

**DAY TO DAY STOCK MARKET PREDICTION USING MACHINE LEARNING ALGORITHM AND DATA MINING TECHNIQUES**

**PROJECT REPORT**

Submitted by

**HEMAPRIYADHARSHINI. R [REGISTER NO. : 17TD1512]**

**KAVITHA. I [REGISTER NO. : 17TD1515]**

**MIRA. M [REGISTER NO. : 17TD1521]**

**RUXANA PARVEEN. F [REGISTER NO. : 17TD1531]**

**Under the guidance of**

**Prof. A . KANNAKI**

**(AP&HEAD/CSE)**

*Submitted in partial fulfillment for the award of the degree of*

**BACHELOR OF TECHNOLOGY**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**ACHARIYA COLLEGE OF ENGINEERING TECNOLOGY**

**PONDICHERRY UNIVERSITY**

**PUDUCHERRY, INDIA**

**AUGUST– 2021**

**ACHARIYA COLLEGE OF ENGINEERING TECHNOLOGY**

**PONDICHERRY UNIVERSITY**

(Approved by AICTE & Affiliated to Pondicherry University)

An ISO 9001:2008 Certified Institution

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**BONAFIDE CERTIFICATE**

This is to certify that the project work entitled **“DAY TO DAY STOCK MARKET PREDICTION USING MACHINE LEARNING ALGORITHM AND DATA MINING TECHNIQUES”** is a bonafide work done by **R. HEMAPRIYADHARSHINI [Reg no: 17TD1512], I. KAVITHA [Reg no: 17TD1515], M. MIRA [Reg no: 17TD1521], F. RUXANA PARVEEN [Reg no: 17TD1531]** in partial fulfillment of the requirement for the award of B.Tech. Degree in Computer Science And Engineering by Pondicherry University during the academic year 2020-2021.

**PROJECT GUIDE**  **HEAD OF THE DEPARTMENT**

Viva-Voce Examination held on \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**ACKNOWLEDGEMENT**

We humbly submit all the glory and thanks to our almighty for showering the blessings upon us and giving us the necessary wisdom for accomplishing this project.

We take immense pleasure in thanking our Chief Mentor **TR.J. ARAWINDAN** of Achariya Educational Trust, for providing us with an environment to complete our project successfully.

We wish to express our heartfelt regards and sincere thanks to our beloved Principal **Dr. S. GURULINGAM,** for his constant encouragement during the course of this project work.

We extend our gratitude to **Prof. A. KANNAKI @ VASANTHA AZHAGU**, Head of the Department for having permitted us to carry out this project work.

Words are inadequate in offering our thanks to our guide **Mrs. A. KANNAKI @ VASANTHA AZHAGU**, Assistant Professor/Head CSE, Department of Computer Science And Engineering for guiding us, all along the project with valuable suggestion, innovative ideas and constant support during every stage and development of our project.

We also thank our Teaching and Non-Teaching staff members for their valuable guidance and suggestions, which helped us in completing project in time.

Finally, yet important, we would like to thank our beloved parents, friends, teaching and non teaching staff who have directly and indirectly contributed to the success of this project.

**DECLARATION**

We affirm that the project work titled **“Day to day Stock market prediction using Machine Learning and Data Mining Techniques”** being submitted in partial fulfillment for the award of Bachelor in **Computer Science and Engineering** is the original work carried out by us. It has not formed the part of any other project work submitted for award of any degree of diploma.

**Signature of the candidates**

**R. HEMAPRIYADHARSHINI [REG NO:17TD1512]**

**I. KAVITHA [REG NO:17TD1515]**

**M. MIRA [REG NO: 17TD1521]**

**F. RUXANA PARVEEN [REG NO: 17TD1531]**

I certify that the declaration made above by the candidate is true.

**Signature of the Guide**

**Mrs. A. KANNAKI**

Assistant Professor/ Head CSE

Department of Computer Science and Engineering

Achariya College of Engineering Technology

Villianur

**iii**

**ABSTRACT**

Machine Learning can play a key role in a wide range of critical applications. In machine learning, Linear Regression (LR) and Support Vector Machines (SVMs) are widely used to predict stock market. But SVM have advanced features such as high accuracy and predictability. The nature of stock market movement has always been ambiguous for investors because of various influential factors. The prediction of a stock market direction may serve as and 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. 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. We use Support Vector Machine Algorithm (SVM) for predict accurate value of day to day stock market.

**iv**

**LIST OF FIGURES**

FIG.NO FIGURE NAME PAGE.NO

1. Applications of ML

2. Applications of SVM

**.**

**v**

**LIST OF ABBREVIATION**

**LR** Linear Regression

**SVM**  Simple Vector Machine

**ML** Machine Learning

**RSI**  Relative Strength Index

**EMA** Exponential Moving Average

**MACD** Moving Average Convergence Divergence

**SMI** Stochastic Momentum Index

**CCI** Commodity Channel Index

**CMO** Collateralized Mortgage Obligation

**ROC** Rate Of Change

**ADI**  Average Directional Index

**WPR** Williams Percentage (%) R

**RMSE** Root-Mean-Square Error

**ACC** Accuracy

**TPR** True Positive Rate

**FPR** False Positive Rate

**AUC** Area Under the Curve

**vi**

**TABLE OF CONTENTS**

**TITLE PAGE NO**

**ACKNOWLEDGEMENT ii**

**DECLARATION iii**

**ABSTRACT iv**

**LIST OF FIGURES v**

**LIST OF TABLES v**

**LIST OF ABBREVIATION vi**

**CHAPTER 1: INTRODUCTION**

* 1. Overview
     1. Working
     2. Application
     3. Advantages
     4. Disadvantages
  2. Methodology
  3. Support vector machine

**CHAPTER 2: RELATED WORK AND LITERATURE SURVEY**

2.1 Definition of the problem

2.2 Theoretical background of the problem

2.3 Related research to solve the problem

2.4 Advantage/disadvantage of those problem

2.5 Our solution to solve the problem

2.6 Why our solution is different from others?

2.7 Why our solution is better?

**CHAPTER 3: EXISTING SYSTEM**

3.1 Introduction

3.2 Architecture

3.3 Methodology

3.4 Drawback

**CHAPTER 4: PROPOSED SYSTEM**

4.1 Introduction

4.2 Objective

4.3 Support vector machine

4.4 Advantage

**CHAPTER 5: SYSTEM REQUIREMENTS**

5.1 Hardware and Software specification

5.1.1 Hardware specification

5.1.2 Software specification

5.2 Hardware components

5.2.1 Hard disk drive

5.2.2 RAM

5.2.3 SD card

5.2.4 USB

5.2.5 Adapter

5.2.6 SD card

5.3 Software requirements

5.3.1 Rstudio desktop

5.3.2 R

**CHAPTER 6: METHODOLOGY**

6.1 How to collect input data?

6.2 How to solve the problem

6.2.1 Details of Technical indicators

6.2.2 Algorithms design

6.2.3 Languages used

6.2.4 Tools used

6.3 How to generate output

6.4 How to prove correctness

**CHAPTER 7: DESIGN DOCUMENT AND FLOWCHART**

7.1 Design document methods used for indicators

7.2 Data flow diagram

7.3 Program source code and documentation

7.3.2 SVM implementation code

**CHAPTER 8: TESTING**

8.1 Testing objective

8.2 Unit Testing

8.3 Integration Testing

8.4 User Interface Testing

8.5 Performance Testing

8.6 System Testing

**CHAPTER 9 : IMPLEMENTATION OUTPUT**

**CHAPTER 10:CONCLUSION & REFERENCES**

Conclusion

References

**CHAPTER 1**

**INTRODUCTION**

* 1. **OVERVIEW :**

**Machine Learning (ML)** is a type of Artificial Intelligence(AI) that provides computer with the ability to learn without being explicitly programmed. Machine Learning is the ability to learn from the past experience i.e to learn from the past data some correlation the help for taking a decision or doing prediction latter.

* Machine Learning can play a key role in a wide range of critical applications. In machine learning, Linear Regression (LR) and Support Vector Machines (SVMs) are widely used to predict stock market. But SVM have advanced features such as high accuracy and predictability.
* **Machine learning** (**ML**) is the study of computer [algorithms](https://en.m.wikipedia.org/wiki/Algorithm) that improve automatically through experience and by the use of data. It is seen as a part of [artificial intelligence](https://en.m.wikipedia.org/wiki/Artificial_intelligence). Machine learning algorithms build a model based on sample data, known as "[training data](https://en.m.wikipedia.org/wiki/Training_data)", in order to make predictions or decisions without being explicitly programmed to do so.
  + 1. **WORKING :**

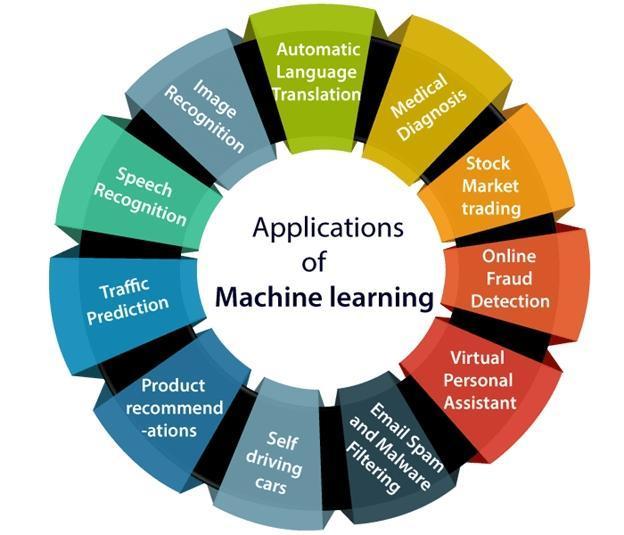
A subset of machine learning is closely related to [computational statistics](https://en.m.wikipedia.org/wiki/Computational_statistics), which focuses on making predictions using computers; but not all machine learning is statistical learning. The study of [mathematical optimization](https://en.m.wikipedia.org/wiki/Mathematical_optimization) delivers methods, theory and application domains to the field of machine learning. [Data mining](https://en.m.wikipedia.org/wiki/Data_mining) is a related field of study, focusing on [exploratory data analysis](https://en.m.wikipedia.org/wiki/Exploratory_data_analysis) through [unsupervised learning](https://en.m.wikipedia.org/wiki/Unsupervised_learning). In its application across business problems, machine learning is also referred to as [predictive analytics](https://en.m.wikipedia.org/wiki/Predictive_analytics).

Machine learning involves computers discovering how they can perform tasks without being explicitly programmed to do so. It involves computers learning from data provided so that they carry out certain tasks. For simple tasks assigned to computers, it is possible to program algorithms telling the machine how to execute all steps required to solve the problem at hand; on the computer's part, no learning is needed. For more advanced tasks, it can be challenging for a human to manually create the needed algorithms. In practice, it can turn out to be more effective to help the machine develop its own algorithm, rather than having human programmers specify every needed step

**1.1.2 APPLICATIONS OF ML :**

Machine learning focuses on the development of computer programs that can access data and use it to learn for themselves. Machine learning algorithms use historical data as input to predict new output values. Some of the applications of Machine learning (ML) are ,

* Image recognition
* Speech recognition
* Product recommendations
* Self-driving cars
* Stock market prediction
* Virtual Personal Assistant:
* Online Fraud Detection



*Fig 1.1 Applications of Machine Learning*

* 1. **METHODOLOGY**

In this project, we predict stock market by using Machine Learning and Data Mining techniques. we used the machine learning algorithm,

* Support Vector Machine(SVM)
  1. **SUPPORT VECTOR MACHINE (SVM)**

Support Vector machine is a machine learning technique used in recent studies to forecast stock prices. The model attempts to predict whether a stock price sometime in the future will be higher or lower than it is on a given day. Support Vector Machines are one of the best binary classifiers. They create a decision boundary such that most points in one category fall on one side of the boundary while most points in the other category fall on the other side of the boundary.

SVM offers very high accuracy compared to other classifiers such as logistic regression, and decision trees. It is known for its kernel trick to handle nonlinear input spaces. It is used in a variety of applications such as face detection, intrusion detection, classification of emails, news articles and web pages, classification of genes, and handwriting recognition.

SVM is an exciting algorithm and the concepts are relatively simple. Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems.

The support vector machine (SVM) is a predictive analysis data-classification algorithm that assigns new data elements to one of labeled categories. This is essentially the problem of image recognition or, more specifically, face recognition: You want the classifier to recognize the name of a person in a photo.

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. SVM algorithm can be used for Face detection, image classification, text categorization, etc.

The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane.

Support vector machine (SVM) is machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible. SVMs are used in text categorization, image classification, handwriting recognition and in the sciences. A support vector machine is also known as a support vector network (SVN).

**1.3.1 Applications of SVM :**

**Support Vector Machine (SVM)** is used in a variety of applications such as

* Face detection,
* Intrusion detection,
* Classification of emails, news articles and web pages,
* Classification of genes,
* Predicting stock market,
* Handwriting recognition,
* Image classification,
* Text categorization.

**CHAPTER 2**

**RELATED WORK AND LITERATURE SURVEY**

**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.

**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.

**2.5 OUR SOLUTION TO SOLVE THIS PROBLEM**

* We will implement the system using machine learning technique
* We will train both the systems using 75% of 2 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 Support Vector Machines with C type classification and Radial Basis Function(RBF) kernel. And we use Decision tree to predict the stock market. This project is useful for both Long and Short term shareholders.

**2.7 WHY OUR SOLUTION IS BETTER?**

It uses SVM and Decision Trees which have better performance than Neural Network. Moreover, using SVM will takes away the burden of matching the present price pattern with historic patterns and also SVM trains faster than a NN and has a lower computational cost.

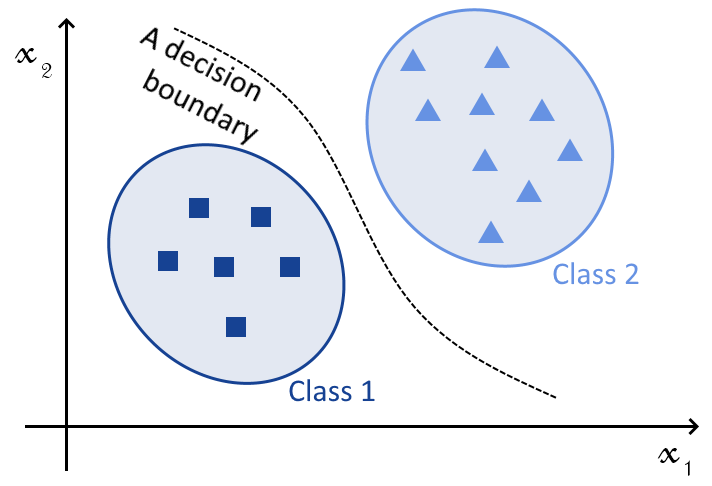
Also, other solution uses the financial data as it is without using any indicators, whereas our solution uses many indicators such as EMA, RSI, MACD, SMI, CCI, ROC, CMO, WPR and ADX to get better results.

**CHAPTER 3**

**EXISTING SYSTEM**

**3.1 INTRODUCTION**

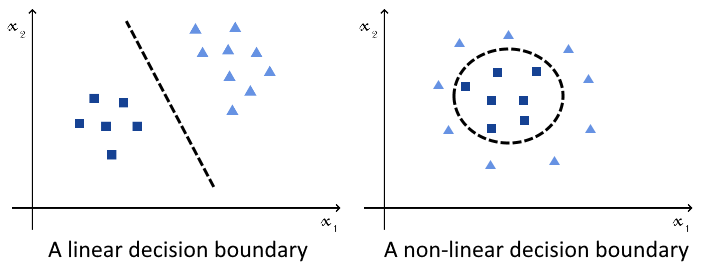
The problem of classification consists of the learning of a function of the form , where  is a feature vector and  is a vector corresponding to the classes associated with observations. This image represents classification in graphical form:



*Fig 3.1 Decision Boundary*

SVMs and NNs can both perform this task; with an appropriate choice of kernel, in the case of the SVM, or of activation function, in the case of NNs. The difference, therefore, isn’t in the types of tasks that they perform; but rather, in other characteristics of their theoretical bases and their implementation, as we’ll see shortly.

They both also can, which is equally important, approximate both linear and non-linear functions:



*Fig 3.2 Linear & Non-Linear Boundary*

This means that both algorithms can equally tackle all types of classification problems; hence, the decision to use one over the other doesn’t depend on the problem itself

 The [universal approximation theorem](https://www.baeldung.com/cs/neural-net-advantages-disadvantages#universal-approximation-theorem-and-its-limitation) tells us that a neural network with a single hidden layer and a non-linear activation can approximate, with an appropriate choice of weights, any continuous function.

**If the decision boundary of a classification problem can be defined as a continuous function,** which is always the case**, then it can also be defined as a continuous mapping of the feature space.** This, in turn, means that the universal approximation theorem guarantees that it can be approximated by a NN.

**3.2 ARCHITECTURE**

Neural networks have a different way of operating and, in particular, don’t require kernels. This, of course, with the exception of [convolutional neural networks](https://www.baeldung.com/cs/ai-convolutional-neural-networks).

A neural network for classification, in this context, correspond to a NN with a single hidden layer and a non-linear activation function. The most common types of non-linear activation functions for NNs for classification are:

* [**logistic function**](https://www.baeldung.com/cs/ml-nonlinear-activation-functions#1-logistic)**,**
* [**hyperbolic tangent**](https://www.baeldung.com/cs/ml-nonlinear-activation-functions#2-hyperbolic-tangent)**,**
* [**softmax**](https://www.baeldung.com/cs/ml-nonlinear-activation-functions#3-softmax)**,**

All these functions take as an input a linear combination of a feature vector and a weight vector . They then return an output that’s comprised in some finite interval.

As mentioned before, a neural network with a single hidden layer and a non-linear activation function can approximate any given continuous function. If the decision boundary isn’t continuous in the original feature space, the addition of further layers in the NN can increase its dimensionality, up to a point in which it is. A deep neural network can, therefore, approximate all decision boundaries comprised of multiple continuous regions.

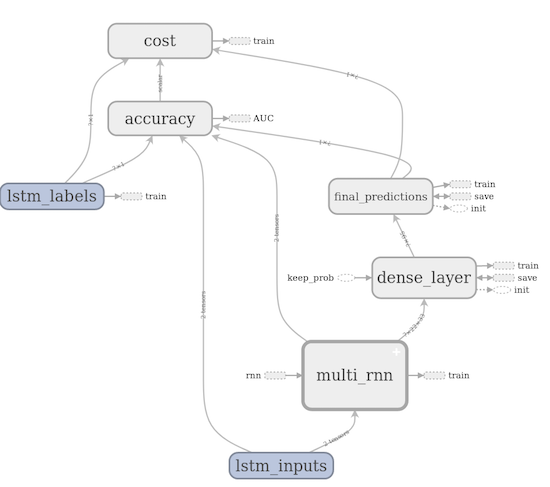
The problem, however, is that the universal approximation theorem includes no guarantee about the learnability of a decision function. This means that, for a given initial random configuration of the weights of the network, the NN may never learn the decision function through gradient descent. SVMs, on the other hand, always guarantee convergence.

**3.3 METHODOLOGY**

It’s a fairly simple network, where a multi-layer RNN (with GRU layers) is fed into a dense layer (which includes a dropout layer). The number of layers, activations, and dropout percentage all are optimized during training.  
The “Accuracy” node is long convoluted set of TF operations that convert a prediction from the dense network into a binary gradient movement. As an experiment, this accuracy is actually currently used in my cost function as:

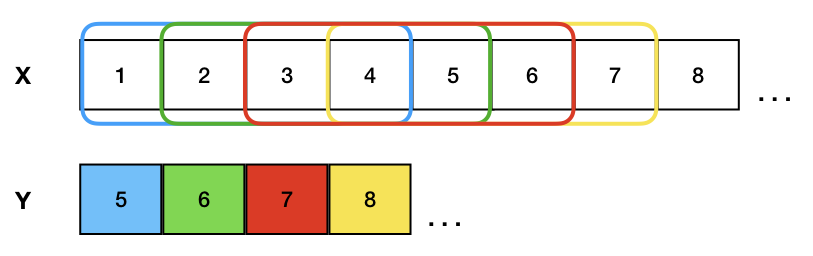
cost = (1-Accuracy) + tf.losses.mean\_squared\_error(labels, final\_predictions)

where the *label* is the normalized price, and *final\_predictions* are the normalized actual price predictions. I use the *AdamOptimiser* with a cyclic function learning rate. Is this the optimal/best way to do it? I’m not entirely sure yet!



*Fig 3.3 LSTM Method*

Then, we create the training, validation and testing datasets.  
Since this is a sequence prediction problem, we use a sliding window algorithm. The premise is shown in the figure below. X number of points (4 in the image) are used, with X+1 taken as the label and forming a new array. The window is then moved 1 point forward and the calculation repeated. This way you have a large array (X, which are your inputs) as well as your labels, Y.

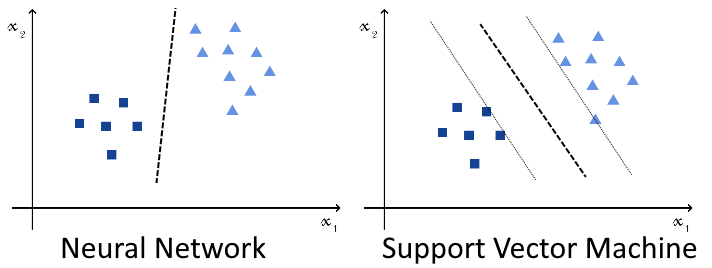


*Fig 3.4 Neural Network*

**3.4 DRAWBACK**

We’re building a system for medical imaging for the detection of microcalcifications in tissues. We have reason to believe that the samples we collected are quite distant from any decision boundary that might distinguish between positive and negative samples. In other words, we believe that the training samples only represent instances of the two classes where most of the components of the input support the class affiliation.

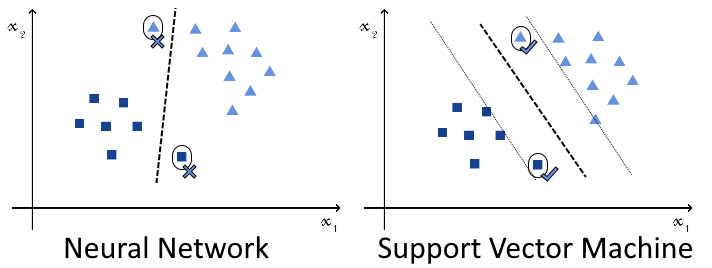
As a consequence, after its deployment in the real world, we can expect the machine learning system to risk classifying incorrectly all observations that are closer to the opposite class than has ever been noted in training:



*Fig 3.5 SVM vs NN*

This system is going to have a predictable impact on the health of the population. For this reason, we’re concerned that it may incorrectly classify edge cases, and want to minimize this possibility.

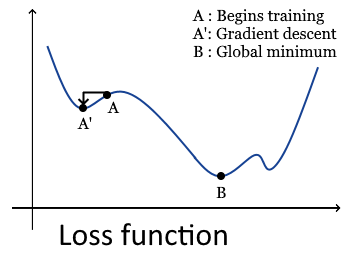
We know that a neural network would certainly learn some decision function and perform well under training, testing, and validation. The function learned, however, isn’t necessarily distant from the observed samples. On the contrary, it may be very close to them, and thus incorrectly classify many future observations against real-world data



*Fig 3.6 SVM vs NN(1)*

In this case, the usage of SVMs may be preferable. This is because, as we discussed above, the SVM learns that decision boundary which maximizes the distance against the closest observations that belong to opposite classes. This, in turn, should produce better performances against the edge cases that we’re going to encounter in the future.

The last difference concerns a consequence of the different usage of optimization techniques. Because NNs use gradient descent, this makes them sensitive to the initial randomization of its weight matrix. This is because, if the initial randomization places the neural network close to a local minimum of the optimization function, the accuracy will never increase past a certain threshold:



*Fig 3.7 Loss function*

SVMs are more reliable instead, and they guarantee convergence to a global minimum regardless of their initial configuration.

**CHAPTER 4**

**PROPOSED SYSTEM**

**4.1 INTRODUCTION**

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 study and improve the supervised learning algorithms to predict the stock price.

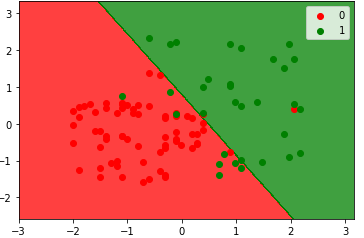
Investors are familiar with says, buy low and sell high but this does not provide enough context to make proper investment decisions. Before an investor invests in any stock, he needs to be aware how the stock market behaves. Investing in a good stock but at a bad time can have disastrous results, while 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

A support vector machine (SVM) is a supervised [machine learning](https://monkeylearn.com/machine-learning/) model that uses [classification algorithms](https://monkeylearn.com/blog/machine-learning-algorithms/) for two-group classification problems. After giving an SVM model sets of labeled training data for each category, they’re able to categorize new text.

Compared to newer algorithms like neural networks, they have two main advantages: higher speed and better performance with a limited number of samples (in the thousands). This makes the algorithm very suitable for text classification problems, where it’s common to have access to a dataset of at most a couple of thousands of tagged samples.

**4.2 OBJECTIVE**

As we discussed earlier SVM is an algorithm that is used to solve classification problems. The objective of SVM algorithm is to find an optimal **hyperplane**that **maximizes the margin between two classes**. Hyperplane is a generalized term to identify the decision boundary in an n-dimensional space.

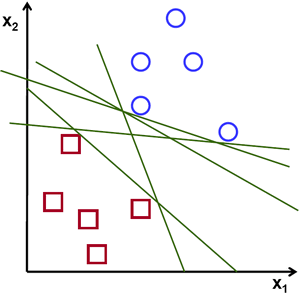
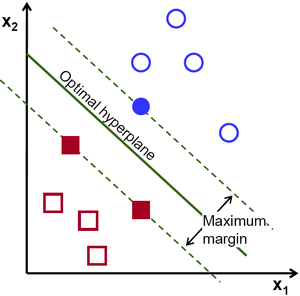


*Fig 4.1 SVM Decision Boundary*

The decision boundary is dividing the plane into 2 halves, thus the data points can fall into their respective classes. Now that we have seen what SVM looks like, let us look at the working of this beautiful algorithm.

**4.3 SUPPORT VECTOR MACHINE**

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

*Fig 4.2 Neural Network*  *Fig 4.3 Support Vector Machine*

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.

**4.4 ADVANTAGE**

[SVM](https://iq.opengenus.org/understand-support-vector-machine-in-depth/) is one of the supervised algorithms mostly used for classification problems.

* SVM is very helpful method if we don’t have much idea about the data. It can be used for the data such as image, text, audio etc.It can be **used for the data that is not regularly distributed and have unknown distribution.**
* The SVM provides a **very useful technique within it known as kernel** and by the application of associated kernel function we can solve any complex problem.  
  Kernel provides choosing a function which is not necessarily linear and can have different forms in terms of different data it operates on and thus is a non-parametric function.
* In Classification problems, there is a strong assumption that is Data have samples that are linearly separable but with the introduction of kernel, **Input data can be converted into High dimensional data** avoiding the need of this assumption.  
  K(x1, x2)=〈f(x1), f(x2)〉Where K is the kernel function, x1, x2 are n-dimensional inputs and f is a function that is used to map n-dimensional space into m-dimensional space and 〈x1, x2〉is used to specify/indicate the dot product
* SVM generally **do not suffer condition of overfitting** and performs well when there is a clear indication of separation between classes. SVM can be used when total no of samples is less than the no of dimensions and performs well in terms of memory.
* SVM Classifier in comparison to other classifiers have **better computational complexity** and even if the number of positive and negative examples are not same ,SVM can be used as it has the ability to normalize the data or to project into the  
  space of the decision boundary separating the two classes.
* The other reason to say that SVM is better than other algorithms is the reason  
  that it can also **perform in n-Dimensional space**.
* The **Execution time comes out to be very little** in comparison to algorithm such as Artificial Neuron Network.
* The other reason to say SVM better is the fact that after **doing little modification in the feature extracted data** does not affect the results which were expected before. It is converging very fast and as earlier stated in the article
* Kernel Functionality, In general Polynomial kernel proves out to be a better factor in terms of Support Vector Machine.
* In comparison with Naive Bayes Algorithm which is also a technique used for classification, Support Vector Machine Algorithm has a **faster prediction along with better accuracy.**
* In comparison with Logistic Regression which is also a classification method SVM proves itself to be cheaper , it has a **time complexity of O(N^2\*K)**where K is no  
  of support vectors whereas logistic Regression had the time complexity of O(N^3).
* SVMs can **be robust, even when the training sample has some bias** and one of the reasons that SVM proves out to be robust is its ability to deliver unique solution better than Neural Networks where we get more than one solutions corresponding to each local minima for different samples.
* The Other Important advantage is that SVM can be **applicable to also semi – supervised Learning Models.** It can be applicable to not only unlabeled data but also with the labeled data.
* SVM has a concept of **"Transductive SVM" ,** in such a concept only one thing is needed to satisfy that is Minimization problem and can be applied accordingly when  
  needed.
* We can also used **Inbuilt functionality of SVM,** It is available in languages such as Python and Matlab and SVM can also be used for non - linearly separable data having soft margin and Linearly separable data having hard margin.

**CHAPTER 5**

**SYSTEM REQUIREMENTS**

**5.1 HARDWARE AND SOFTWARE SPECIFICATION**

The minimum hardware required for the designer/developer and user are given below

**5.1.1 Hardware specification :**

The hardware used are as follows

* Processor - Minimum 2GHz
* Micro SD card
* Wi-Fi
* Hard Disk - Minimum 80GB
* RAM - 4 GB

**5.1.2 Software specification :**

The minimum software required are given below

* Environment - RStudio 2010
* OS - Windows Environment 7 and above
* Backend - R

**5.2 HARDWARE COMPONENTS**

**5.2.1 Hard disk drive**

A hard disk drive (HDD), hard disk, hard drive, or fixed disk is an elector-mechanical data storage device that uses magnetic storage to store and retrieve digital data using one or more rigid rapidly rotating platters coated with magnetic material. The platters are paired with magnetic heads, usually arranged on a moving actuator arm, which read and write data to the platter surfaces. Data is accessed in a random-access manner, meaning that individual blocks of data can be stored and retrieved in any order. HDD are a type of non-volatile storage, retaining stored data even when powered off.

**5.2.2 RAM**

Random Access Memory (RAM) is a form of computer memory that can be read and changed in any order, typically used to store working data and machine code. A random-access memory device allows data items to be read or written in almost the same amount of time irrespective of the physical location of data inside the memory. In contrast, with other direct-access data storage media such as hard disks, CD-RW, DVD-RW and the order magnetic tapes and drum memory, the time required to read and write data items varies significantly depending on their physical locations on the recording medium, due to mechanical limitations such as media rotations speeds and arm movement.

**5.2.3 USB**

Universal Serial Bus (USB) is an industry standard that establishes specifications for cable and connectors and protocols for connections, communications and power supply between computers, peripheral devices and other computers. Released in 1996, the USB standard is currently maintained by the USB Implementer Form (USB-IF). There have been four generations of USB specifications:

* USB 1.x
* USB 2.0
* USB 3.x and
* USB 4

**5.2.4** **Adapter**

An AC adapter, AC/DC adapter, or AC/DC converter is a type of external power supply, often enclosed in a case similar to an AC plug. Other common names include plug pack, plug-in adapter, adapter block, domestic mains adapter, line power adapter, wall wart, power brick, and power adapter. Adapters for battery-powered equipment may be described as chargers or re-chargers (see also battery charger). AC adapters are used with electrical devices that require power but do not contain internal components to drive the required voltage and power from mains power. The internal circuitry if an external power supply is vary similar to the design that would be used for a built-in or internal supply. External power supplies are used both with equipment with no other source of power and with battery-powered equipment, where the supply when plugged in, can sometimes charge the battery in addition to powering the equipment.

**5.2.5 SD card**

Secure Digital, officially abbreviated as SD, is a proprietary non-volatile memory card format developed by the SD Card Association (SDA) for use in portable devices. Cards can protect their contents from erasure or modification, prevent access by non-authorized users, and protect copyrighted content using digital rights management. When looking at the SD card from the top, the right side (the side with the beveled corner) must be notched. On the left side, there may be a write-protection notch. If the notch is omitted, the card can be read and written. If the card is notched, it is read only.The usage of SD card is portable devices, including digital cameras and hand held computers. It is extended from Multimedia Card.

**5.3 SOFTWARE COMPONENTS**

**5.3.1 Rstudio desktop**

RStudio is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.

RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux) or in a browser connected to RStudio Server or RStudio Workbench (Debian/Ubuntu, Red Hat/CentOS, and SUSE Linux).

**Overview :**

* Access RStudio locally
* Syntax highlighting, code completion, and smart indentation
* Execute R code directly from the source editor
* Quickly jump to function definitions
* View content changes in real-time with the Visual Markdown Editor
* Easily manage multiple working directories using projects

**5.3.2 R**

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. R is a language and environment for statistical computing and graphics. It is a GNU project which is similar to the S language and environment which was developed at Bell Laboratories (formerly AT&T, now Lucent Technologies) by John Chambers and colleagues. R can be considered as a different implementation of S. There are some important differences, but much code written for S runs unaltered under R.

R provides a wide variety of statistical (linear and nonlinear modelling, classical statistical tests, time-series analysis, classification, clustering) and graphical techniques, and is highly extensible. The S language is often the vehicle of choice for research in statistical methodology, and R provides an Open Source route to participation in that activity.

## The R environment :

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes

* an effective data handling and storage facility,
* a suite of operators for calculations on arrays, in particular matrices,
* a large, coherent, integrated collection of intermediate tools for data analysis,
* graphical facilities for data analysis and display either on-screen or on hardcopy, and
* a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

The term “environment” is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R, like S, is designed around a true computer language, and it allows users to add additional functionality by defining new functions. Much of the system is itself written in the R dialect of S, which makes it easy for users to follow the algorithmic choices made. For computationally-intensive tasks, C, C++ and Fortran code can be linked and called at run time. Advanced users can write C code to manipulate R objects directly.

Many users think of R as a statistics system. We prefer to think of it as an environment within which statistical techniques are implemented. R can be extended (easily) via *packages*. There are about eight packages supplied with the R distribution and many more are available through the CRAN family of Internet sites covering a very wide range of modern statistics.

R has its own LaTeX-like documentation format, which is used to supply comprehensive documentation, both on-line in a number of formats and in hardcopy.

**CHAPTER 6**

**METHODOLOGY**

**6.1 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.

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

3. System will take last 2 years’ stock data of the company using quantmod package in R.

4. Further we divide the data into two parts, training data and testing data, where 75% of the data will be used for training and 25% of the data will be used for testing.

*Figure 6.1 Steps to collect input data*

**6.2 HOW TO SOLVE THE PROBLEM?**

We will solve the problem using the following supervised machine learning technique to build our model.

* Support Vector Machine with Technical Indicators.

To solve the problem, we will follow below steps

1. Fetch the data of a stock from Yahoo Finance of last 2 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.

* + 1. **Details of Technical Indicators**

• **Relative Strength Index – RSI**

The relative strength index ( RSI) is a technical indicator used in the analysis of financial markets. It is intended to chart the current and historical strength or weakness of a stock or market based on the closing prices of a recent trading period.

The relative strength index (RSI) is a technical momentum indicator that compares the magnitude of recent gains to recent losses in an attempt to determine overbought and oversold conditions of an asset. It is calculated using the following formula:

RSI=100– 100 / (1+RS\*)

Where, RS=Average of x days' up closes/Average of x days' down closes.

• **Exponential Moving Average - EMA**

An exponential moving average (EMA) is a type of moving average that is similar to a simple moving average, except that more weight is given to the latest data. The exponential moving average is also known as "exponentially weighted moving average".

An exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA), is a first-order infinite impulse response filter that applies weighting factors which decrease exponentially. The weighting for each older datum decreases exponentially, never reaching zero.

• **Moving Average Convergence Divergence - MACD**

Moving average convergence divergence (MACD) is a trend-following momentum indicator that shows the relationship between two moving averages of prices. The MACD is calculated by subtracting the 26-day exponential moving average (EMA) from the 12-day EMA. A nine-day EMA of the MACD, called the "signal line", is then plotted on top of the MACD, functioning as a trigger for buy and sell signals.

Moving average convergence divergence (MACD) is among the technical indicators with a huge popularity when it comes to trading. The MACD is a preferred method by traders worldwide, because it is simple to understand and also flexible. It is usually used both as a trend-following indicator and as one gauging momentum.

• **Stochastic Momentum Index - SMI**

The stochastic momentum index (SMI) is a technical analysis indicator that shows price momentum by calculating its closing price distance relative to its median high-low price range. The SMI attempts to improve upon the traditional stochastic oscillator.

The Stochastic oscillator is a technical momentum indicator that compares a security's closing price to its price range over a given time period. The oscillator's sensitivity to market movements can be reduced by adjusting the time period or by taking a moving average of the result. This indicator is calculated with the following formula:

%K=100[(C- L14)/(H14- L14)]

Where,

* C=the most recent closing price
* L14=the low of the 14 previous trading sessions
* H14 = the highest price traded during the same 14-day period.
* %D = 3-period moving average of %K

• **Commodity Channel Index - CCI**

The Commodity Channel Index (CCI) measures the current price level relative to an average price level over a given period of time. CCI is relatively high when prices are far above their average. CCI is relatively low when prices are far below their average. Using this method, CCI can be used to identify overbought and oversold levels.

An oscillator used in technical analysis to help determine when an investment vehicle has been overbought and oversold. The Commodity Channel Index, first developed by Donald Lambert, quantifies the relationship between the asset's price, a moving average (MA) of the asset's price, and normal deviations (D) from that average. It is computed with the following formula:

CCI=price– MA 0.015 x D

• **Collateralized Mortgage Obligation - CMO**

A collateralized mortgage obligation(CMO) is a type of mortgage-backed security in which principal repayments are organized according to their maturities and into different classes based on risk. A collateralized mortgage obligation is a special purpose entity that receives the mortgage repayments and owns the mortgages it receives cash flows from (called a pool). The mortgages serve as collateral, and are organized into classes based on their risk profile. Income received from the mortgages is passed to investors basedon a predetermined set of rules, and investors receive money based on the specific slice of mortgages invested in (called a tranche).

A collateralized mortgage obligation is a type of complex debt security that repackages and directs the payments of principal and interest from a collateral pool to different types and maturities of securities, thereby meeting investor needs.

• **Rate of Change – ROC**

The price rate of change (ROC) is a technical indicator that measures the percentage change between the most recent price and the price "n" periods in the past. It is calculated by using the following formula:

(Closing Price Today – Closing Price "n" Periods Ago)/Closing Price "n" Periods Ago ROC is classed as a price momentum indicator or a velocity indicator because it measures the rate of change or the strength of momentum of change.

The rate of change (ROC) is the speed at which a variable changes over a specific period of time. ROC is often used when speaking about [momentum](https://www.investopedia.com/terms/m/momentum.asp), and it can generally be expressed as a ratio between a change in one variable relative to a corresponding change in another; graphically, the rate of change is represented by the slope of a line.

• **Average Directional Index - ADX**

The average directional index (ADX) is an indicator used in technical analysis as an objective value for the strength of trend. ADX is non-directional so it will quantify a trend's strength regardless of whether it is up or down. ADX is usually plotted in a chart window along with two lines known as the DMI (Directional Movement Indicators). ADX is derived from the relationship of the DMI lines.

The average directional index (ADX) is a technical analysis indicator used by some traders to determine the strength of a trend. The trend can be either up or down, and this is shown by two accompanying indicators, the Negative Directional Indicator (-DI) and the Positive Directional Indicator (+DI).

**Williams % R - WPR**

Williams % R - WPR

Williams %R, in technical analysis, is a momentum indicator measuring overbought and oversold levels, similar to a stochastic oscillator. It was developed by Larry Williams and compares a stock's close to the high-low range over a certain period of time, usually 14 days.

Williams %R, also known as the Williams Percent Range, is a type of momentum indicator that moves between 0 and -100 and measures overbought and oversold levels.

**6.2.2 ALGORITHM DESIGN**

**Using Support Vector Machines**

We will be using C-classification Support Vector Machine with RBF Kernel.

Step 1: Read the required data [Date, Open, High, Low, Close,Volume, Adjusted]

Step 2: Calculate all the required indicators -

1. RSI

2. EMA Crossover

3. CCI

4. ROC

5. CMO

6. WPR

7. ADX

Step 3: Calculate the prediction variable (Up/Down)

Step 4: Provide the data from above steps to train SVM(RBF,C=1,gamma=½)

Step 5: Provide test data and display the results

Step 6:Compare the output from step 5 and 6 and show the observations.

**6.2.3 LANGUAGE USED   
R (Programming Language)**

**R** is a 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

* Quantmod
* lubridate
* e1071
* Rpar
* ROCR

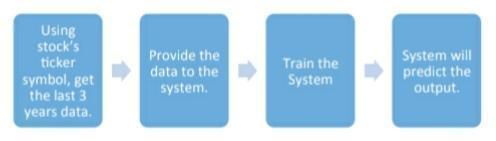
**6.2.4 TOOLS USED   
R Studio Desktop [5]**

R Studio is a free and open source integrated development environment (IDE) for R, a programming language for statistical computing and graphics. R Studio was founded by JJ Allaire, creator of the programming language ColdFusion. Hadley Wickham is the Chief Scientist at R Studio.

* R Studio is available in two editions: R Studio Desktop, where the program is run locally as a regular desktop application; and R Studio Server, which allows accessing R Studio using a web browser while it is running on a remote Linux server. Prepackaged distributions of R Studio Desktop are available for Microsoft Windows, Mac OS X, and Linux. R Studio is available in open source and commercial editions and runs on the desktop(Windows, Mac, and Linux) or in a browser connected to R Studio Server or R Studio Server Pro (Debian/Ubuntu, RedHat/CentOS, and SUSE Linux).
* R Studio is written in the C++ programming language and uses the Qt framework for its graphical user interface.

**6.3 HOW TO GENERATE OUTPUT?**

Perform following steps to generate output:

  
a) Using stock’s ticker symbol, get the last 3 years’ data.  
b) Provide the data to the system.  
c) Train the system.  
d) System will predict the output.

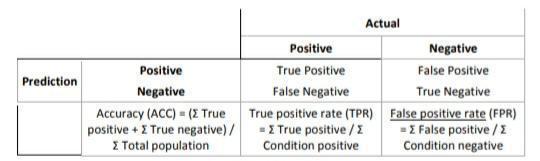
*Fig 6.2 Output Generation*

**6.4 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).



*Table 6.1 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.

**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.

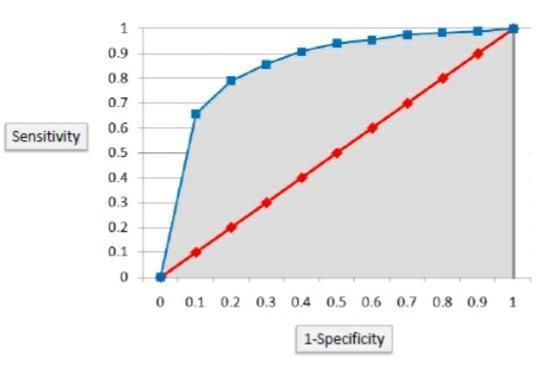
**ROC Chart :**

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).



*Fig 6.3 ROC chart*

**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 to1. In practice, most of the classification models have an AUC between 0.5 and 1.

*Fig 6.4 AUC Chart*

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).

**RMS Error**

The regression line predicts the average y valuebassociated with a given x value. Note that is also necessary to get a measure of the spread of theny 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 the spread of

**CHAPTER 7**

**DESIGN DOCUMENT AND FLOWCHART**

**7.1 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 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

**7.2 PROGRAM SOURCE CODE AND DOCUMENTATION**

**7.2.1 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)

#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","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,

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')

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('-------------------------------------------------------------------- -----') }

**7.3 DATA FLOW DIAGRAM**

**START**

**READ STOCK DATA**

**CALCULATE INDICATOR VARIABLE**

**CREATE INPUT DATA FRAME**

**DIVIDE DATASET INTO TRAINING AND TESTING DATASET**

**TRAIN THE SYSTEM USING TRAINING DATASET**

**TEST THE SYSTEM USING TESTING DATASET**

DISPLAY THE OUTPUT

*Fig 7.1 Data Flow Diagram*

**CHAPTER 8**

**TESTING**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.1 TESTING OBJECTIVE**

Software testing is an important phase in the development of the system. Generally, system testing involves testing integration of each module in the system. The objective while testing the system is to test the discrepancies between the system and the original objective. The quality of an information system depends on its design, development and implementation. Testing is the most important activity in the development phase. Testing is the process of finding errors or bugs in the system. Testing ensure that the user‟s needs are satisfied. In other words it is a process by which one detects the defects in the system.

* Unit testing
* Integration testing
* User interface testing
* Functional testing
* System testing

**8.2 Unit testing**

This is the first level of testing. In here different modules are tested against the specification produced during the design of the modules. Unit testing is done for the verification of code during the coding of single program module in an isolated environment. Unit testing first focuses on the modules independently of one another to locate errors. After coding each task is tested and run individually. All unnecessary coding were removed and it was ensured that all the modules worked, as the programmer would expect. Logical errors found were corrected. Thereby working all the modules independently and verifying the outputs of each module.

**8.3 Integration testing**

Integration testing is the systematic technique of for constructing the program structure while conducting test to uncover errors associated with integrating after unit testing each modules are integrated to from one fine system. All the modules when unit tested will work properly but after integrating the data can cause error. One module can have an inadvertent, adverse effect on another; sub functions when combined may not produce the desired major function; global data structures can cause problems, etc.

Hence, the objective of integration testing is to take unit tested modules and build a final program structure. In this project, modules are combined to find the overall performance of the system.

In this project the modules are integrated properly, the emphasis being and testing interfaces between modules. Thus all these modules are combined, verified and the information about the items is properly carried on to the next module and then it is checked.

**8.4 User Interface Testing**

Tests are often generated using a components interface. Certainly, the interface itself forms a part of the components requirements and hence this form of testing is often called black box testing. However, the focus on interface leads us to consider interface testing in its own right. Techniques such as pair wise testing and interface mutation are used to generate tests from a components interface specification**.**

**8.5 Performance Testing**

Performance testing is designed to test the run-time performance of the software within the context of an integrated system.

Performance testing (Navigation Testing) in the “Enhanced JOB Services” is done from the first till the end of the system process to check out the performance to know whether it satisfies the user requirements.

**8.6 System Testing**

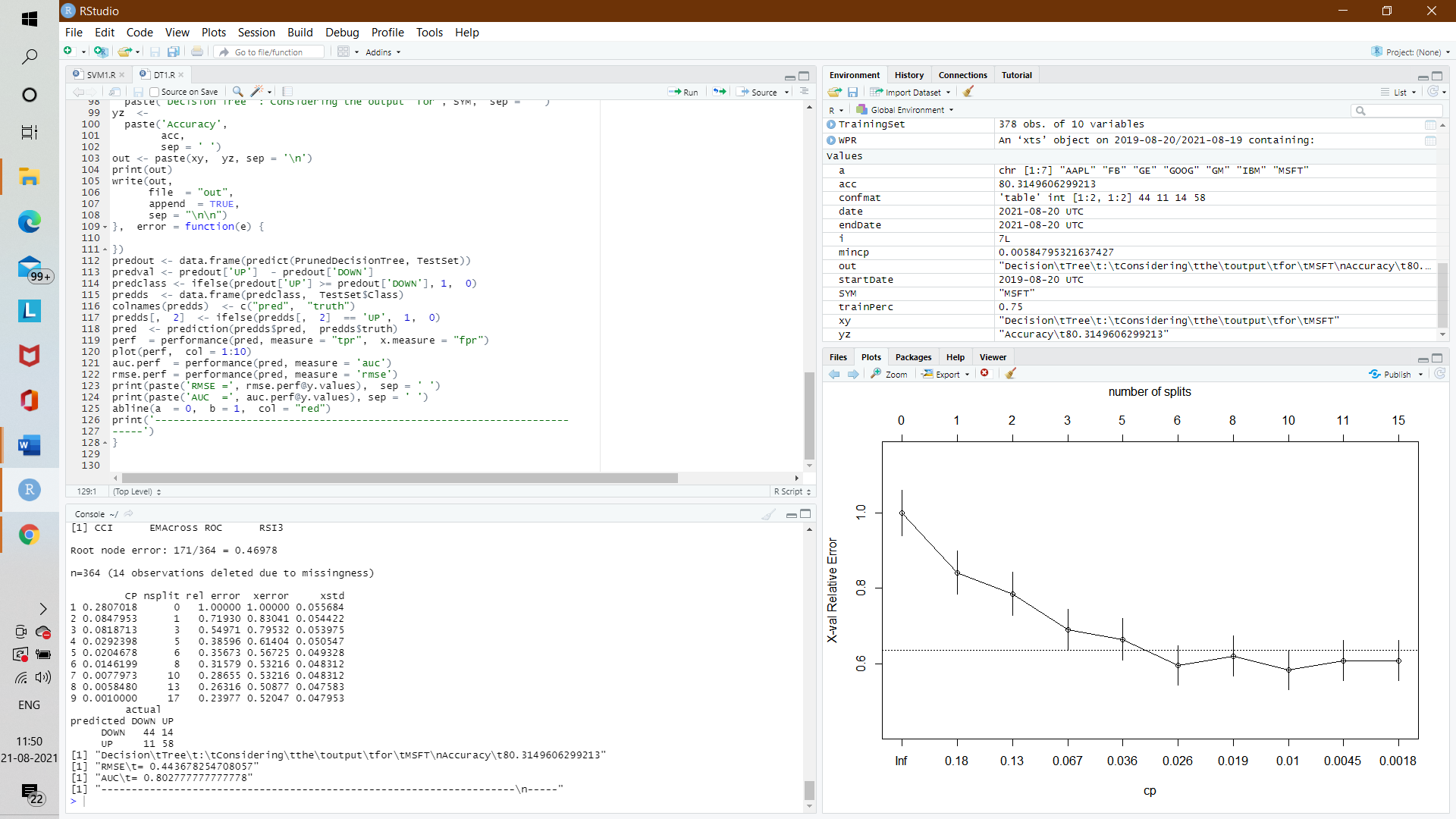
After every module is integrated, the system test is performed. System testing does not test the software but the integration of each module in the system. It also tests to find discrepancies between the system and its original objective, current specifications and systems documentation. The primary concern is the compatibility of individual modules.

**CHAPTER 9**

**IMPLEMENTATION OUTPUT**

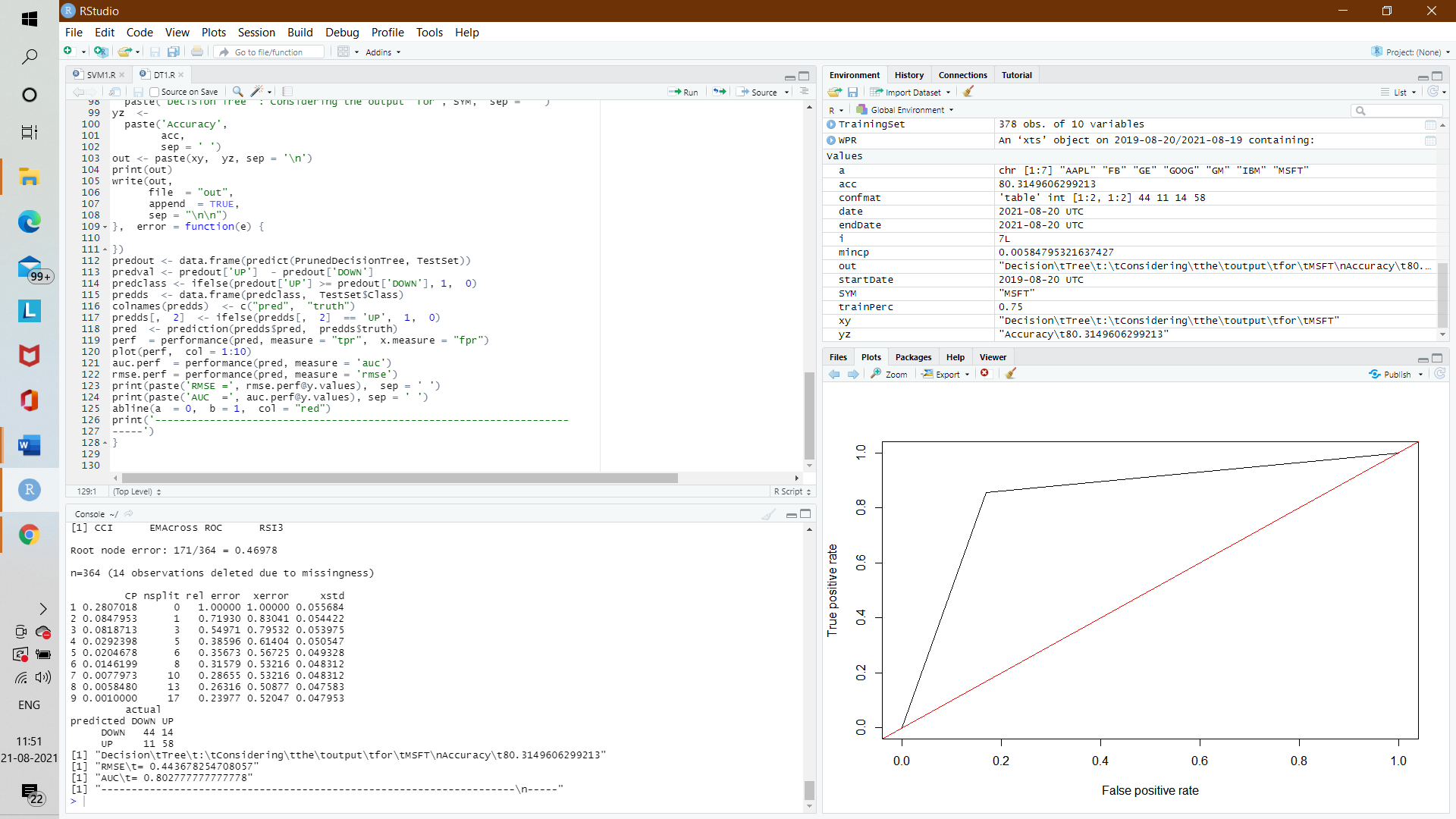
**9.1 AAPL**

**RMSE CHART:**



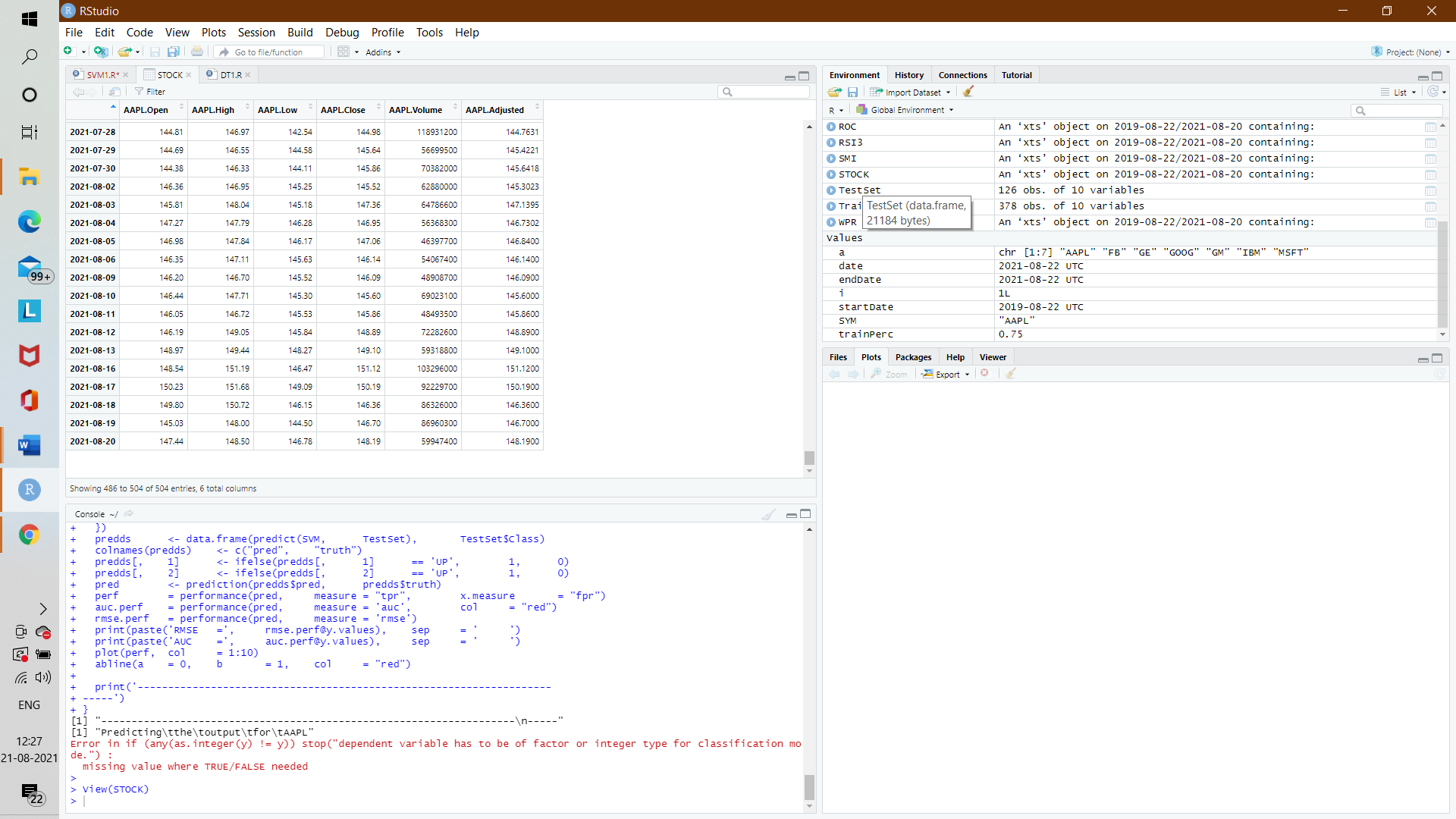
*Fig 9.1 APPL RMSE chart*

**AUC CHART**



*Fig 9.2 APPL AUC chart*

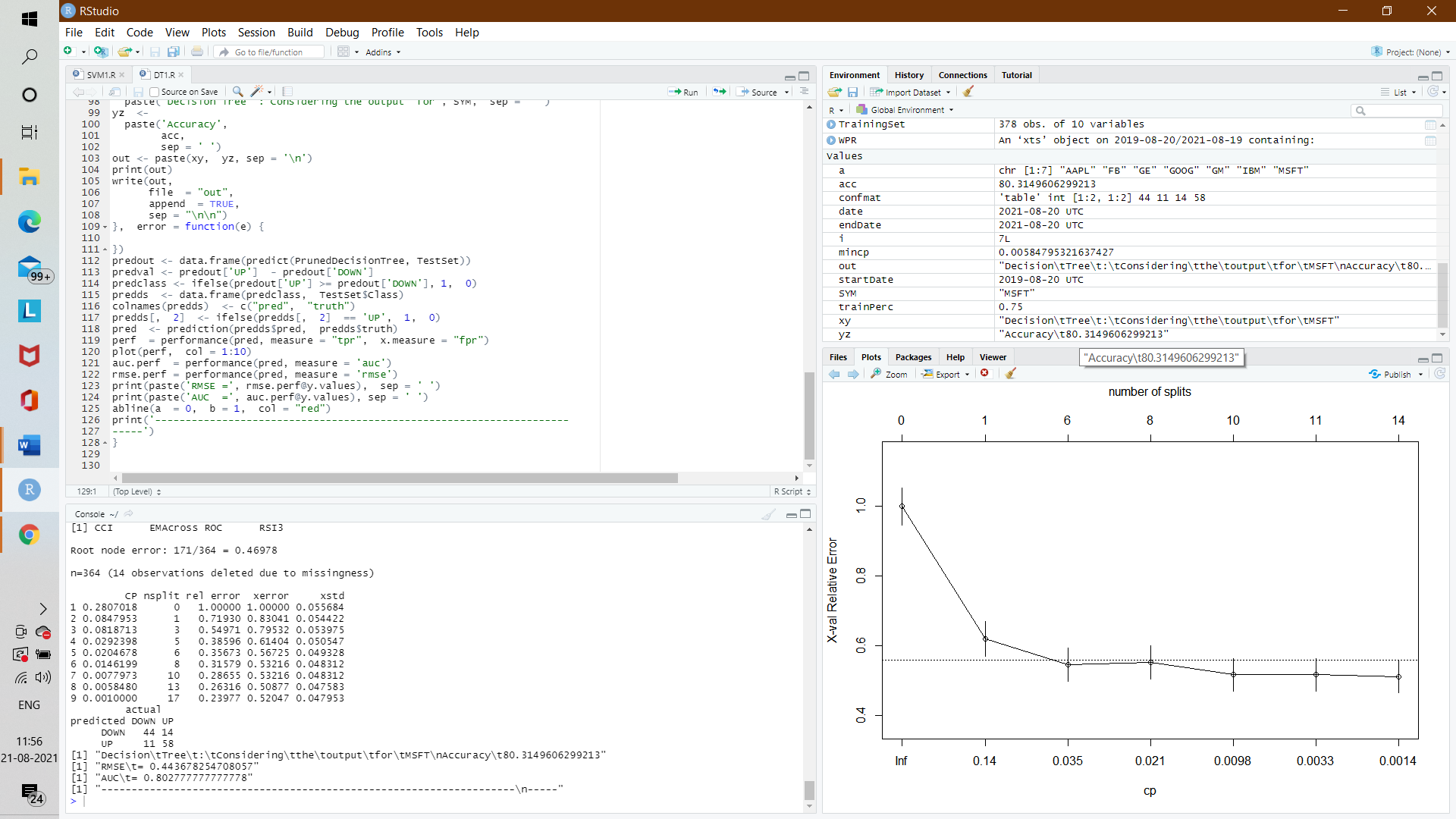
**APPL STOCK OUPUT:**



*Table 9.1 APPL Stock output*

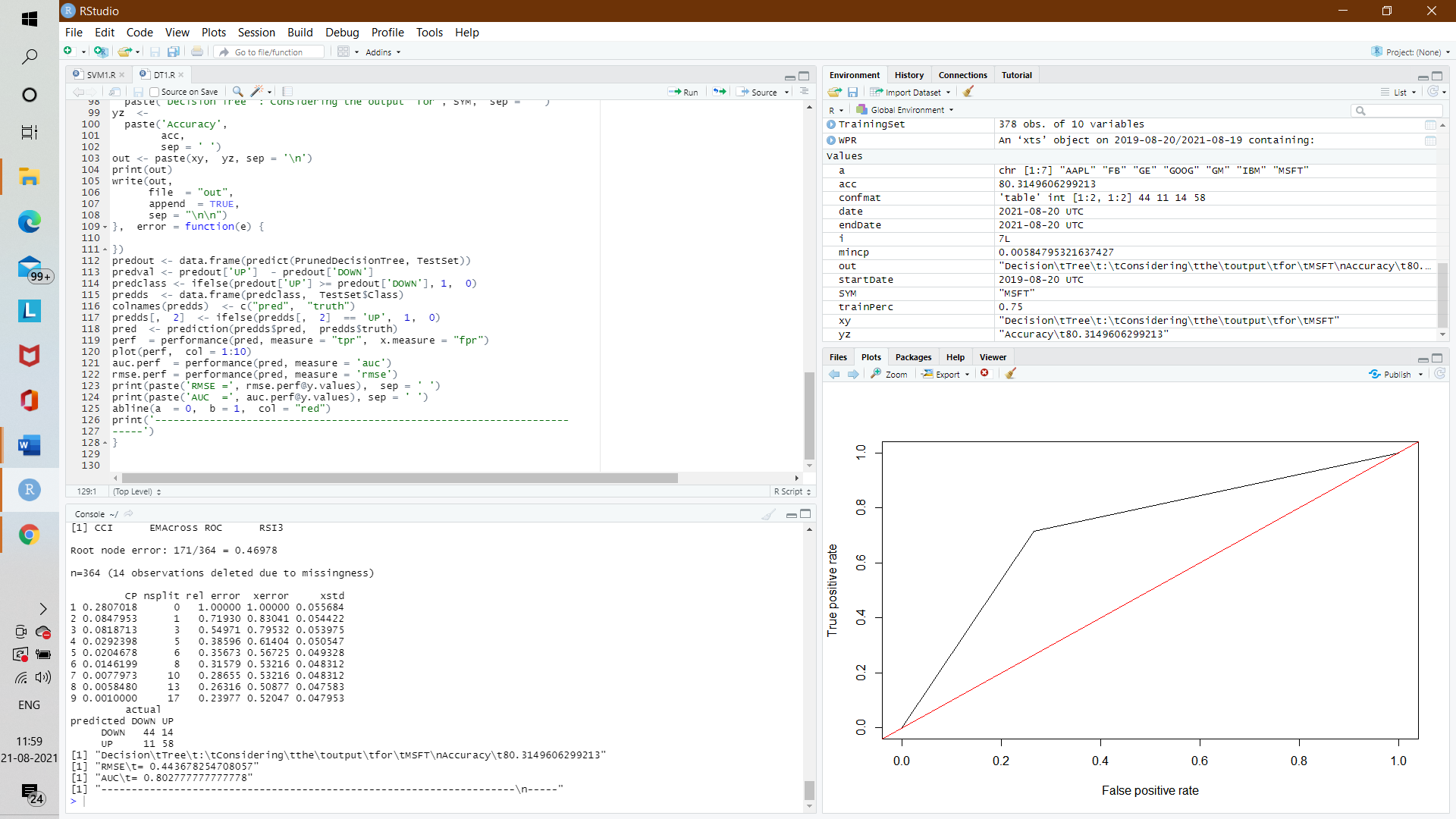
**9.2 FB**

**RMSE CHART:**



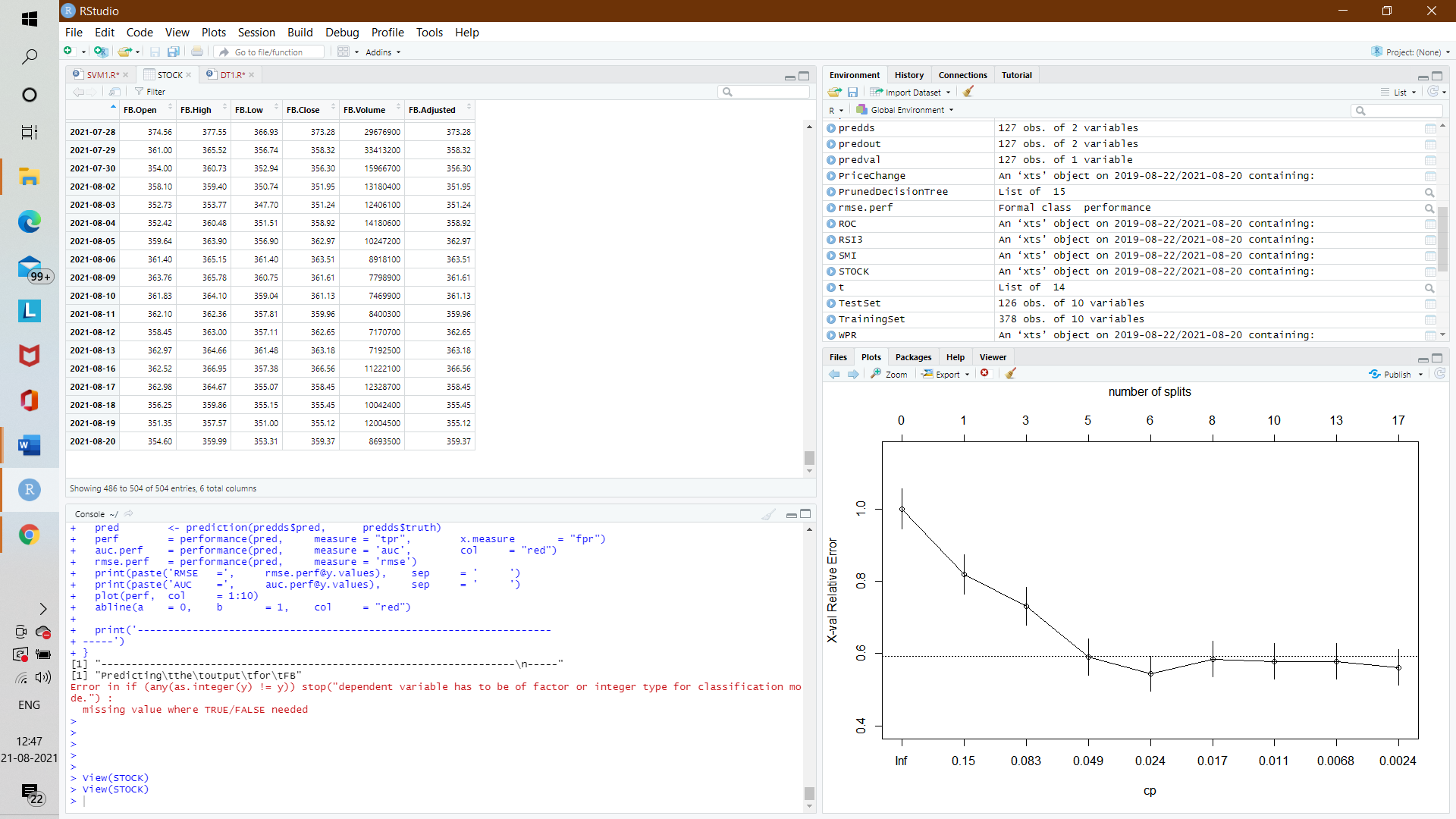
*Fig 9.3 FB RMSE chart*

**AUC CHART:**



*Fig 9.4 FB AUC Chart*

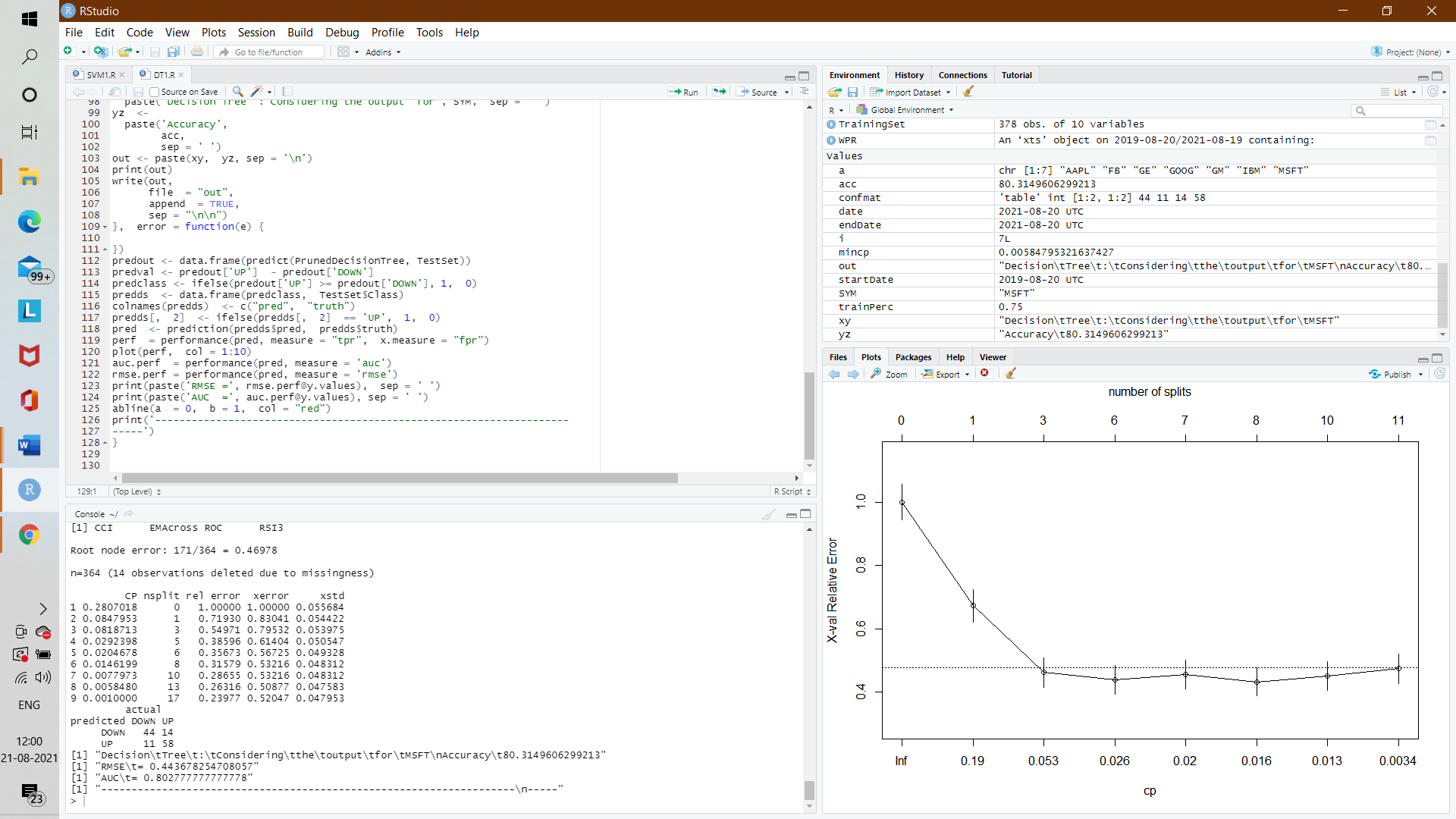
**FB STOCK OUTPUT:**



*Table 9.2 FB stock output*

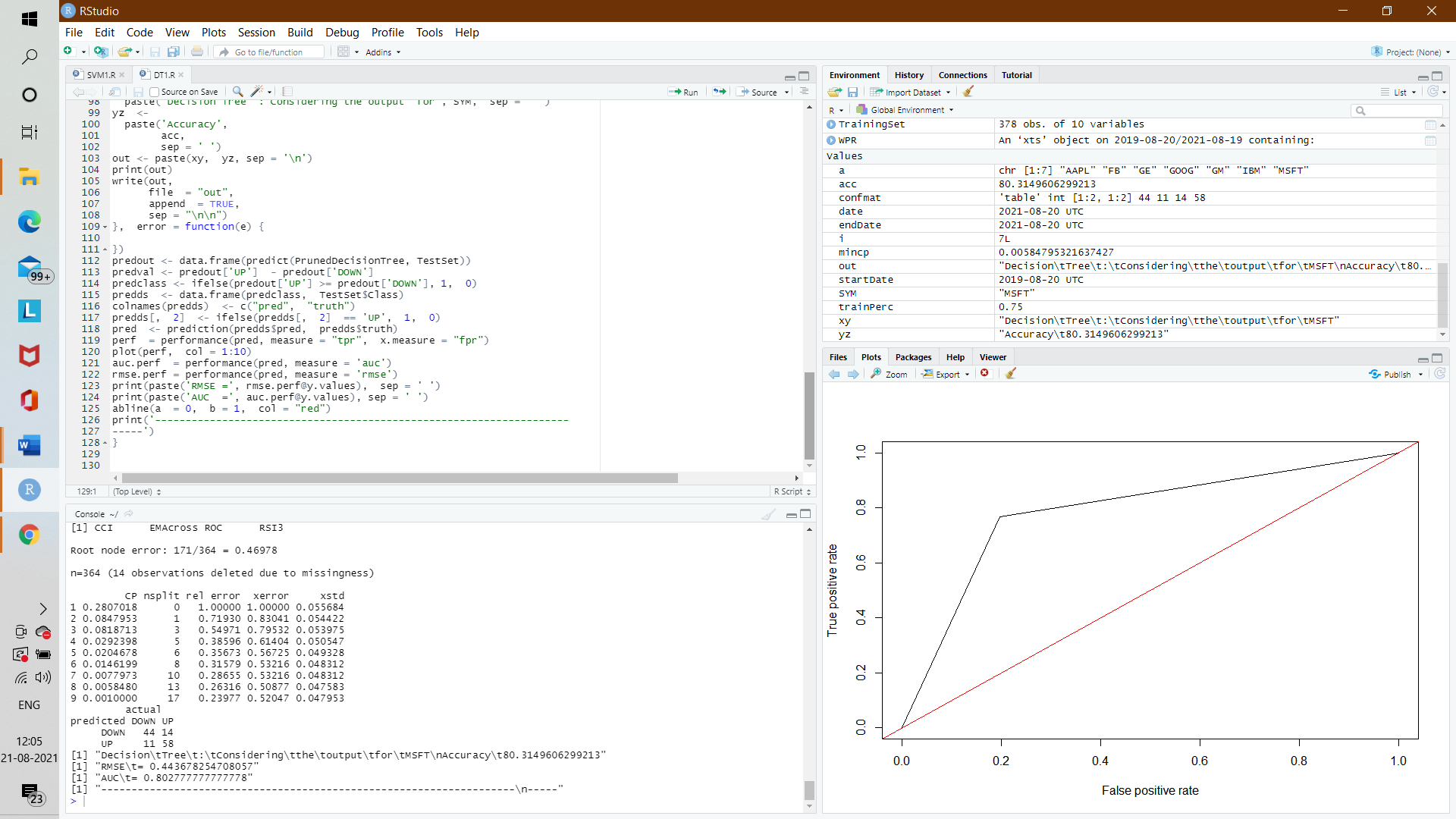
**9.3 GE**

**RMSE CHART:**



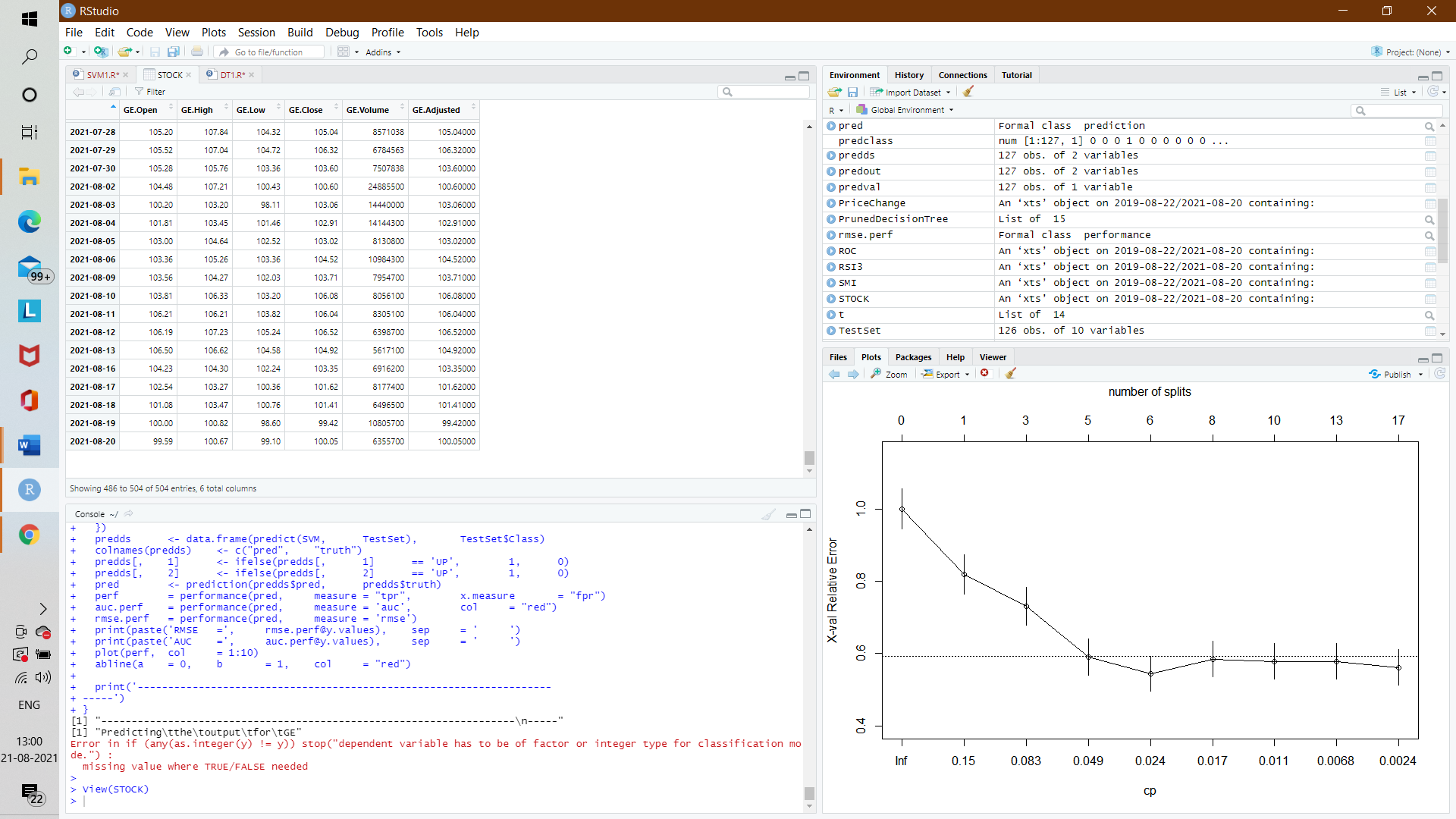
*Fig 9.5 GE RMSE Chart*

**AUC CHART:**



*Fig 9.6 GE AUC Chart*

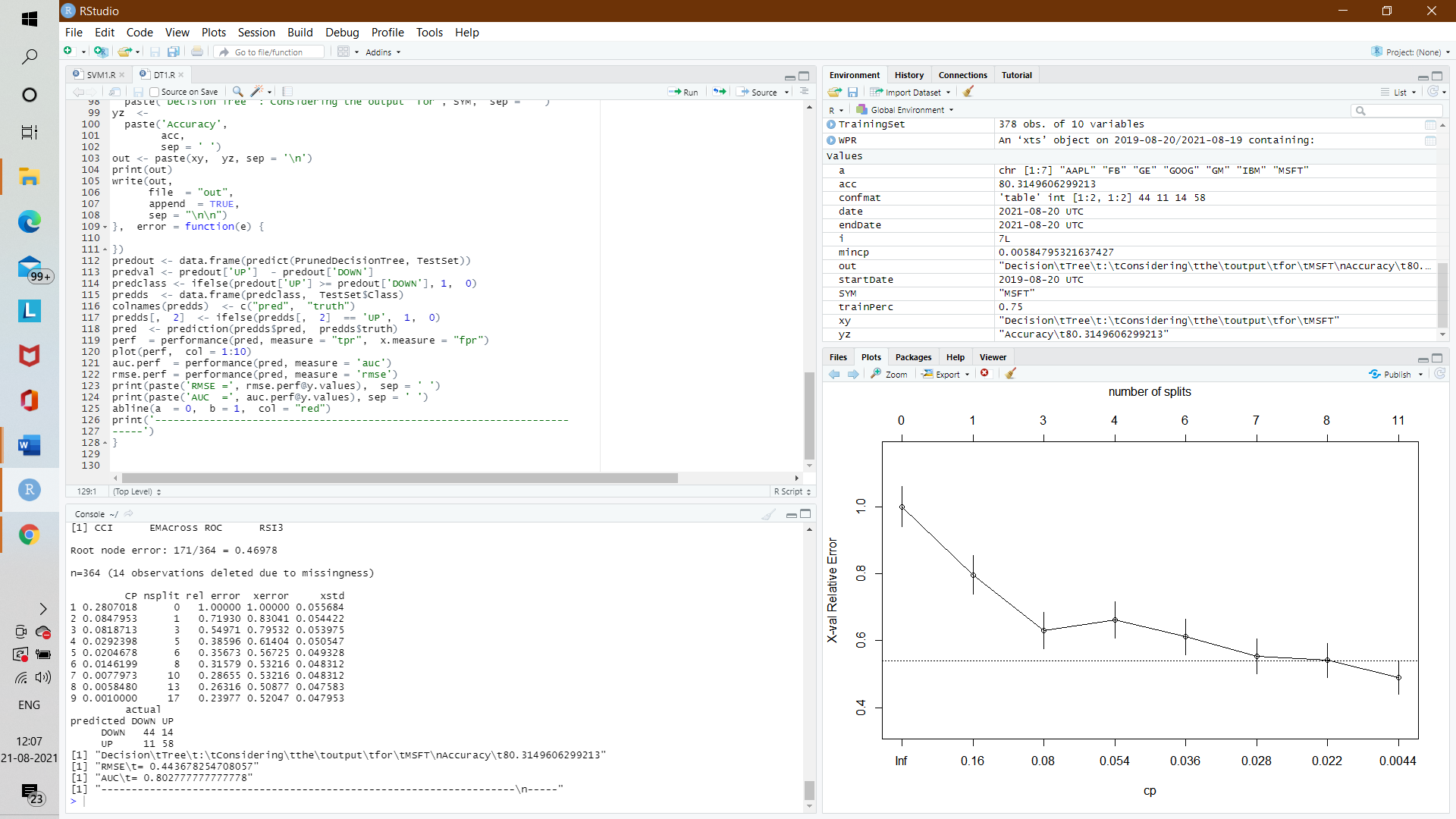
**GE STOCK OUTPUT:**



*Table 9.3 GE Stock ouput*

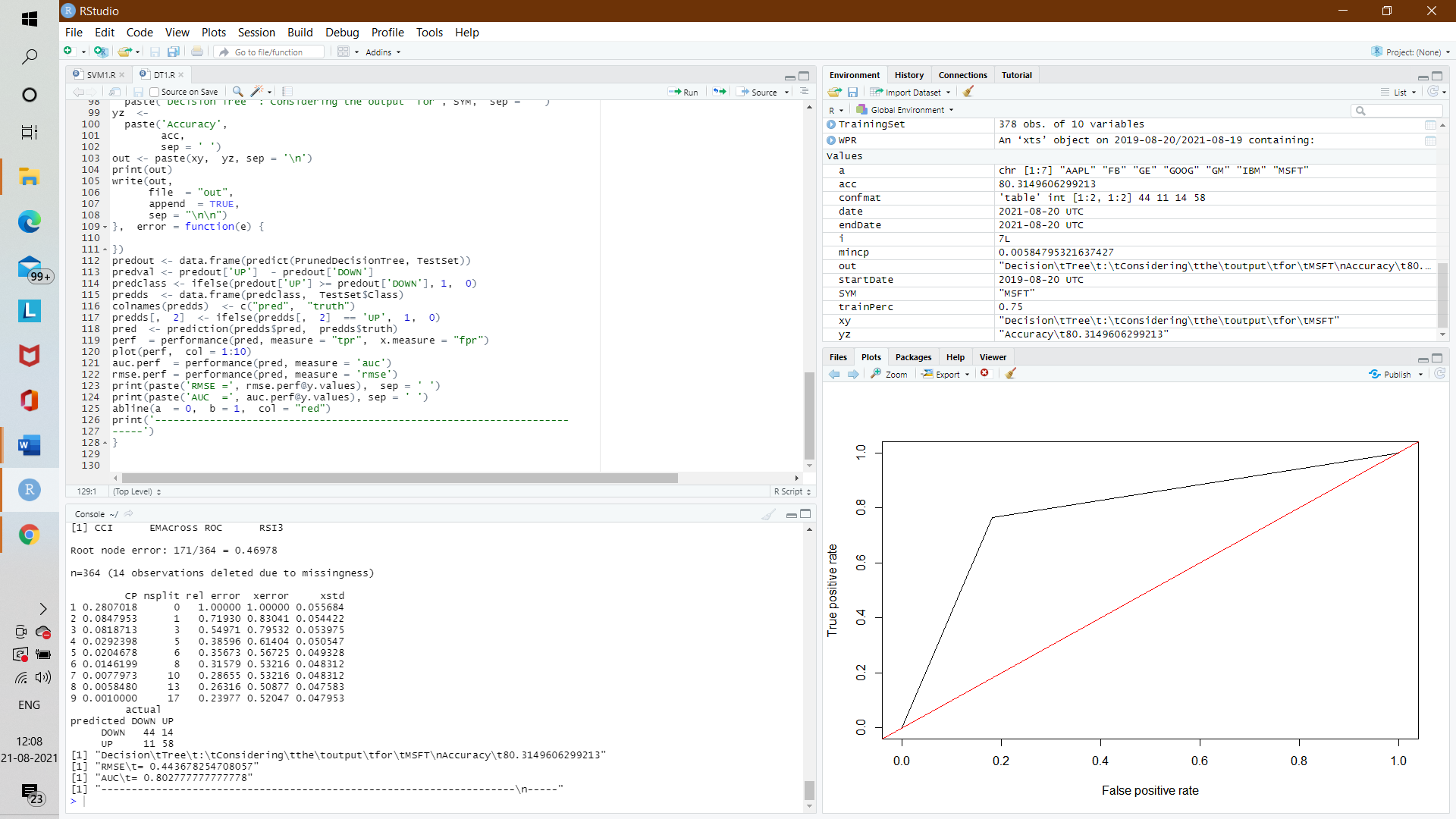
**9.4 GOOG**

**RMSE CHART:**



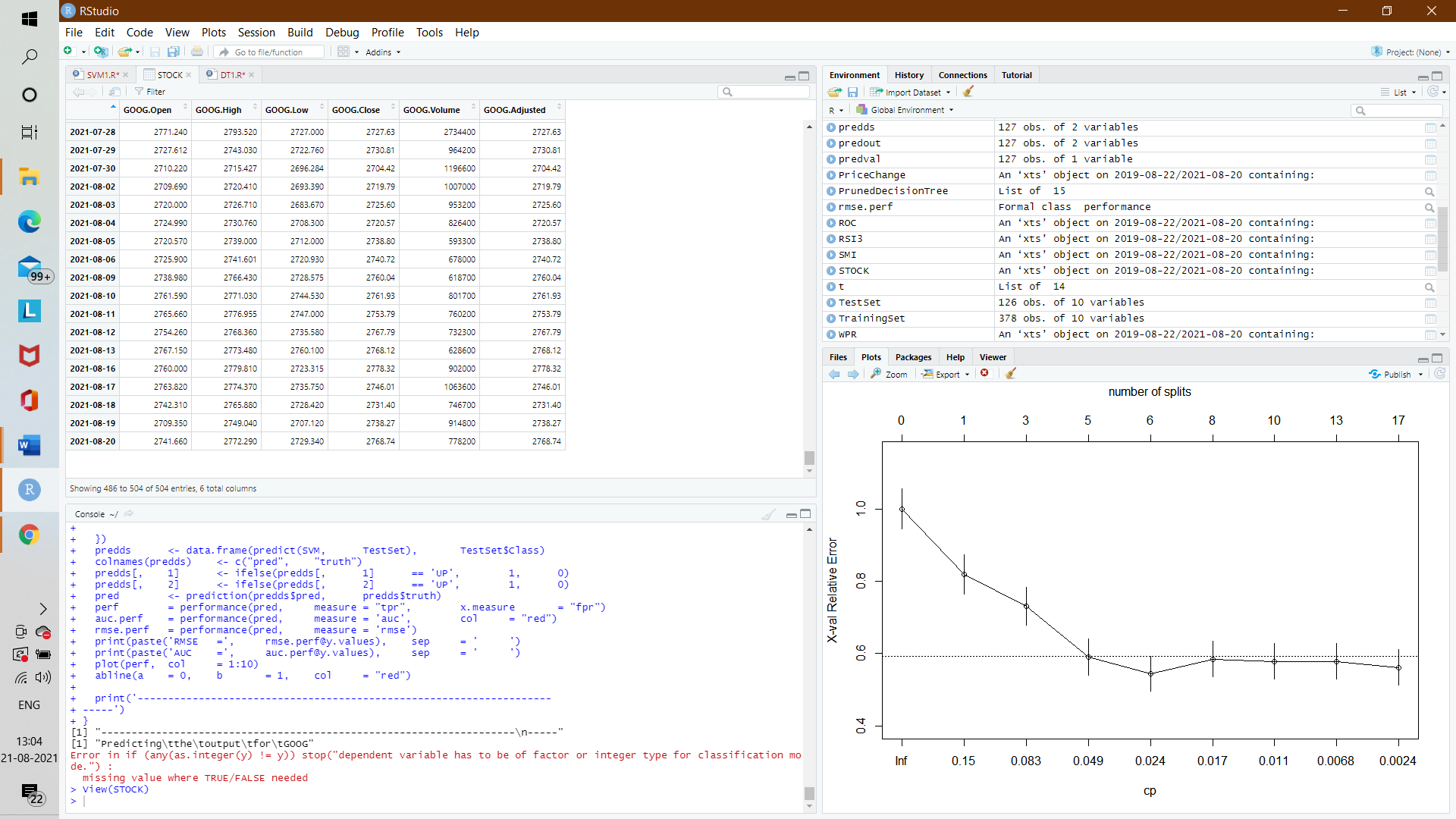
*Fig 9.7 GOOG RMSE Chart*

**AUC CHART:**



*Fig 9.8 GOOG AUC Chart*

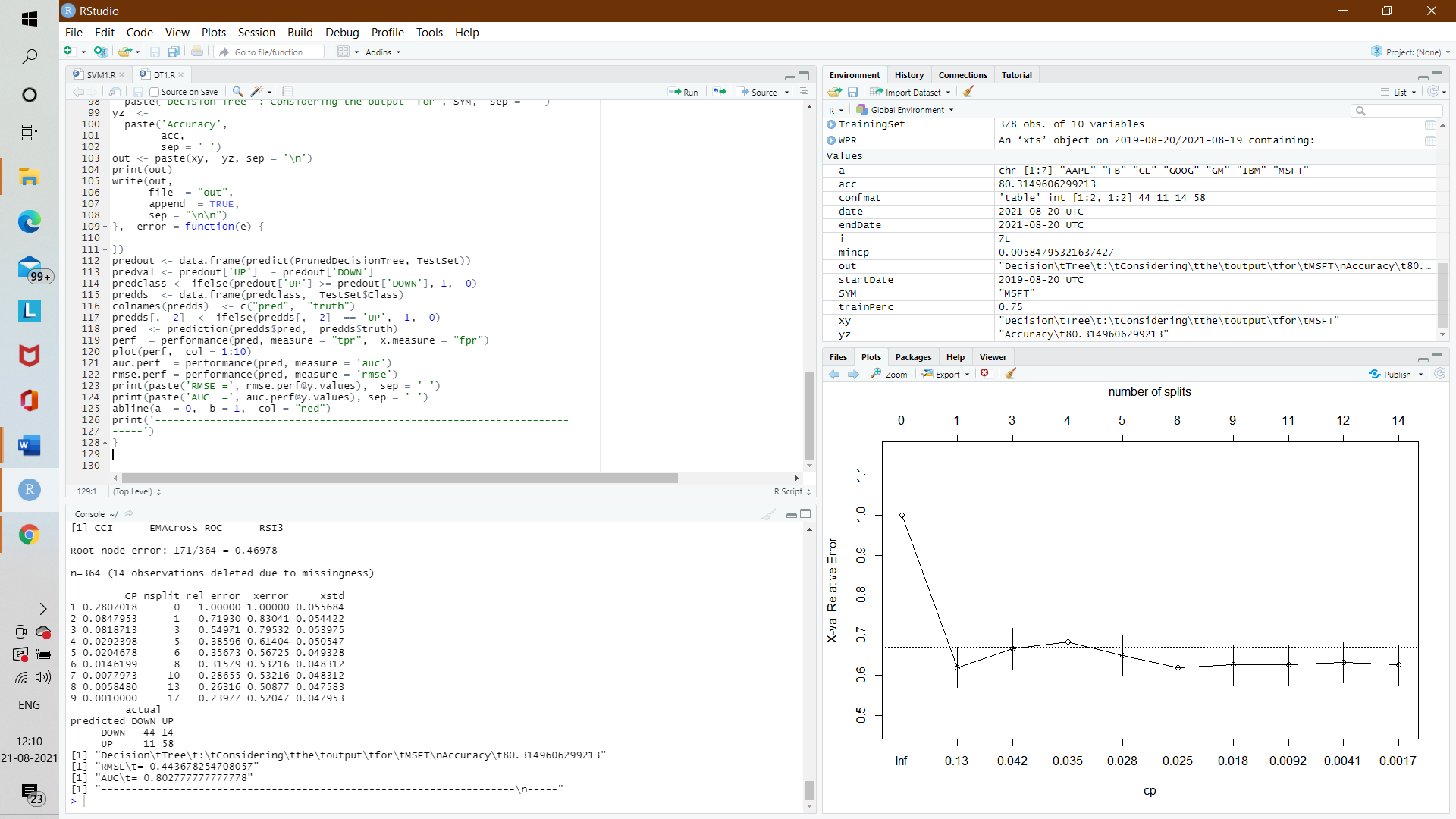
**GOOG STOCK OUTPUT:**



*Table 9.4 GOOG Stock output*

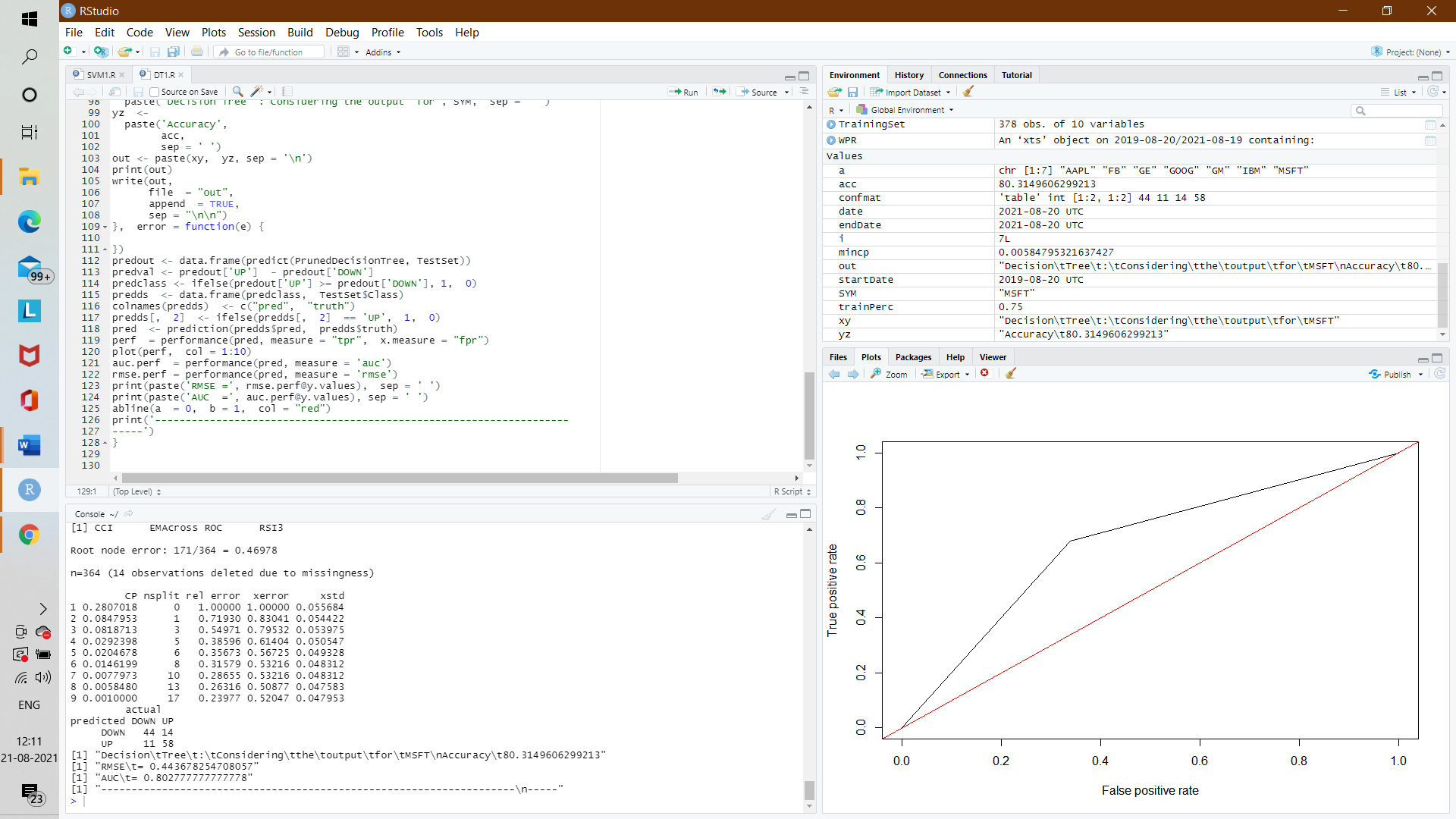
**9.5 GM**

**RMSE CHART:**



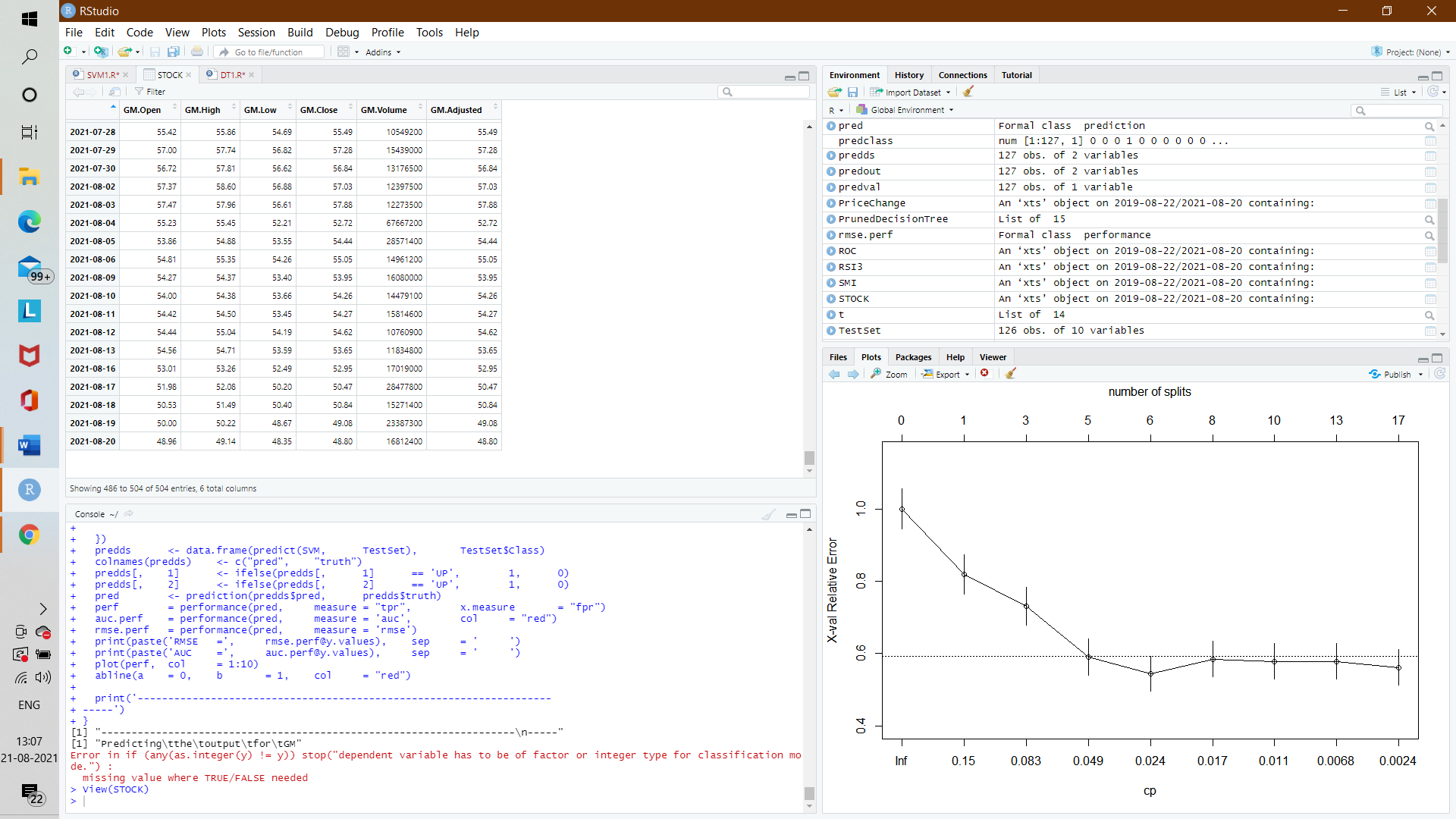
*Fig 9.9 GM RMSE Chart*

**AUC CHART:**



*Fig 9.10 GM AUC chart*

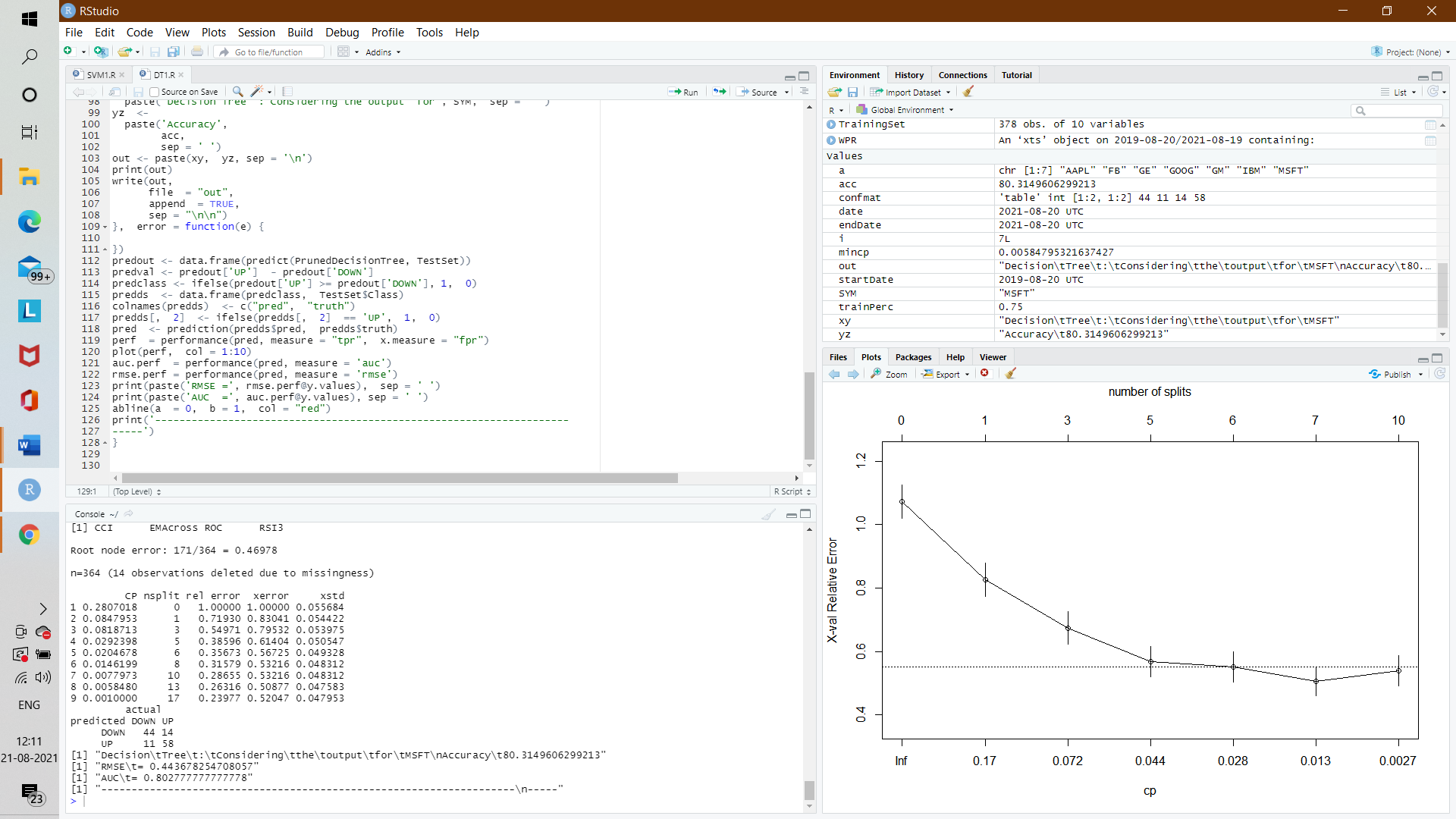
**GM STOCK OUPUT:**



*Table 9.5 GM Stock output*

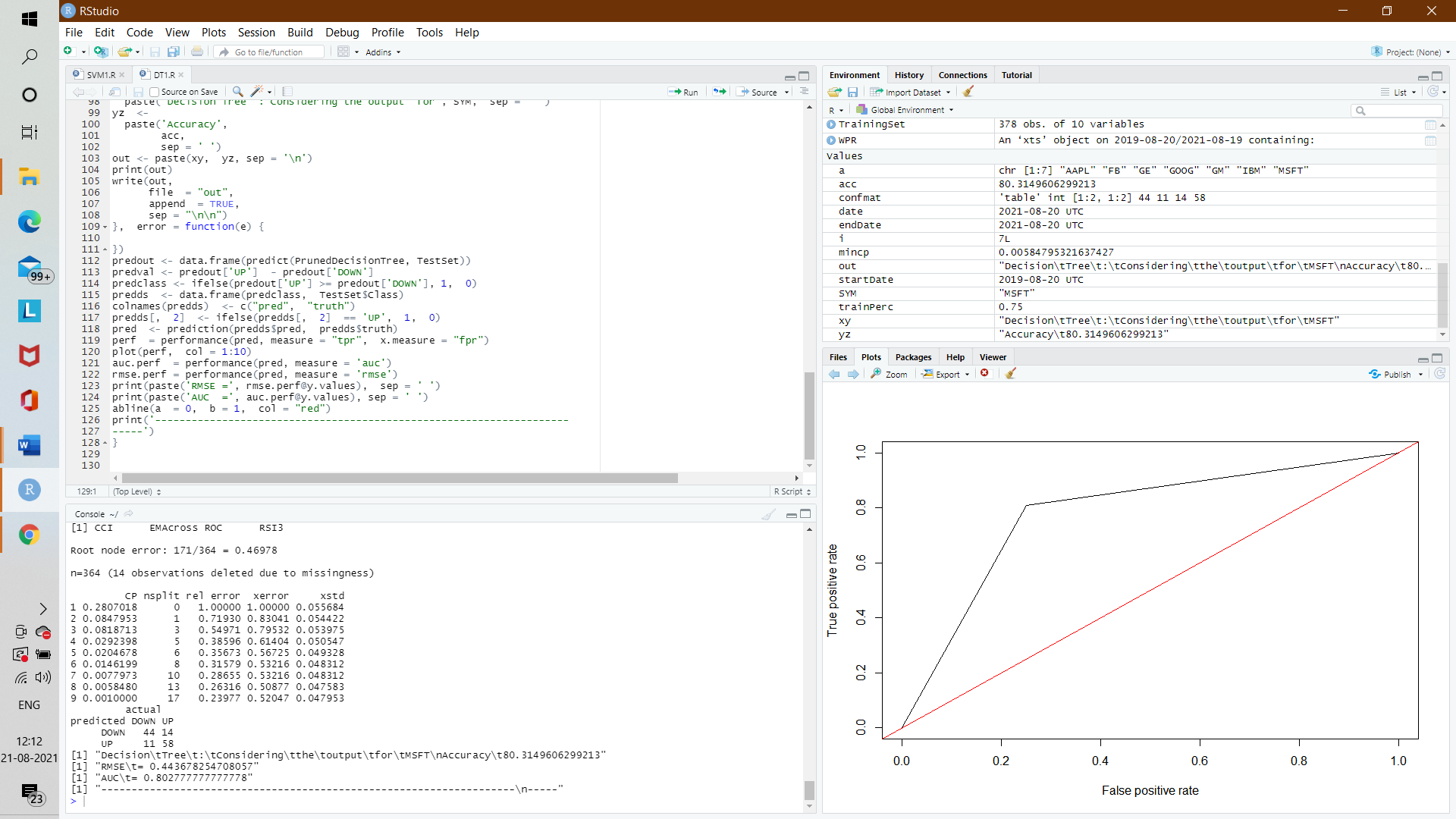
**9.6 IBM**

**RMSE CHART:**



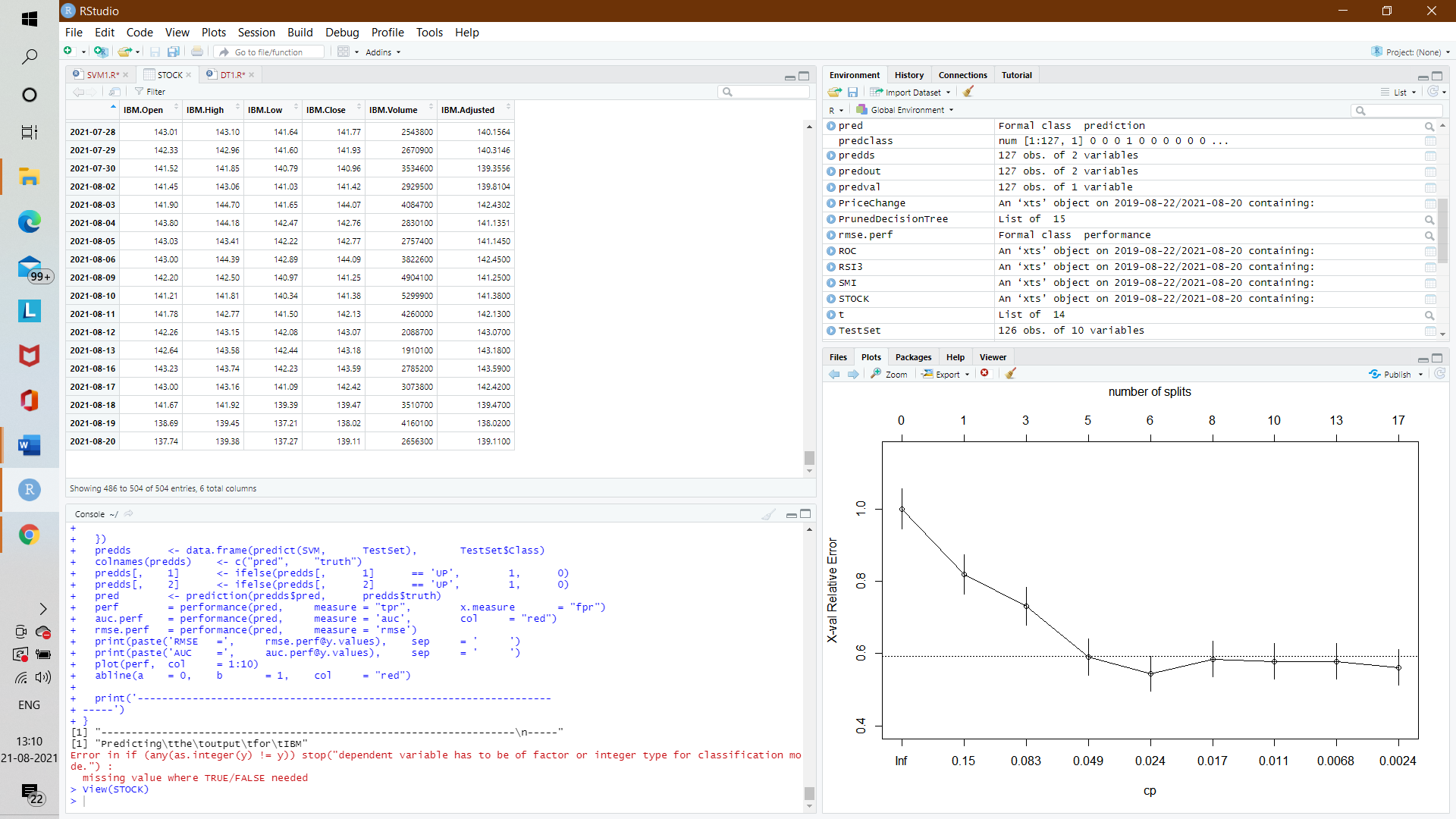
*Fig 9.11 IBM RMSE Chart*

**AUC CHART:**



*Fig 9.12 IBM AUC Chart*

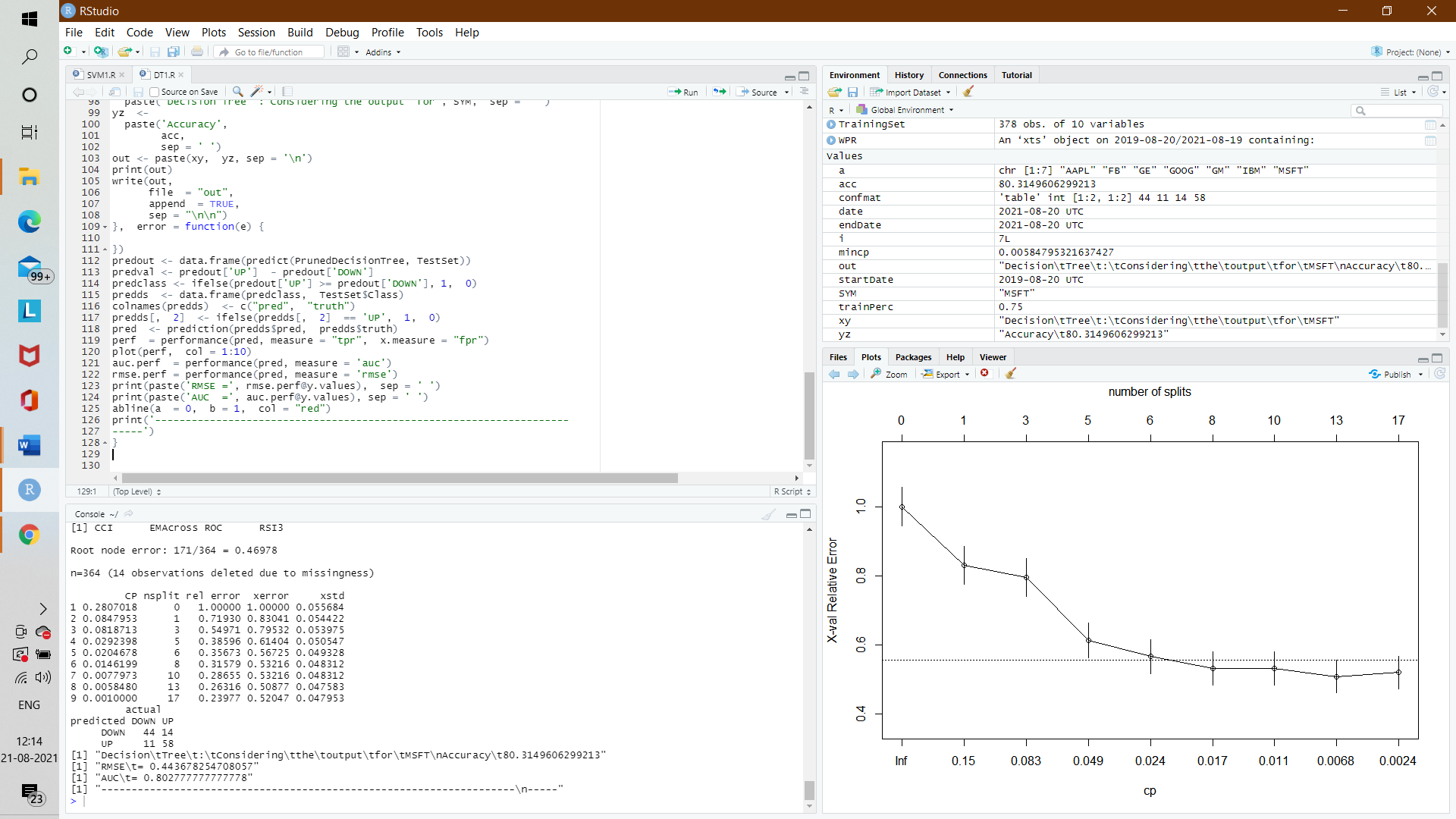
**IBM STOCK OUTPUT:**



*Table 9.6 IBM Stock output*

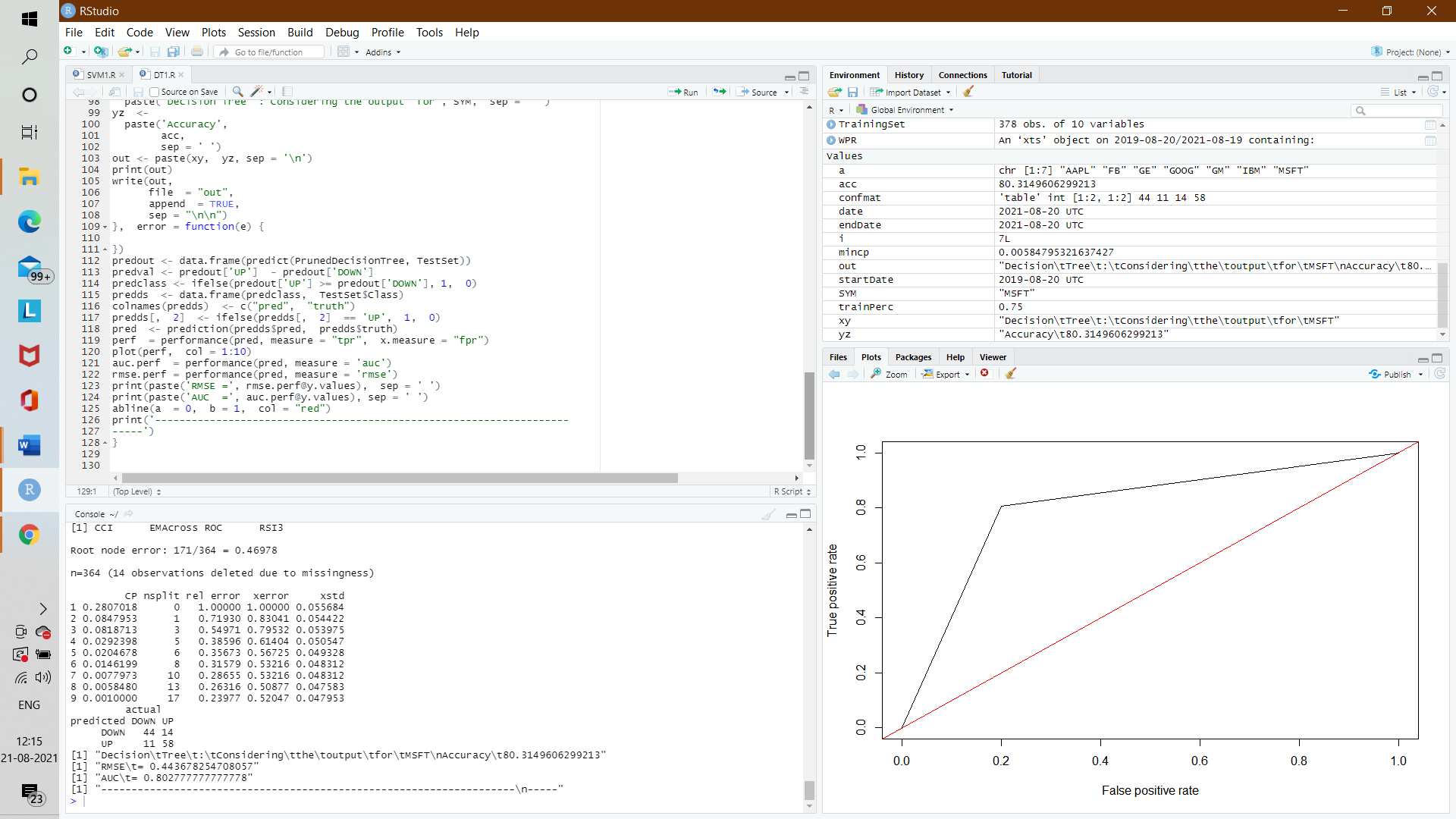
**9.7 MSFT**

**RMSE CHART:**



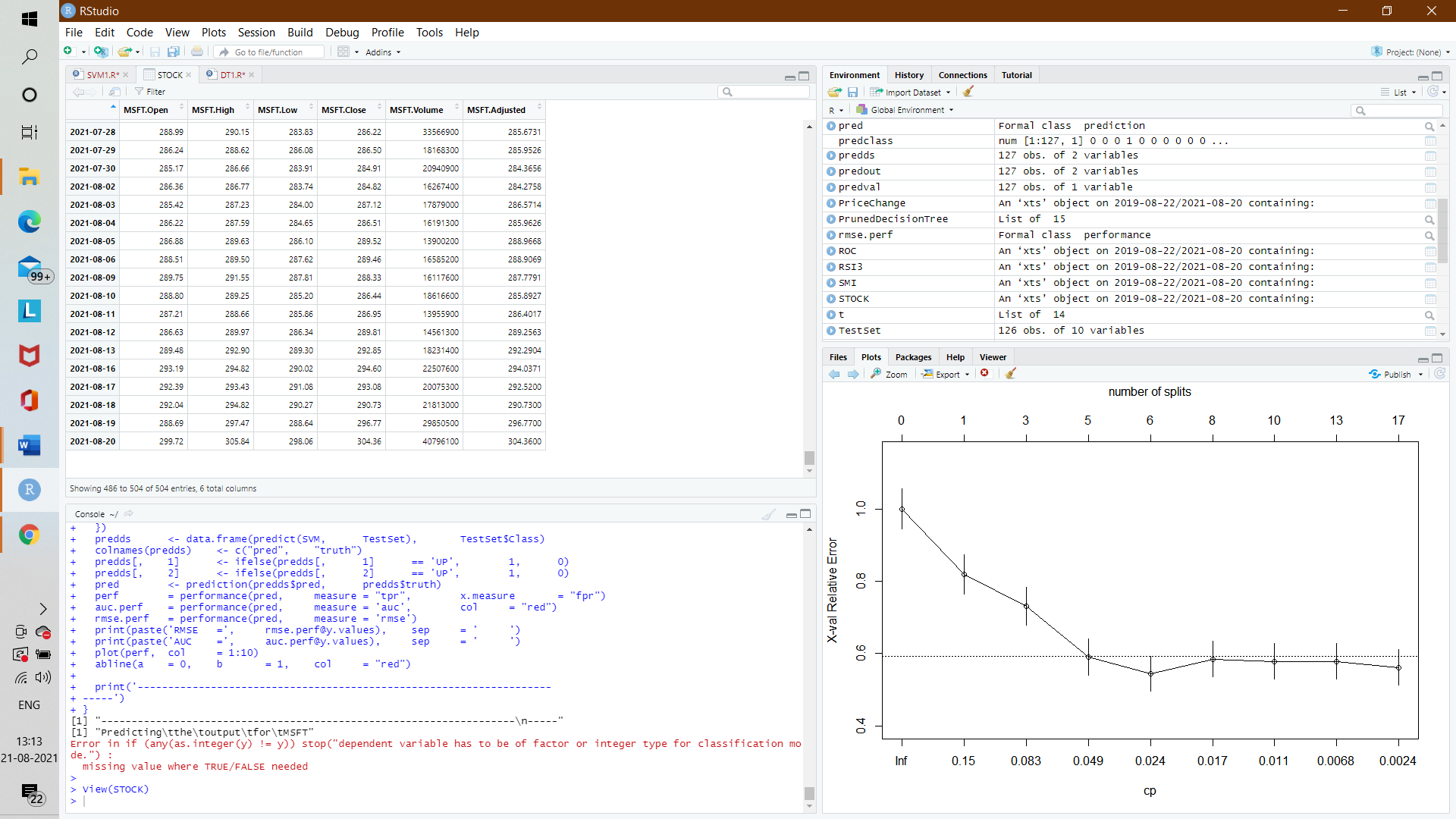
*Fig 9.13 MSFT RMSE Chart*

**AUC CHART:**



*Fig 9.13 MSFT AUC Chart*

**MSFT STOCK OUTPUT:**



*Table 9.7 MSFT Stock output*

**CHAPTER 10**

**CONCLUSION & REFERENCES**

**10.1 CONCLUSION**

SVM is a promising type of tool for financial forecasting. SVM is superior to the other individual classification methods in forecasting daily movement direction. This is a clear message for financial forecasters and traders, which can lead to a capital gain. However, each method has its own strengths and weaknesses. In this model, the principal components identified by the SVM are used along with internal and external financial factors in SVM for forecasting. We also observed that the choice of the indicator function can dramatically improve/reduce the accuracy of the prediction system. Also a particular Machine Learning Algorithm might be better suited to a particular type of stock, say Technology Stocks, whereas the same algorithm might give lower accuracies while predicting some other types of Stocks, say Energy Stocks.

**Input**

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

**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.

**10.2 REFERENCE**

[1]. Prof. V. Vijayarajan, Prof. Anand Mahendran, Prof. A. Krishnamoorthy, Harsh Panday, V.B. Surya Prasath, Dept. of Computer Science & Engineering, Vellore Institute of Technology, Vellore, India. – Stock Market Prediction Using Sentimental analysis, Volume: 07 Issue: 05 | May-2020.

[2]. Prof. Jogi John, Aatish Kumar, Ankit Abhishek, Tejas A. Dhule, Amit Roy, Abhishek Jha, Dept. of Computer Technology, Priyadarshini College Of Engineering, Nagpur, Maharashtra, India – Stock market prediction using Machine Learning, Volume: 07, Issue: 01 | February-2020.

[3]. Mr. Sachin Sampat Patio, Prof.Kailash Patidar, Assistant Prof. Meghan Jain, SSSIST,Sehore,Madhya Pradesh,India – Stock Market Prediction Using Support Vector Machine, Volume:02,Issue:01 | Jan-Feb,2019.

[4]. Rohan Taneja, Vaibhav, Dept. of Computer Science & Engineering, IMS Engineering College, Ghaziabad, India. – Stock Market Prediction Using Regression, Volume: 05 Issue: 05 | May-2019.

[5] S. Liu, G. Liao and Y. Ding, "Stock transaction prediction modeling and analysis based on LSTM," 2019 13th IEEE Conference on Industrial Electronics and Applications (ICIEA), Wuhan, 2019.

[6] K. A. Althelaya, E. M. El-Alfy and S. Mohammed, "Evaluation of bidirectional LSTM for short-and long term stock market prediction," 2018 9th International Conference on Information and Communication Systems(ICICS), Irbid, 2018, pp. 151-156[14] Debashish Das and Mohammad shorif uddin data mining and neural network techniques in stock market prediction: a methodological review, international journal of artificial intelligence & applications, vol.4, no.1, January 2018.

[7] K. Jae Kim, “Financial time series forecasting using support vector machines,” Neurocomputing, vol. 55, 2018.

[8] H. L. Siew and M. J. Nordin, "Regression techniques for the prediction of stock price trend," 2012 International Conference on Statistics in Science, Business and Engineering (ICSSBE), Langkawi, 2017, pp. 1-5.

[9] K. Raza, "Prediction of Stock Market performance by using machine learning techniques," 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT), Karachi, 2017, pp. 1-1.

[10] H. Gunduz, Z. Cataltepe and Y. Yaslan, "Stock market direction prediction using deep neural networks," 2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, 2017, pp. 1-4.

[11] T. Gao, Y. Chai and Y. Liu, "Applying long short term memory neural networks for predicting stock closing price," 2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS),Beijing, 2017, pp. 575-578.

[12] M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.

[13] M. Billah, S. Waheed and A. Hanifa, "Stock market prediction using an improved training algorithm of neural network," 2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), Rajshahi, 2016, pp. 1-4.

[14] K. V. Sujatha and S. M. Sundaram, "Stock index prediction using regression and neural network models under non normal conditions," INTERACT-2010, Chennai, 2015, pp. 59-63.

[15] Zhen Hu, Jibe Zhu, and Ken Tse “Stocks Market Prediction Using Support Vector Machine”, 6th International Conference on Information Management, Innovation Management and Industrial Engineering, 2015.M.

[16] Wei Huang, Yoshiteru Nakamori, Shou-Yang Wang, “Forecasting stock market movement direction with support vector machine”, Computers & Operations Research, Volume 32, Issue 10, October 2015, Pages 2513–2522.

[17] N. Ancona, Classification Properties of Support Vector Machines for Regression, Technical Report, RIIESI/CNR-Nr. 02/99.