**Stock Value Prediction   
using   
Machine Learning**

# **Overview**

This project is based on the IEEE paper “Robust Stock Value Prediction using Support Vector Machines with Particle Swarm Optimization” by Trevor M. Sands, Deep Tayal, Matthew E. Morris, and Sildomar T. Monteiro, Electrical and Microelectronic Engineering, Rochester Institute of Technology, May 2015.

The paper involves two machine learning algorithms, Support Vector Machines (SVM) and Particle Swarm Optimization (PSO). But the implemented project is scoped with the first half of the paper with SVM only.

The project is implemented using “R” programming language which is the most popular language for developing machine learning and data science applications. And stock details of Apple and Google from 2004 to current are used for learning and testing the SVM.

# **Project Details**

Below are the steps involved in developing the project.

1. The Apple and Google stock data are downloaded from the Yahoo finance website in the csv format.
2. In “R” studio the environment is set and all the required libraries are loaded.
3. The downloaded csv files are read and imported in the format in which “R” can recognize it as stock data so that we can apply inbuilt API like generating the chart series, calculating the daily, monthly returns, etc., can be done very easily.
4. A new “Values” column is added to the stock dataset which denotes 1 if the stock in a particular day is higher than the previous day else 0.
5. The Stock dataset is divided into two.   
    1. Training data, containing stock data from 2004 to 2014.   
    2. Testing data, containing stock data of 2015 and “Values” column is renamed to “Actual”
6. The SVM algorithm is applied to the Training data using inbuilt SVM() API. For this project a basic linear SVM classifier is used. So the parameters are given appropriately ad below.  
    svmFit = svm (Values ~ .,

data=trainData,

type="C",  
 kernel= "linear",  
 scale=FALSE)

* 1. The first parameter is the formula which denoted “Values” column need to be learned in order to be predicted for test data and .(dot) denoted the rest of the columns in the dataset are selected as Attributes.
  2. The second parameter denotes the training data
  3. The third parameter represents the type of SVM, which in here is “C” that is a Classifier.
  4. The fourth parameter denotes which kernel function to use. Here, Linear Kernel is used.
  5. The fifth parameter denotes whether the attributes needs to be scaled.

1. The summary for the output of the SVM function is viewed and fitted to check for any errors.
2. Plot the svmFit in a graph to visualize the data and the dividing line which separates the increased stocks and the rest.
3. Predict the values for test data using the svmFit
4. Add the predicted values column to the test data and plot and compare both predicted and Actual values.
5. Calculate the Accuracy using the below formula.  
    Accuracy = (Number of correct values / Total number of predicted values ) \* 100

# **Results**

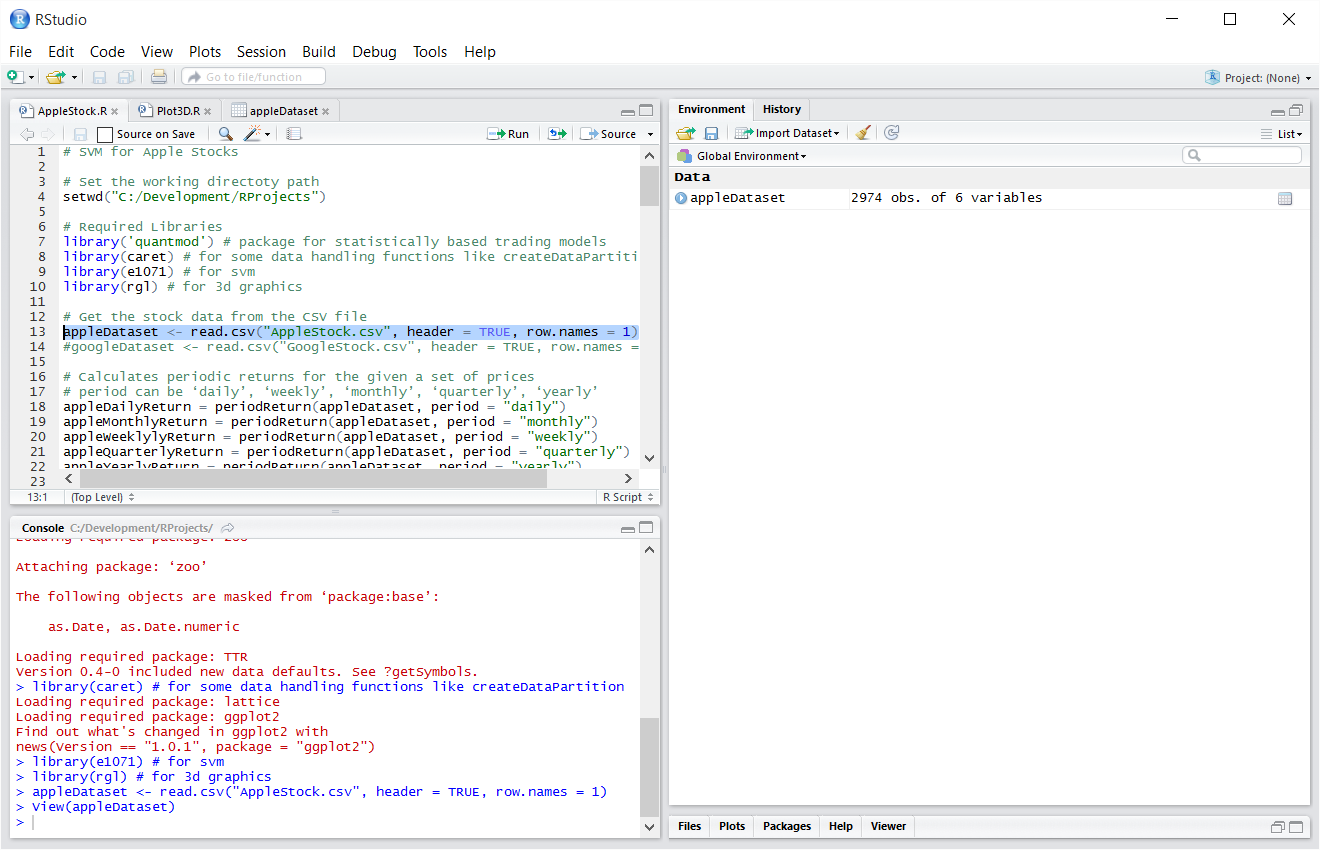
The Accuracy got for Apple daily stock changes is 49.33%

And the accuracy got for Google monthly stock changes is 50.02%

Even though the accuracy is low, it is expected as per the IEEE paper used. Only when this algorithm is used in conjunction with the PSO algorithm we would get around 96% accuracy.

# **Screenshots**

“R” Studio loading csv data.



Loaded Apple stock data set

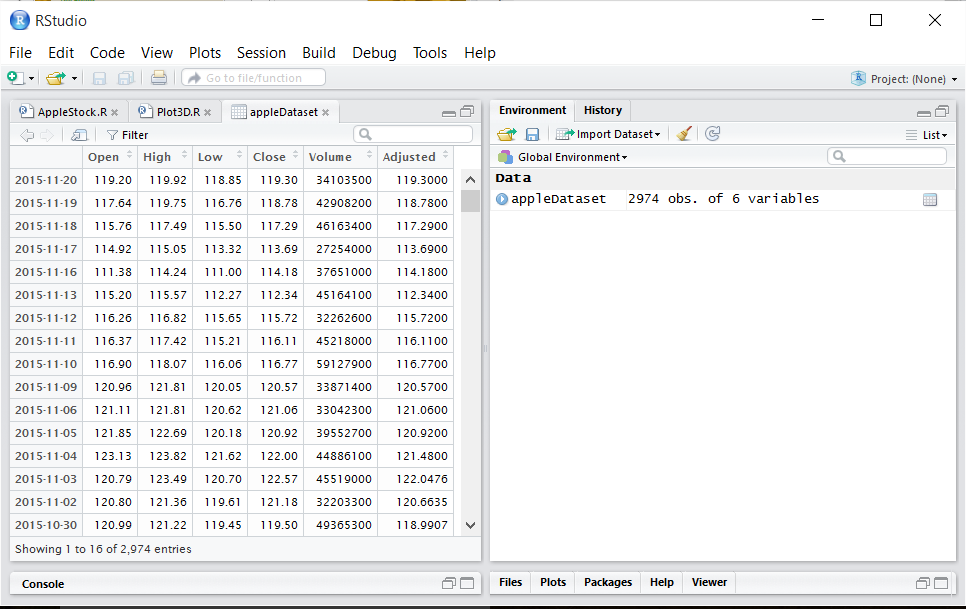
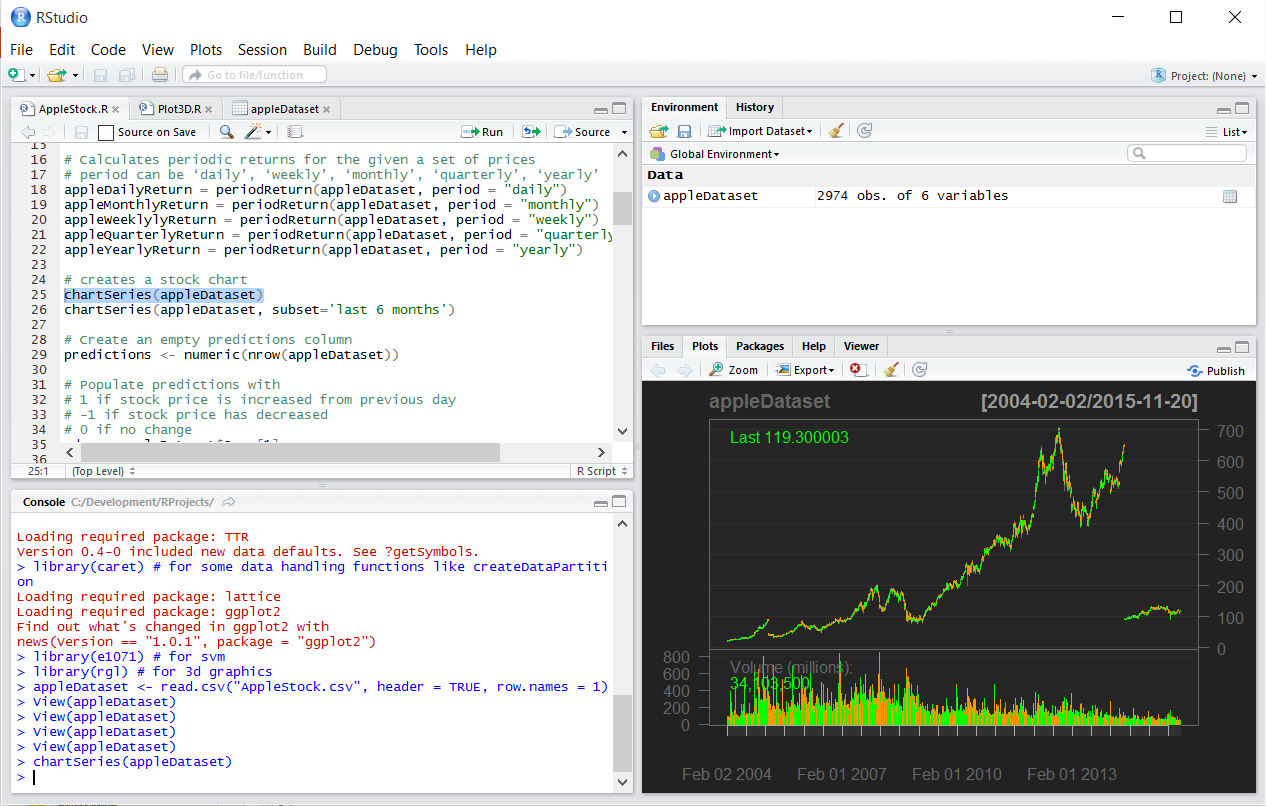
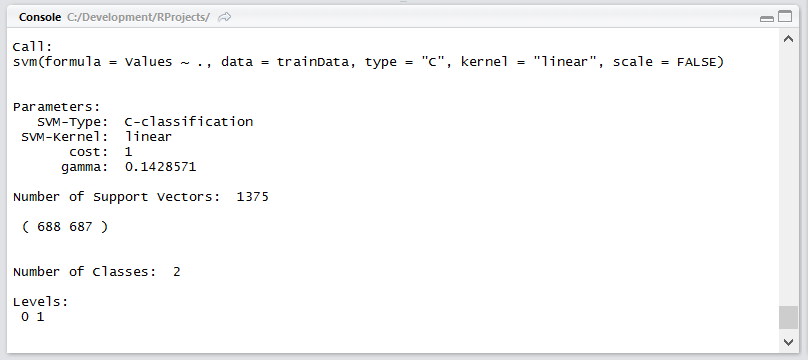


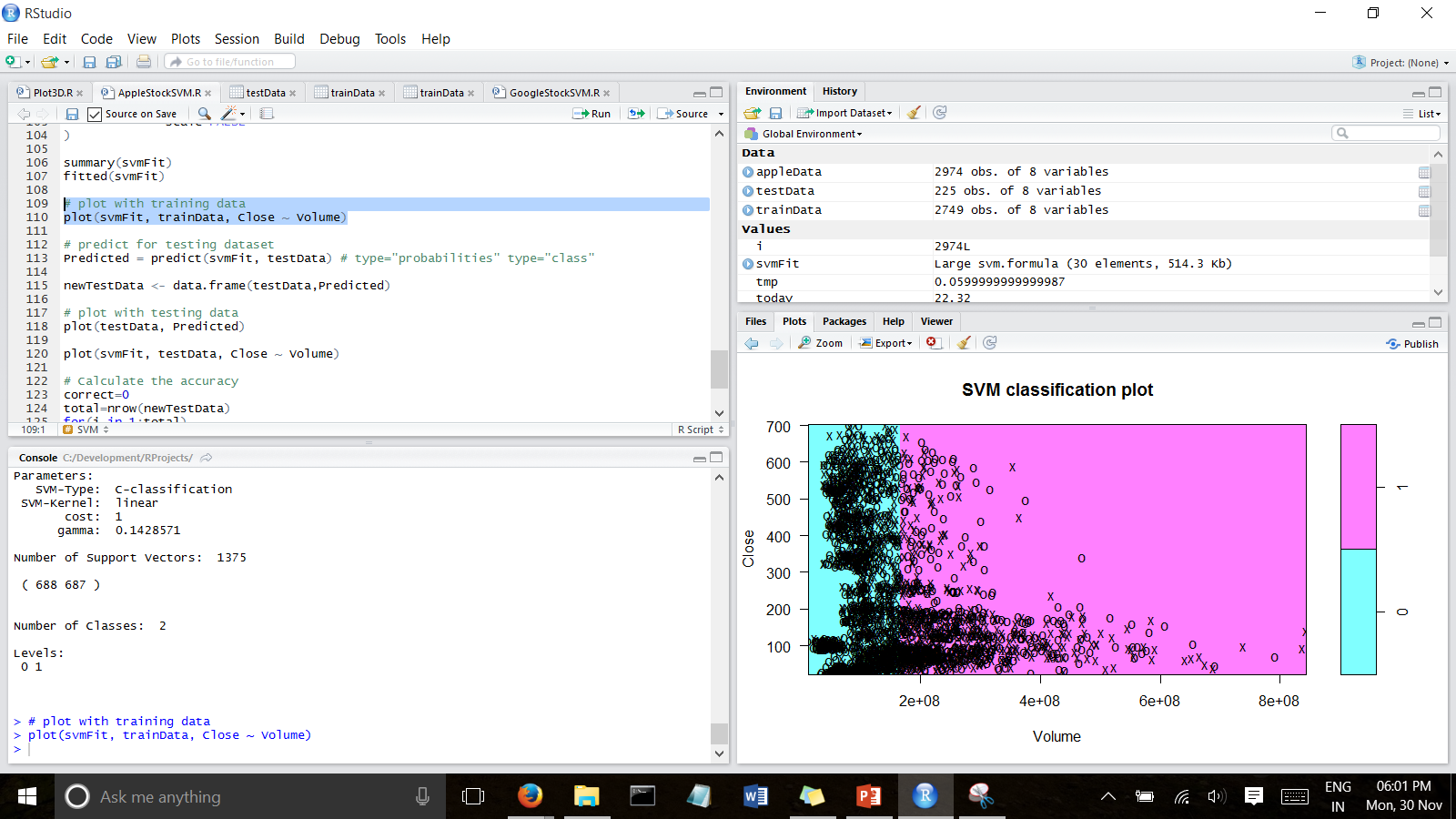
Chart series for Apple data



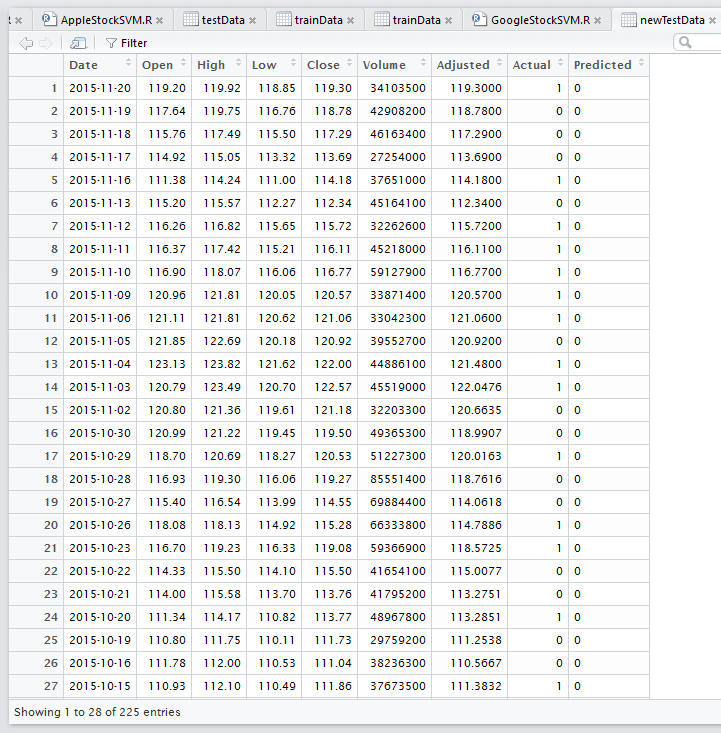
Output of summary of SVM fit for training data



Plotting the training data with SVM fit



Test data with Predicted and actual values column



Calculation of Accuracy

