4 STATISTIC REGRESSION

The independent variables used are RSI, EMA Crossover, CCI, ROC, CMO, WPR and ADX. The

effect of each of these independent variables on the accuracy of output can be seen in Table 2.

6.5 DISCUSSION

We selected stocks of 7 companies to train and test the system. Two years of data is

downloaded from Yahoo Finance, of which 75% is used to train the system and the remaining

25% is used for testing.

Many indicator functions and their permutations were tested while training and testing the

system. Of all the indicator functions tested, the ones which gave the best prediction result

were selected. The system performs very well for the prediction of the selected stocks.

33 | P a g e



7 CONCLUSION AND RECOMMENDATIONS

7.1 SUMMARY AND CONCLUSION

34 | P a g e

Stock markets are hard to monitor and require plenty of

context when trying to interpret the movement and predict

prices. In ANN, each hidden node is simply a node with a

single activation function, while in LSTM, each node is a

memory cell that can store contextual information. As such,

LSTMs perform better as they are able to keep track of the

context-specific temporal dependencies between stock

prices for a longer period of time while performing

predictions.

At its core, the stock market is a reflection of human

emotions. Pure number crunching and analysis have their

limitations; a possible extension of this stock prediction

system would be to augment it with a news feed analysis

from social media platforms such as Twitter, where

emotions are gauged from the articles. This sentiment

analysis can be linked with the LSTM to better train weights

and further improve accuracy.



8 BIBLIOGRAPHY

[1] A. Beattie, "7 Controversial Investing Theories," [Online]. Available:

http://www.investopedia.com/articles/financial-theory/controversial-financial-theories.asp.

[2] L. QianYu and F. ShaoRong, "Stock market forecasting research based on Neural Network and Pattern

Matching," 2010. [Online]. Available:

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5592759.

[3] Wikipedia, "List of S&P 500 Companies," [Online]. Available:

https://en.wikipedia.org/wiki/List\_of\_S%26P\_500\_companies.

[4] Wikipedia, "R," [Online]. Available: https://en.wikipedia.org/wiki/R\_(programming\_language).

[5] Wikipedia, "RStudio," [Online]. Available: https://en.wikipedia.org/wiki/RStudio.

[6] Wikipedia, "Eclipse," [Online]. Available: https://en.wikipedia.org/wiki/Eclipse\_(software).

[7] Wikipedia, "Java," [Online]. Available: https://en.wikipedia.org/wiki/Java\_(programming\_language).

[8] SVMS, "Support Vector Machines vs Artificial Neural Networks," [Online]. Available:

http://www.svms.org/anns.html.

35 | P a g e

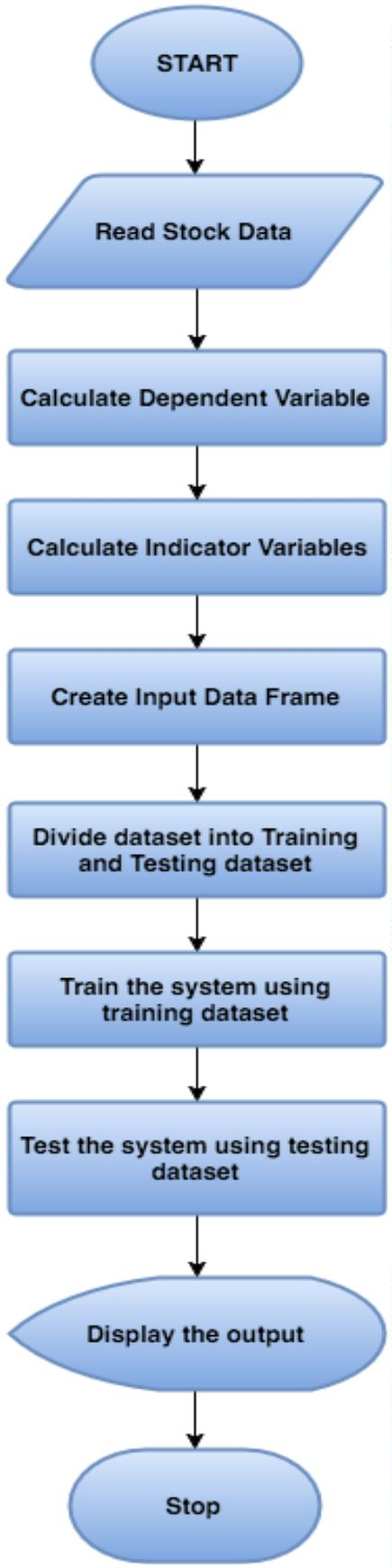


9 APPENDICES

9.1 PROGRAM FLOWCHART

Figure 6: Program Flow

36 | P a g e



9.2 PROGRAM SOURCE CODE AND DOCUMENTATION

9.2.1 Decision Tree Implementation Code

library(quantmod)

library(lubridate)

library(e1071)

library(rpart)

library(rpart.plot)

library(ROCR)

options(warn = -1)

a <- c('AAPL', 'FB', 'GE', 'GOOG', 'GM', 'IBM', 'MSFT')

for (i in 1:length(a))

{

SYM <- a[i]

print('--------------------------------------------------------------------

-----')

print(paste('Predicting the output for', SYM, sep = ' '))

trainPerc <- 0.75

date <- as.Date(Sys.Date() - 1)

endDate <- date#as.Date("2016-01-01")

d <- as.POSIXlt(endDate)

d$year <- d$year - 2

startDate <- as.Date(d)

STOCK <- getSymbols(

SYM,

env = NULL,

src = "yahoo",

from = startDate,

to = endDate

)

RSI3 <- RSI(Op(STOCK), n = 3)

#Calculate a 3-period relative strength index (RSI) off the open price

EMA5 <- EMA(Op(STOCK), n = 5)

#Calculate a 5-period exponential moving average (EMA)

EMAcross <- Op(STOCK) - EMA5

#Let us explore the difference between the open price and our 5-period EMA

MACD <- MACD(Op(STOCK),

fast = 12,

slow = 26,

signal = 9)

#Calculate a MACD with standard parameters

MACDsignal <- MACD[, 2]

#Grab just the signal line to use as our indicator.

SMI <- SMI(

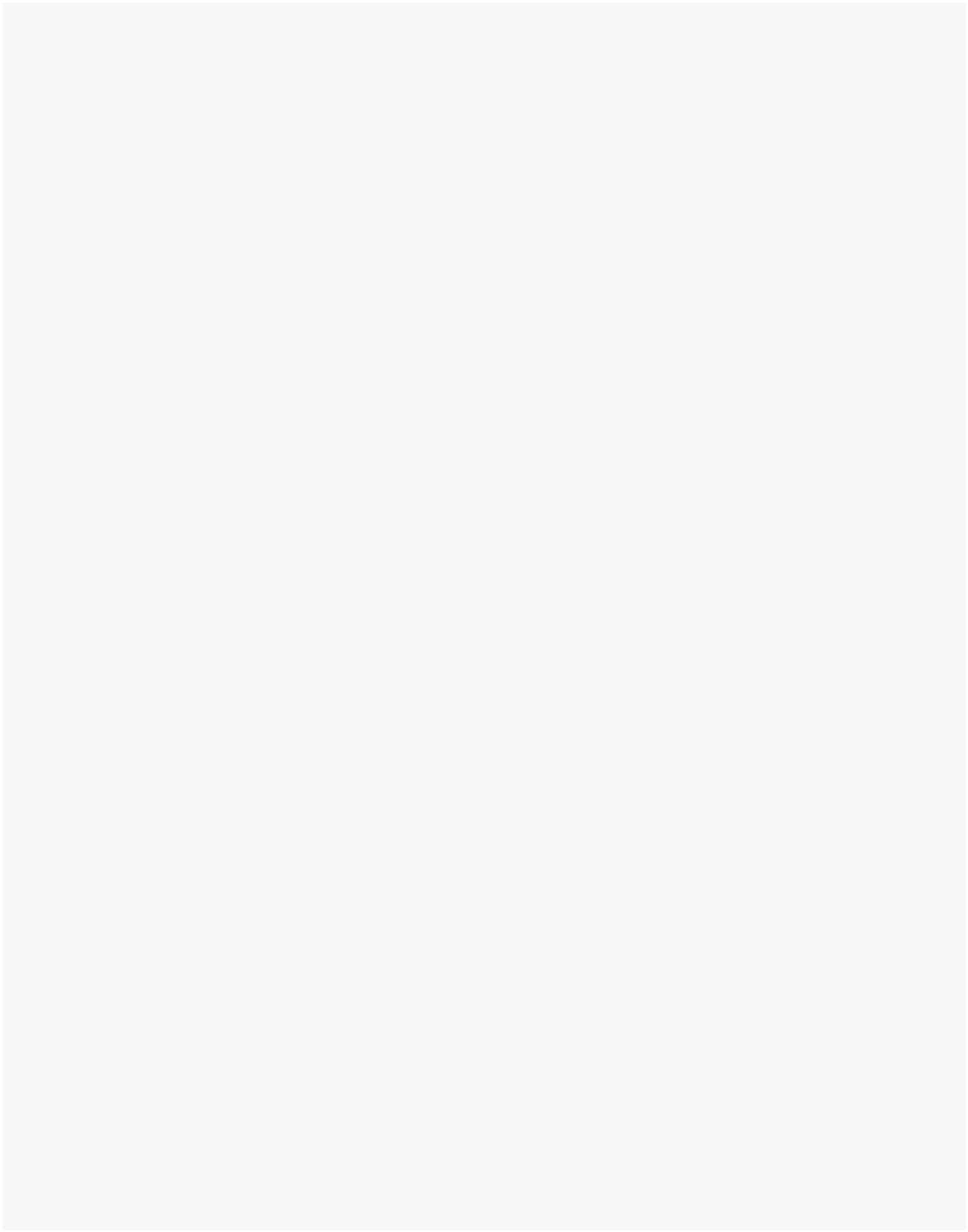
Op(STOCK),

n = 13,

slow = 25,

fast = 2,

37 | P a g e



signal = 9

)

#Stochastic Oscillator with standard parameters

SMI <- SMI[, 1]

#Grab just the oscillator to use as our indicator

WPR <- WPR(Cl(STOCK), n = 14)

WPR <- WPR[, 1]

ADX <- ADX(STOCK, n = 14)

ADX <- ADX[, 1]

CCI <- CCI(Cl(STOCK), n = 14)

CCI <- CCI[, 1]

CMO <- CMO(Cl(STOCK), n = 14)

CMO <- CMO[, 1]

ROC <- ROC(Cl(STOCK), n = 2)

ROC <- ROC[, 1]

PriceChange <- Cl(STOCK) - Op(STOCK)

#Calculate the difference between the close price and open price

Class <- ifelse(PriceChange > 0, "UP", "DOWN")

#Create a binary classification variable, the variable we are trying to pre

dict.

DataSet <-

data.frame(Class, RSI3, EMAcross, MACDsignal, SMI, WPR, ADX, CCI, CMO, RO

C)

#Create our data set

colnames(DataSet) <-

c(

"Class",

"RSI3",

"EMAcross",

"MACDsignal",

"Stochastic",

"WPR",

"ADX",

"CCI",

"CMO",

"ROC"

)

#Name the columns

#DataSet <- DataSet[-c(1:33),]#33

#Get rid of the data where the indicators are being calculated

TrainingSet <- DataSet[1:floor(nrow(DataSet) \* trainPerc), ]

#Use 2/3 of the data to build the tree

TestSet <-

DataSet[(floor(nrow(DataSet) \* trainPerc) + 1):nrow(DataSet), ]

#And leave out 1/3 data to test our strategy

DecisionTree <-

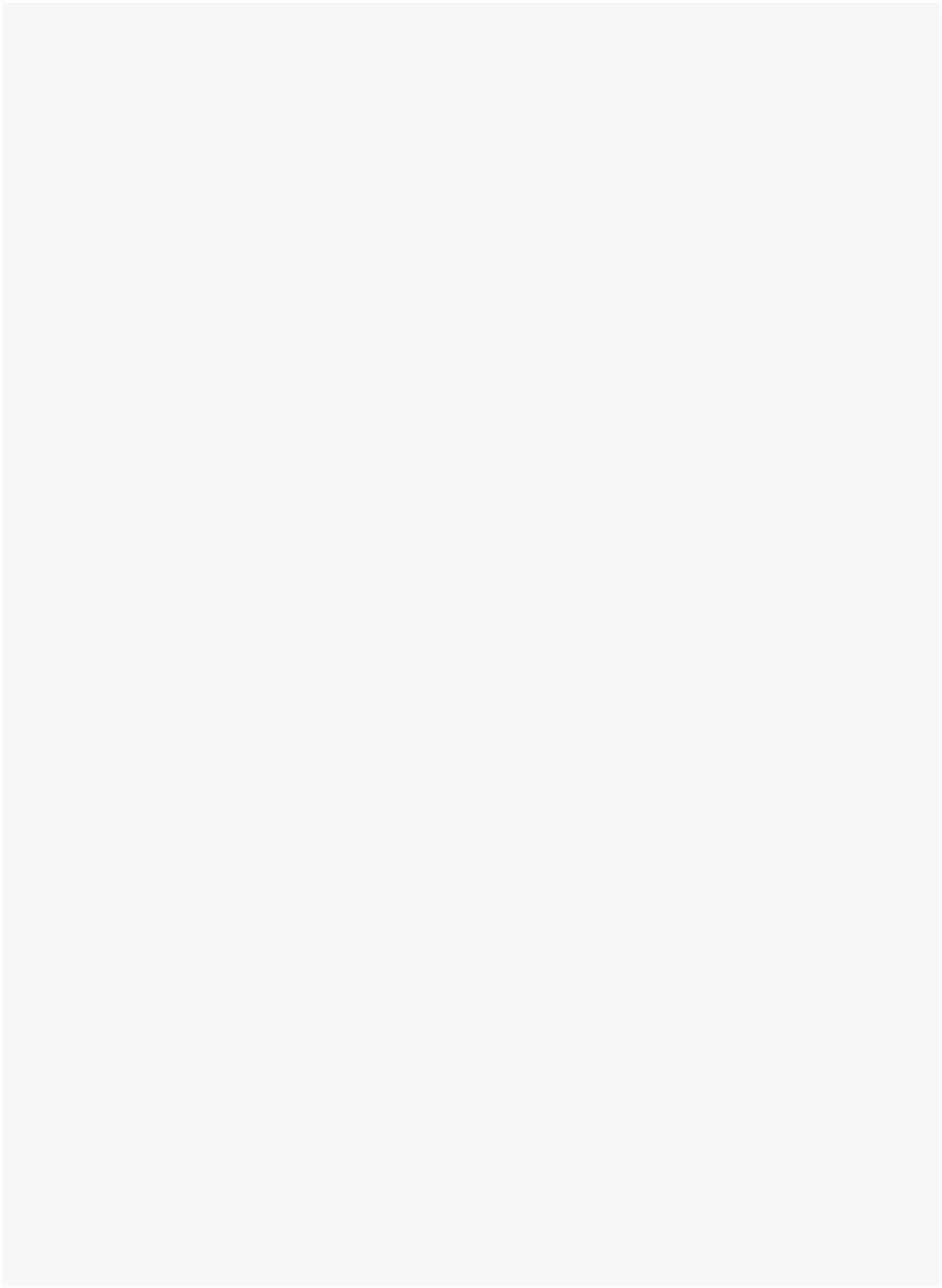
rpart(

Class ~ RSI3 + EMAcross + WPR + ADX + CMO + CCI + ROC,

#+ MACDsignal + Stochastic,

data = TrainingSet,

38 | P a g e



na.action = na.omit,

cp = .001

)

#Specifying the indicators to we want to use to predict the class and contr

olling the growth of the tree by setting the minimum amount of information ga

ined (cp) needed to justify a split.

prp(DecisionTree, type = 2, extra = 8)

#Nice plotting tool with a couple parameters to make it look good. If you w

ant to play around with the visualization yourself, here is a great resource.

fit <- printcp(DecisionTree)

#shows the minimal cp for each trees of each size.

mincp <- fit[which.min(fit[, 'xerror']), 'CP']

#Get the lowest cross-validated error (xerror)

plotcp(DecisionTree, upper = "splits")

#plots the average geometric mean for trees of each size.

PrunedDecisionTree <- prune(DecisionTree, cp = mincp)

#We are selecting the complexity parameter (cp) that has the lowest cross-v

alidated error (xerror)

t <- prp(PrunedDecisionTree, type = 2, extra = 8)

confmat <-

table(

predict(PrunedDecisionTree, TestSet, type = "class"),

TestSet[, 1],

dnn = list('predicted', 'actual')

)

print(confmat)

tryCatch({

acc <-

(confmat[1, "DOWN"] + confmat[2, "UP"]) \* 100 / (confmat[2, "DOWN"] + c

onfmat[1, "UP"] + confmat[1, "DOWN"] + confmat[2, "UP"])

#if (acc > 60) {

xy <-

paste('Decision Tree : Considering the output for', SYM, sep = ' ')

yz <-

paste('Accuracy',

acc,

sep = ' ')

out <- paste(xy, yz, sep = '\n')

print(out)

write(out,

file = "out",

append = TRUE,

sep = "\n\n")

#}

}, error = function(e) {

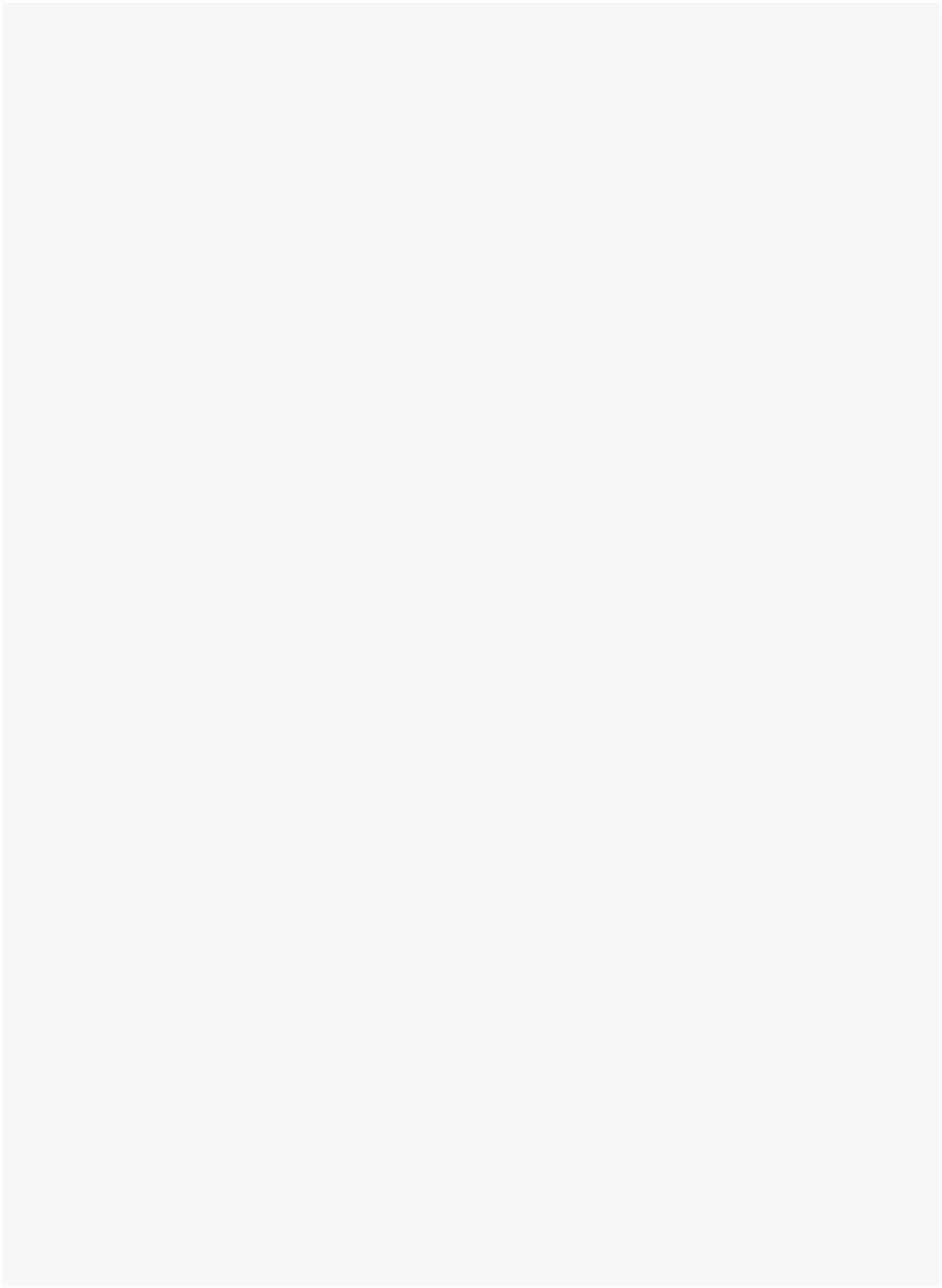
})

predout <- data.frame(predict(PrunedDecisionTree, TestSet))

predval <- predout['UP'] - predout['DOWN']

predclass <- ifelse(predout['UP'] >= predout['DOWN'], 1, 0)

39 | P a g e



predds <- data.frame(predclass, TestSet$Class)

colnames(predds) <- c("pred", "truth")

predds[, 2] <- ifelse(predds[, 2] == 'UP', 1, 0)

#print(predds)

pred <- prediction(predds$pred, predds$truth)

perf = performance(pred, measure = "tpr", x.measure = "fpr")

plot(perf, col = 1:10)

auc.perf = performance(pred, measure = 'auc')

rmse.perf = performance(pred, measure = 'rmse')

print(paste('RMSE =', rmse.perf@y.values), sep = ' ')

print(paste('AUC =', auc.perf@y.values), sep = ' ')

abline(a = 0, b = 1, col = "red")

print('--------------------------------------------------------------------

-----')

}

9.2.2 SVM Implementation Code

library(quantmod)

library(lubridate)

library(e1071)

library(rpart)

library(rpart.plot)

library(ROCR)

options(warn = -1)

a <- c('AAPL', 'FB', 'GE', 'GOOG', 'GM', 'IBM', 'MSFT')

for (i in 1:length(a))

{

SYM <- a[i]

print('--------------------------------------------------------------------

-----')

print(paste('Predicting the output for', SYM, sep = ' '))

trainPerc <- 0.75

date <- as.Date(Sys.Date() - 1)

endDate <- date#as.Date("2016-01-01")

d <- as.POSIXlt(endDate)

d$year <- d$year - 2

startDate <- as.Date(d)

STOCK <- getSymbols(

SYM,

env = NULL,

src = "yahoo",

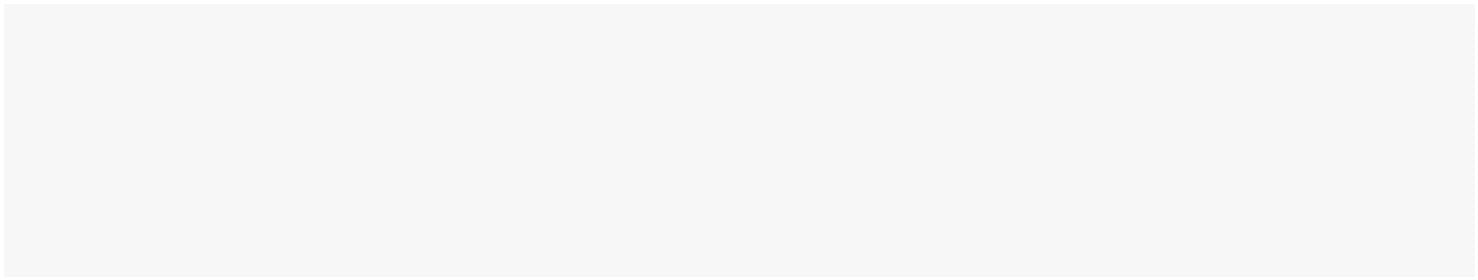
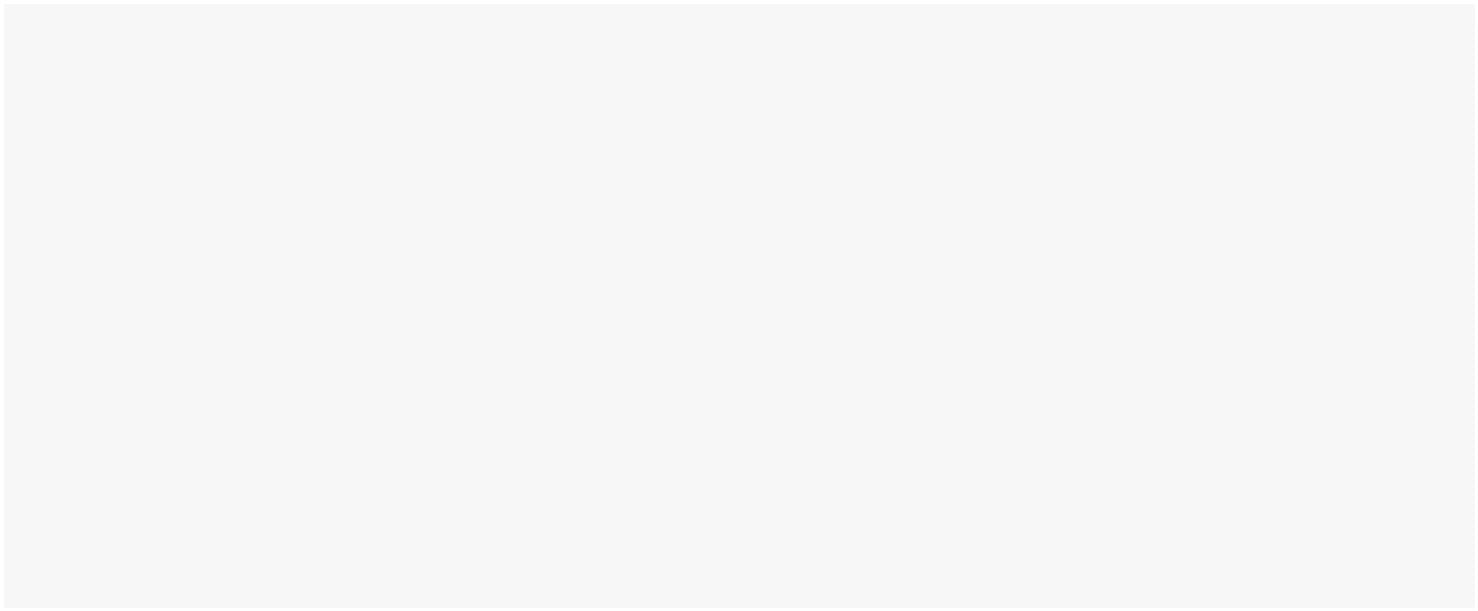
from = startDate,

to = endDate

)

RSI3 <- RSI(Op(STOCK), n = 3)

40 | P a g e



#Calculate a 3-period relative strength index (RSI) off the open price

EMA5 <- EMA(Op(STOCK), n = 5)

#Calculate a 5-period exponential moving average (EMA)

EMAcross <- Op(STOCK) - EMA5

#Let us explore the difference between the open price and our 5-period EMA

MACD <- MACD(Op(STOCK),

fast = 12,

slow = 26,

signal = 9)

#Calculate a MACD with standard parameters

MACDsignal <- MACD[, 2]

#Grab just the signal line to use as our indicator.

SMI <- SMI(

Op(STOCK),

n = 13,

slow = 25,

fast = 2,

signal = 9

)

#Stochastic Oscillator with standard parameters

SMI <- SMI[, 1]

#Grab just the oscillator to use as our indicator

WPR <- WPR(Cl(STOCK), n = 14)

WPR <- WPR[, 1]

ADX <- ADX(STOCK, n = 14)

ADX <- ADX[, 1]

CCI <- CCI(Cl(STOCK), n = 14)

CCI <- CCI[, 1]

CMO <- CMO(Cl(STOCK), n = 14)

CMO <- CMO[, 1]

ROC <- ROC(Cl(STOCK), n=2)

ROC <- ROC[, 1]

#DPO <- DPO(Cl(STOCK), n = 10)

#DPO <- DPO[, 1]

PriceChange <- Cl(STOCK) - Op(STOCK)

#Calculate the difference between the close price and open price

Class <- ifelse(PriceChange > 0, 'UP', 'DOWN')

#Create a binary classification variable, the variable we are trying to

predict.

DataSet <-

data.frame(Class, RSI3, EMAcross, MACDsignal, SMI, WPR, ADX, CCI, CMO,

ROC)

#Create our data set

colnames(DataSet) <-

c(

"Class",

41 | P a g e



"RSI3",

"EMAcross",

"MACDsignal",

"Stochastic",

"WPR",

"ADX",

"CCI",

"CMO",

"ROC"

)

#Name the columns

#DataSet <- DataSet[-c(1:33), ]

#Get rid of the data where the indicators are being calculated

TrainingSet <- DataSet[1:floor(nrow(DataSet) \* trainPerc),]

#Use 2/3 of the data to build the tree

TestSet <-

DataSet[(floor(nrow(DataSet) \* trainPerc) + 1):nrow(DataSet),]

#And leave out 1/3 data to test our strategy

SVM <-

svm(

Class ~ RSI3 + EMAcross + WPR + ADX + CMO + CCI + ROC,

# + MACDsignal + Stochastic,

data = TrainingSet,

kernel = "radial",

type = "C-classification",

na.action = na.omit,

cost = 1,

gamma = 1 / 5

)

print(SVM)

confmat <-

table(predict(SVM, TestSet, type = "class"),

TestSet[, 1],

dnn = list('predicted', 'actual'))

print(confmat)

tryCatch({

acc <-

(confmat[1, "DOWN"] + confmat[2, "UP"]) \* 100 / (confmat[2, "DOWN"] +

confmat[1, "UP"] + confmat[1, "DOWN"] + confmat[2, "UP"])

#if (acc > 60) {

xy <- paste('SVM : Considering the output for', SYM, sep = ' ')

yz <-

paste('Accuracy =',

acc,

sep = ' ')

out <- paste(xy, yz, sep = '\n')

42 | P a g e



print(out)

write(out,

file = "out",

append = TRUE,

sep = "\n\n")

#}

}, error = function(e) {

})

predds <- data.frame(predict(SVM, TestSet), TestSet$Class)

colnames(predds) <- c("pred", "truth")

predds[, 1] <- ifelse(predds[, 1] == 'UP', 1, 0)

predds[, 2] <- ifelse(predds[, 2] == 'UP', 1, 0)

pred <- prediction(predds$pred, predds$truth)

perf = performance(pred, measure = "tpr", x.measure = "fpr")

auc.perf = performance(pred, measure = 'auc', col = "red")

rmse.perf = performance(pred, measure = 'rmse')

print(paste('RMSE =', rmse.perf@y.values), sep = ' ')

print(paste('AUC =', auc.perf@y.values), sep = ' ')

plot(perf, col = 1:10)

abline(a = 0, b = 1, col = "red")

print('--------------------------------------------------------------------

-----')

}

9.3 INPUT/OUTPUT LISTING

Input

The input is two years of historic data along with the indicator variables for all the stocks and both the

algorithms. A sample of input is shown in Table 3.

## Date

Class

RSI

EMAcross

MACD

SMI

## 2014-05-05 UP

## 2014-05-06 DOWN

## 2014-05-07 DOWN

## 2014-05-08 DOWN

## 2014-05-09 UP

## 2014-05-12 UP

## 2014-05-13 UP

## 2014-05-14 UP

71.420701 3.194429e+00

88.852547 9.902935e+00

58.688940 2.235288e+00

38.005966 -3.176475e+00

29.688201 -4.590977e+00

44.242592 -1.094002e+00

62.190599 2.277339e+00

63.854668 1.804908e+00

1.181512792 68.2833293

1.522929335 72.7037309

1.806807870 74.7893073

2.015919827 74.4003458

2.152247208 72.5946848

2.238940024 70.1038033

2.297133579 68.4194937

2.330191625 66.8260825

Output

Technically the Output of the system should be the direction of stock for the next day. But for the better

understanding of the prediction model we are displaying the confusion matrix, root mean squared error

and area under curve.

43 | P a g e

