**Advance Data Science/Architecture**

MID-TERM PROJECT

Submitted By:

Karan Millan Thevar

Pratik Patre

Rahul Chandra

(Team 3)

17th March, 2017

Introduction

The Project focuses on 2 major sections i.e Data wrangling and Building and evaluating models. In the first section the main focus is on Data retrieval and deriving a summary matrix out of it. This process consists of programmatically login to the Single-family loan data using this url - <https://freddiemac.embs.com/FLoan/secure/login.php>. Here, the list of all the loan data present which are to be processed and analyzed. This analysis is then shown visually on Tableau. The second section of the project focusses on the prediction and classification models for the loan data. This is done by using the unsupervised learning process where the different regression models are used for variable selection. Using these variables, the four prediction algorithms are processes, i.e., Linear Regression, Random Forest, Neural Network models and KNN algorithms. Depending upon the MAE, RMS, MAPE for training and testing datasets of these algorithms the best algorithm is selected and further processes is done. The next part for this section is classification. Linear Regression, Random Forest, Neural Network and SVN algorithms are performed and depending on their ROC curve and Confusion matrices for training and testing datasets, the best classification model is selected. Using this model, a table is generated for all the historical data.

This all process is done taking in consideration the scenario of a company trying to analyze the US housing market. Thus, to channelize the process and make it easier for sharing the data, the whole process is done dockerize.

**Part 1: Data wrangling**

Data wrangling is loosely defined as the process of converting or mapping data from one "raw" form into another format that allows for more convenient consumption of the data with the help of automated tools. This section is done using Python coding.

**1.A Deliverables for this section**

* Docker image for the whole process.
* Instruction to run docker image to replicate the whole process
* Tableau dashboard which is shared through Tableau public

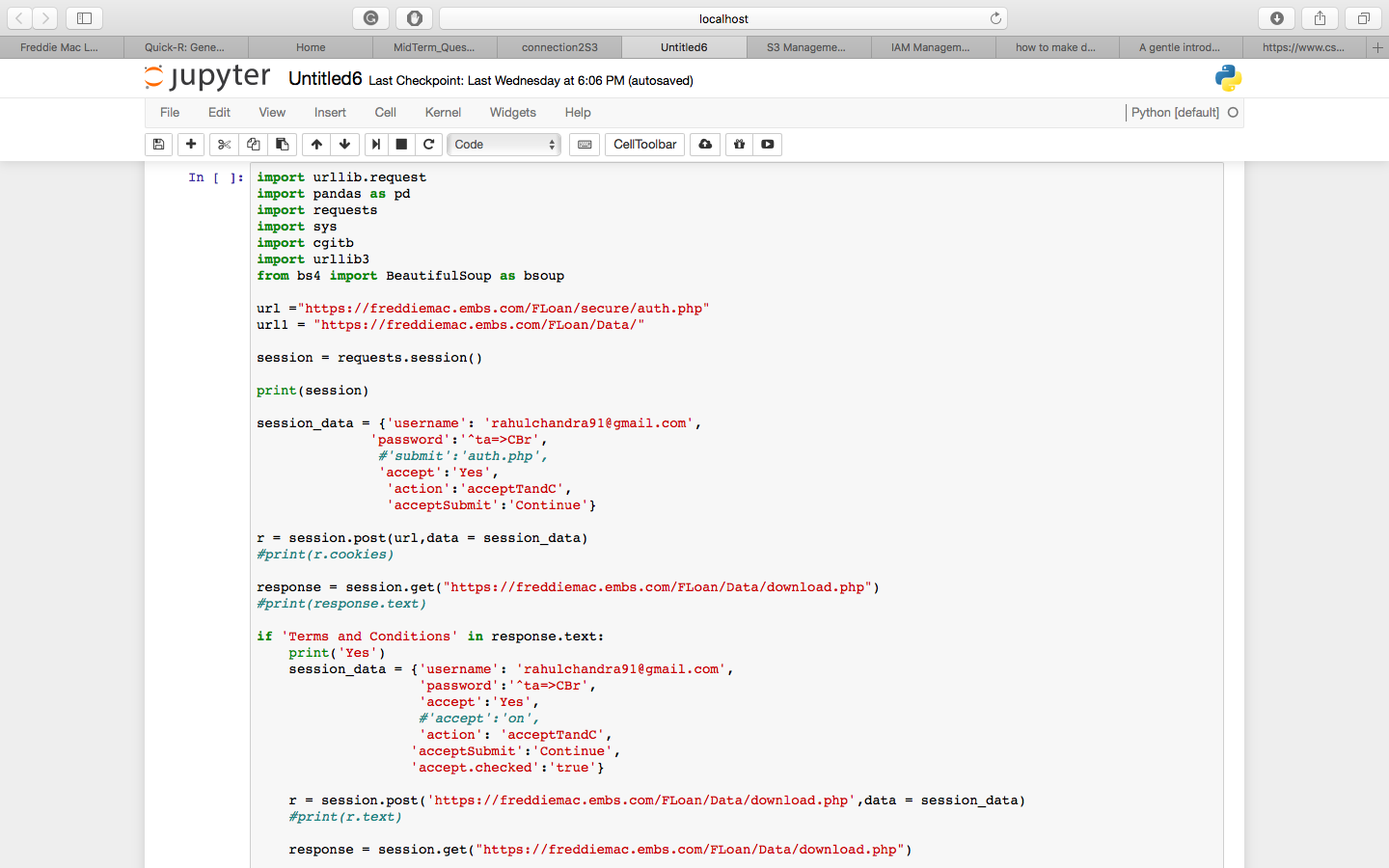
**1.B Process followed**

Steps followed to execute this section and generate the data programmatically without human inter-face.

**Step1: Login to the web-site**

The first part of this section is to programmatically Login to the Loan data url (<https://freddiemac.embs.com/FLoan/secure/login.php> ). We have used python to complete this section. Libraries like requests and BeautifulSoup are used to perform this process. Session data is saved and then it is passed through the pages to get the target page’s XML.

Below is the code snap shot that is used to perform this process:



**Step 2: Scraping of Sample data and getting the origination file for all year to generate the summary matrix.**

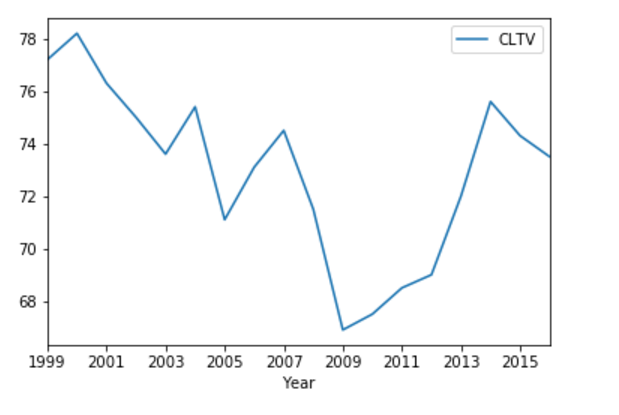
Scrapping through the response XML, sample data files are downloaded and then it is passed through function to get the summary matrix.

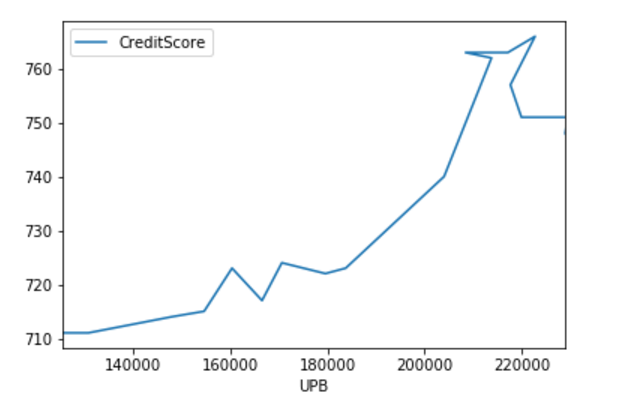
The summary matrix contains year wise classification of Avg(ORIGINAL DEBT-TO-INCOME (DTI) RATIO), Avg(COMBINED LOAN-TO-VALUE (CLTV)), Avg(Credit Score), Sum(UPB), Avg(UPB).

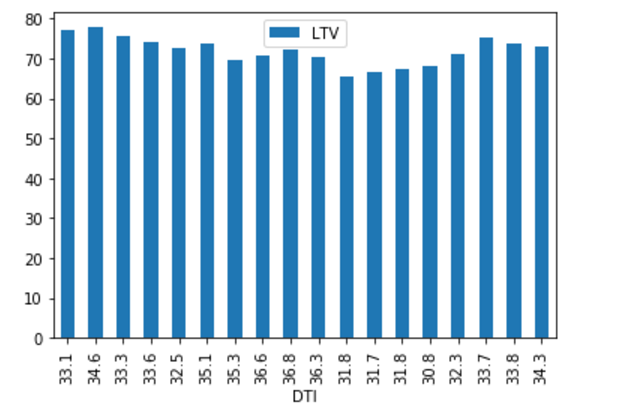
Below is the code snippet for the above:



Graphs for the above matrix was generated using MATPLOTLIB library in python:







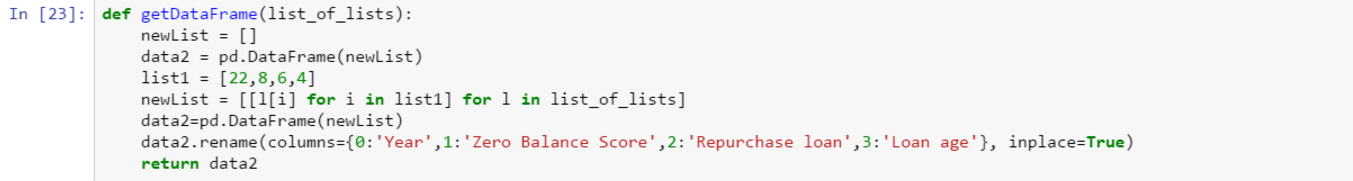
**Step 3: Scraping of Sample data and getting the Performance file for all year to generate the summary matrix**

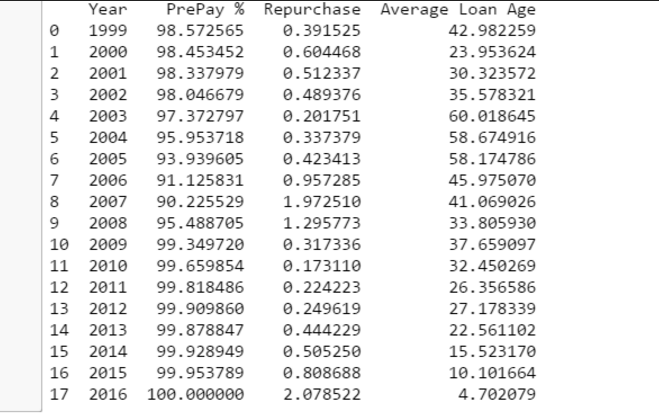
Scrapping through the response XML, sample data files are downloaded and then it is passed through function to get the summary matrix.

The summary matrix contains year wise classification of percentage (Prepay of loan(PrePay %), Percentage(Repurchase), Avg(Average Loan Age).

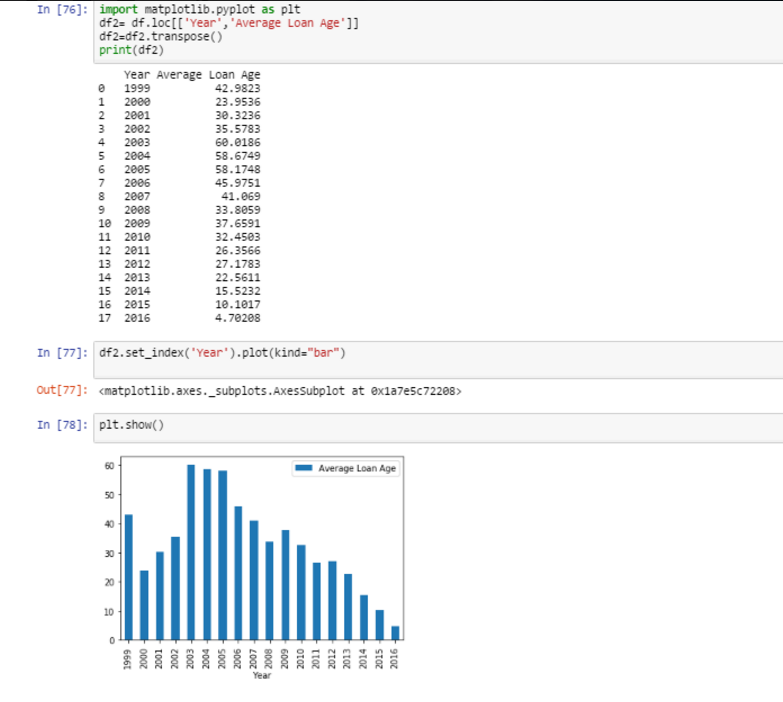
Below is the code snippet for the above:













**Step 4: Using the above algorithm for the origination data we generate the matrix table for Historical data of 2005**

Scrapping through the response XML, sample data files are downloaded and then it is passed through function to get the summary matrix.

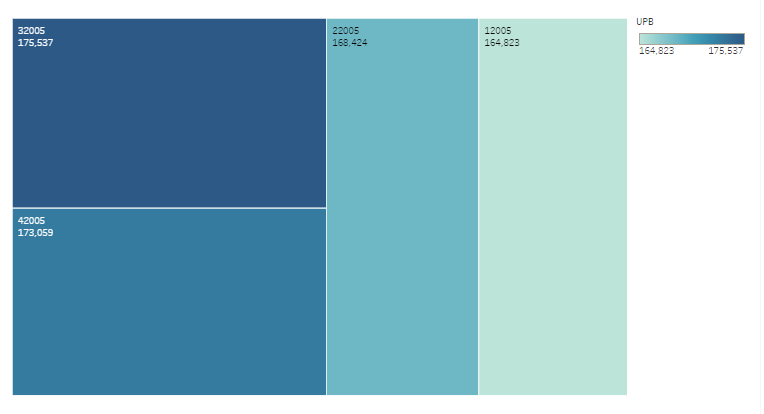
**Origination Classification Matrix**

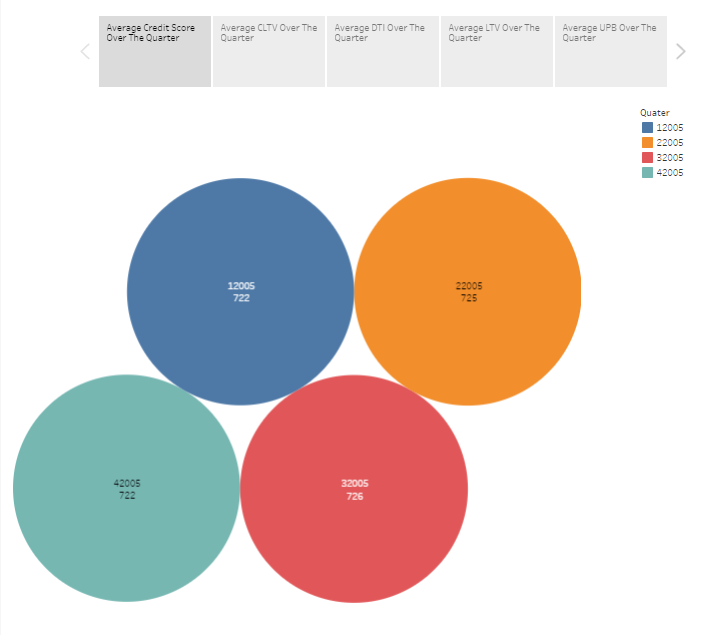
Below is the code snippet:



Tableau for the above matrix:

<https://public.tableau.com/profile/publish/ADS_2005_Origination_Summary/Story1#!/publish-confirm>





**Step 5: Analysis of the Origination Data and Generation of Risk Score for the Freddie Mac for 2005.**

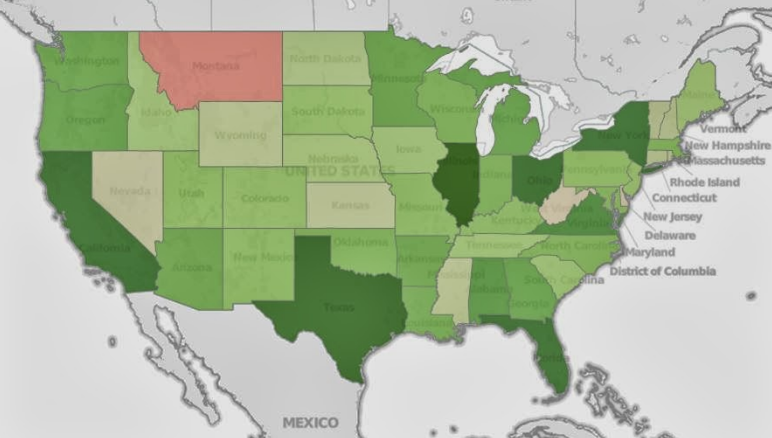
Freddie Mac being a home loan lender is always at risk of losing out on its loan. So, it is very important for the company to know its risk status. We have derived an algorithm by visual analysis on the attributes/columns that are defining factors for a particular loan. Thus, defining a risk score value to it. This is aggregated to state level where the risk score for a state is defined, for that particular Quarter.

The data taken are all the quarters of 2005 year from the historic data set. This is plotted onto the map of USA, showing the trend for the changing quarter values.

The outcome of this analysis is that it will give Freddie Mac a clear knowledge of the risk it has on recovering the loan from that particular state. This is a very vital information as through this, it can make many important policies to decrease the risk in a particular state, check its standing over the time frame, etc.

The code snippet for the same is given below:

The Tableau visualization is given below:



**1.C Conclusion**

This is a very interesting case study as the data set has a lot of interdependencies. Predictions and analysis on the data can be done easily with efficient results.

**Part 2: Building and evaluating models**

This section comprises of two parts Predictive analysis and Classification analysis. This outcome of the model will determine the best algorithm for prediction and classification. These trained model can be used for any data set to predict and classify its standings for the next quarter or so.

Predictive analytics encompasses a variety of statistical techniques from Linear Regression, KNN algorithm, Neural Network and Random Forest that analyze current and historical facts to make predictions about future or otherwise unknown events.

Classification is a general process related to [categorization](https://en.wikipedia.org/wiki/Categorization), the process in which ideas and objects are recognized, differentiated, and understood. The algorithms used to accomplish it are Logistic Regression, KNN algorithm, Neural Network and SVN algorithms.

**2.A Deliverables for this section**

* Docker file to get the input as the desired year to get the best predictive model and its outcome.
* Preforming variable selection on the historic data of Q12005 and determining the best combination for the same.
* Selecting the best algorithm based on the variables selected.
* Using the best found predictive algorithms get the output for the data of the specified years.
* Performing Classification on the historic data of Q12005 and determining the best classification model for the same.
* Using it to make the Delinquents table for each year.

**1.B Prediction - Part 1 (Variable selection)**

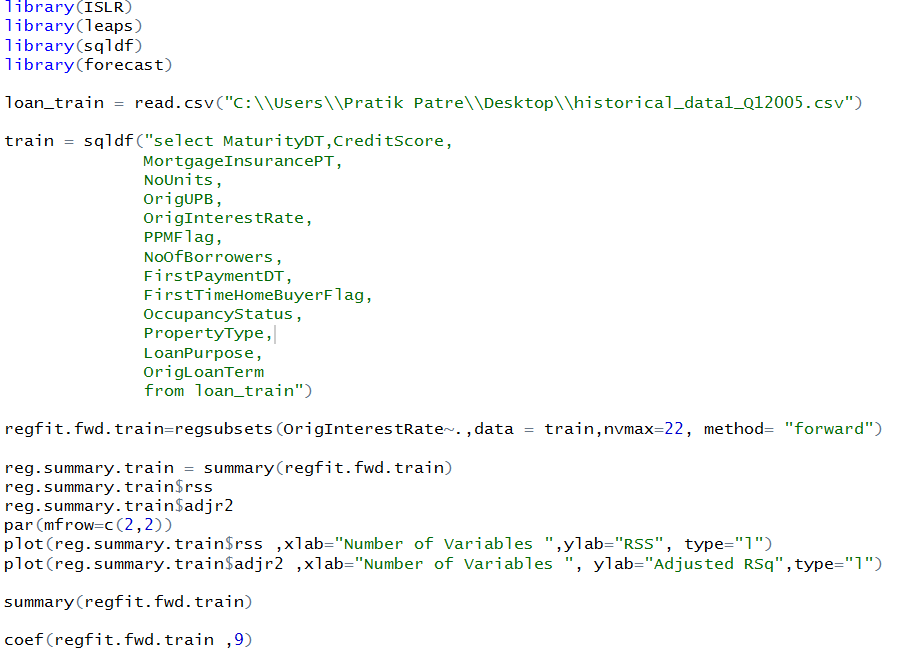
The process of determining the best possible combination of variables to get the minimal error of predictions is known as Variable selection. This can be done using the three types – Forward, Backward, Exhaustive.

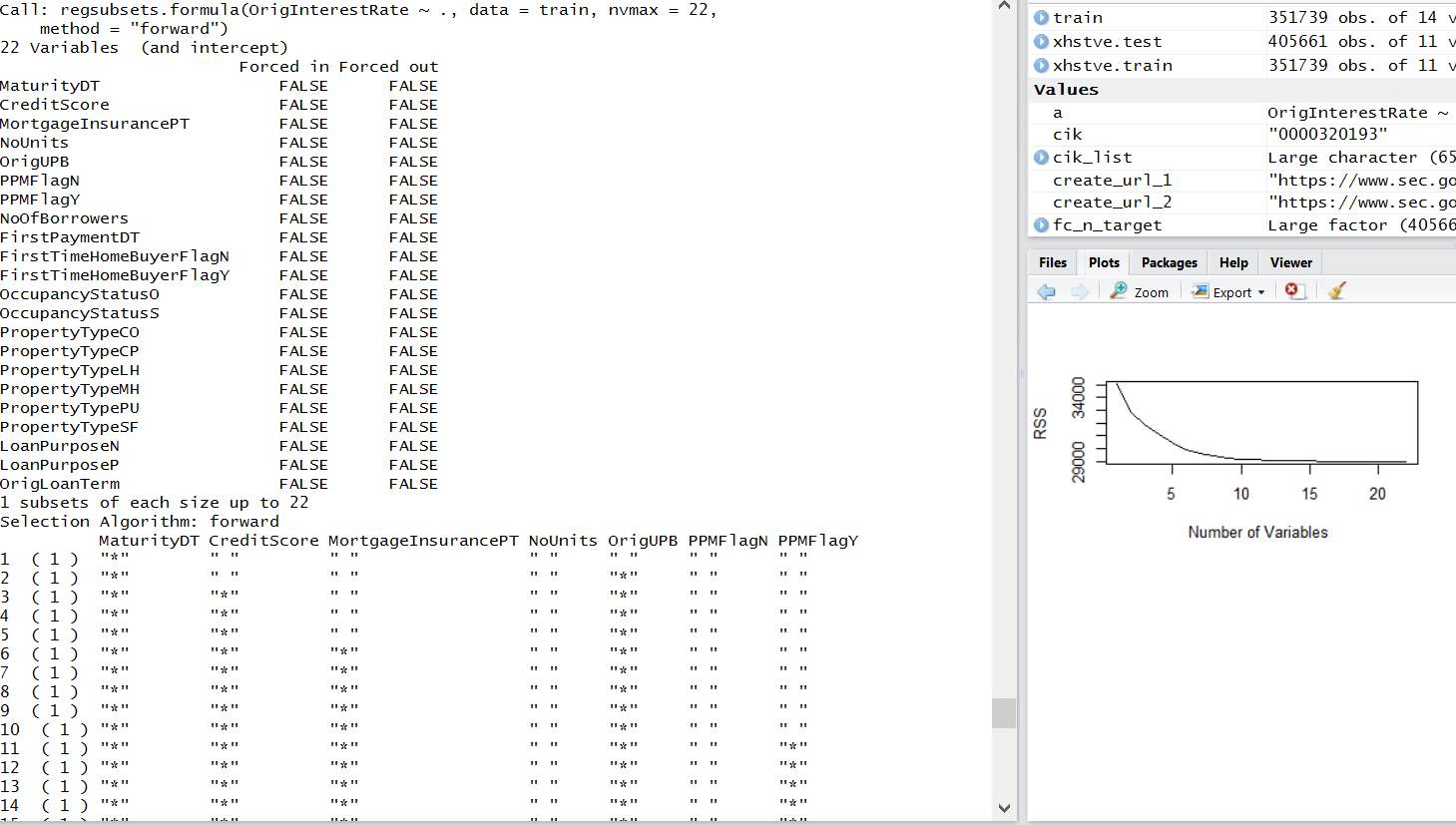
1. Forward Selection

Forward selection is done in R by using the Leaps, forecast library to determine the best possible combination.

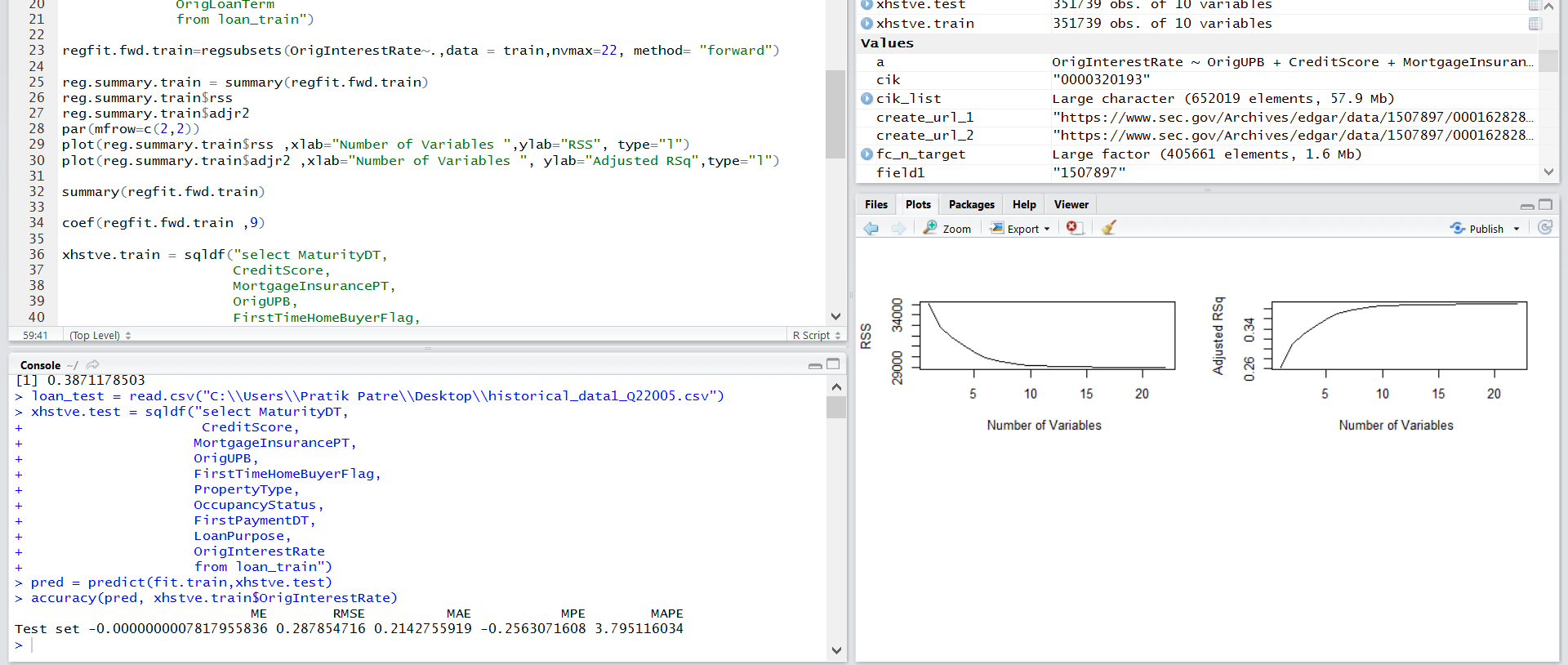
Using the regsubsets function we train the model to determine the best possible variable set. The method used in the function is "forward". Here, nvmax is set to 22 to get the most efficient selection.

The code snippet is attached below:





The RSS and AdjR2 graphs and the summary matrix from where variables are selected are displayed below:



1. Backward Selection

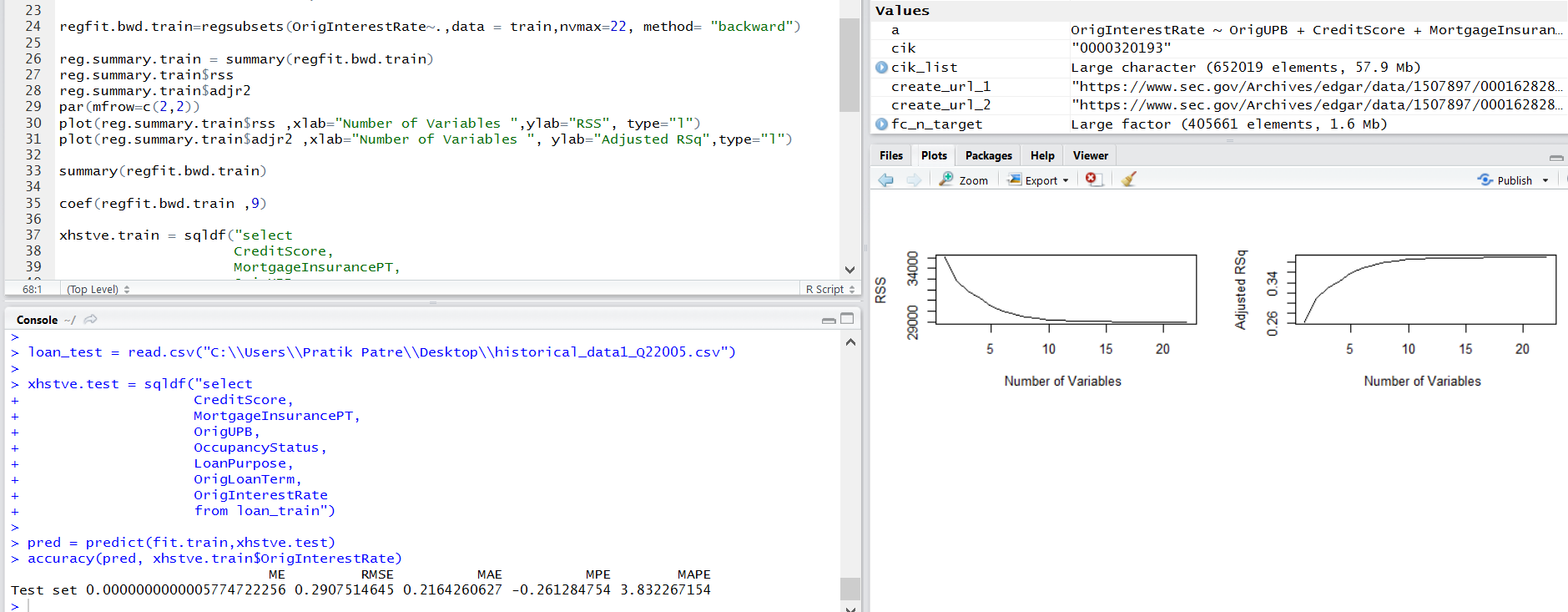
Backward selection is done in R by using the Leaps, forecast library to determine the best possible combination.

Using the regsubsets function we train the model to determine the best possible variable set. The method used in the function is "backward". Here, nvmax is set to 22 to get the most efficient selection.

The code snippet is attached below:



The RSS and AdjR2 graphs and the summary matrix from where variables are selected are displayed below:



1. Exhaustive Selection

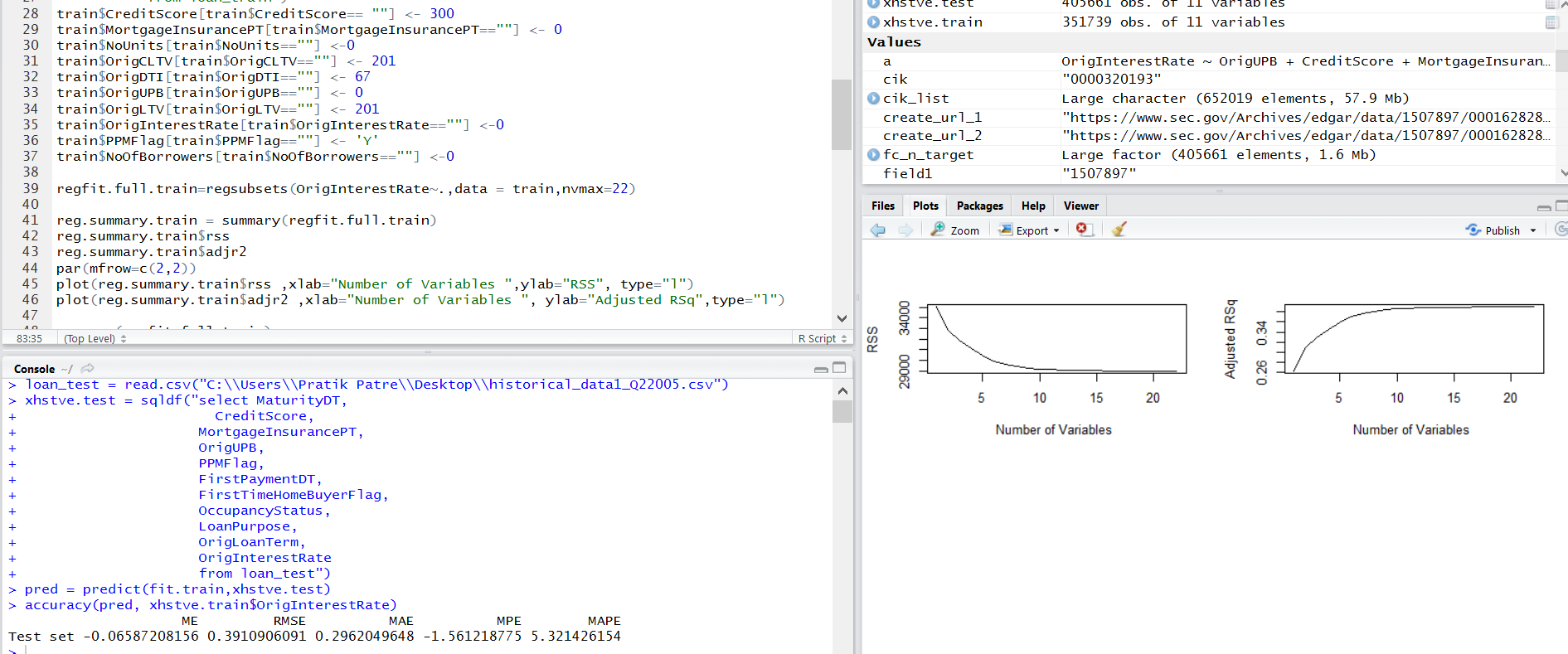
Exhaustive selection is done in R by using the Leaps, forecast library to determine the best possible combination.

Using the regsubsets function we train the model to determine the best possible variable set. The method used in the function is "exhaustive". Here, nvmax is set to 22 to get the most efficient selection.

The code snippet is attached below:



The RSS and AdjR2 graphs and the summary matrix from where variables are selected are displayed below:



**Determining the best Model**

After selecting variables from all the models, it is tested by running Linear regression on all the selected variables to get the RMSE from each models. After visual analysis and comparing the RMSE of each Forward model was selected the best out of all of them.

**Prediction - Part 2 (Algorithm selection)**

After getting the variable from Forward model we perform the different algorithms to compute the best algorithm.

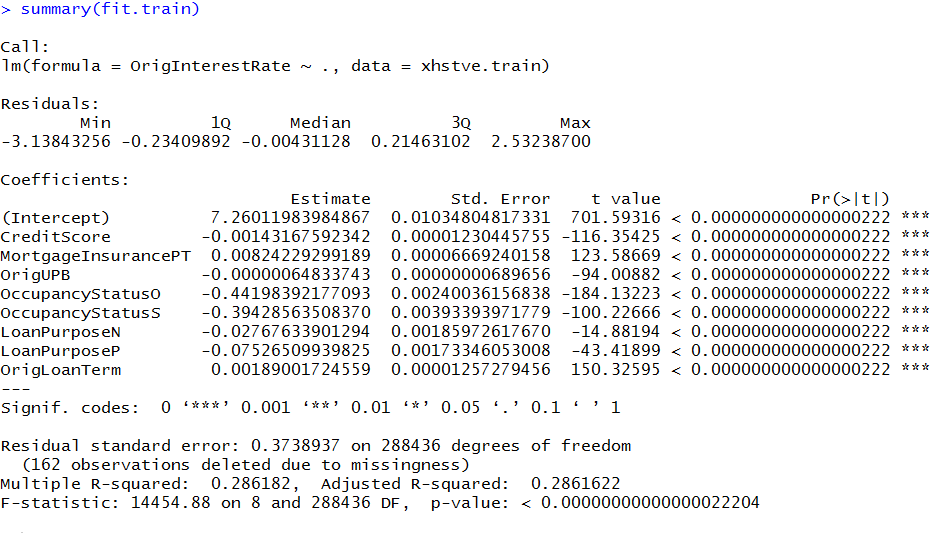
**Algorithm 1: Linear Regression**

In linear Regression the function used is lm() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



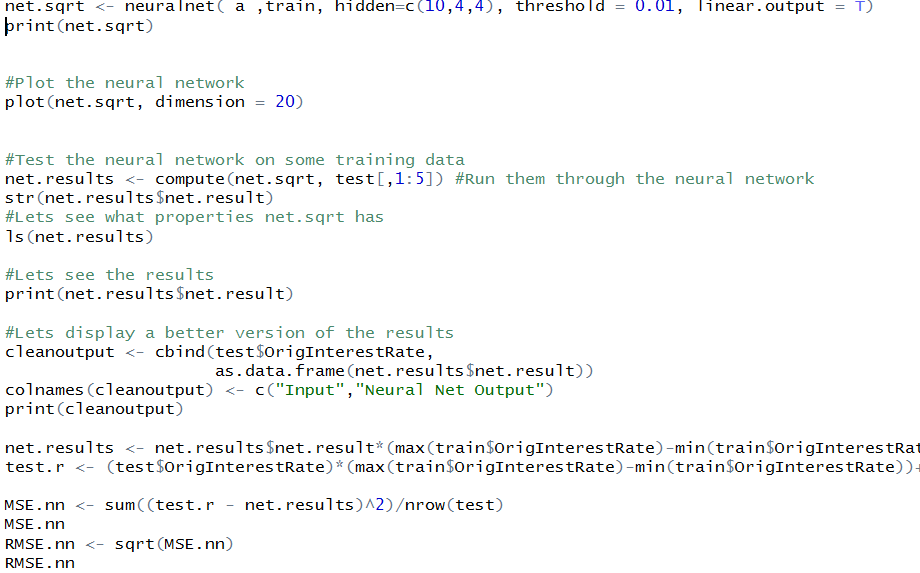
The Output is given below:



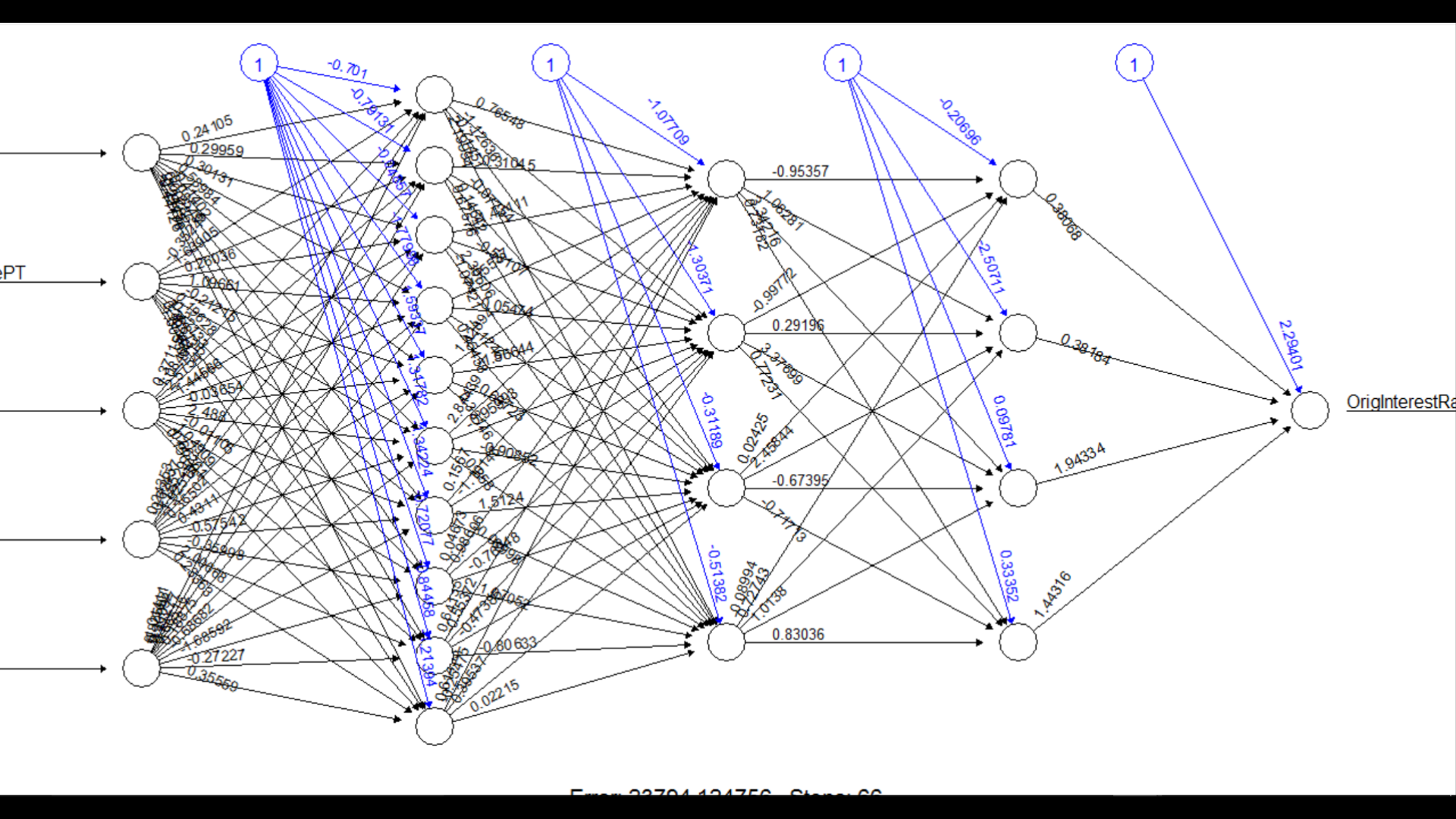
**Algorithm 2: Neural Network**

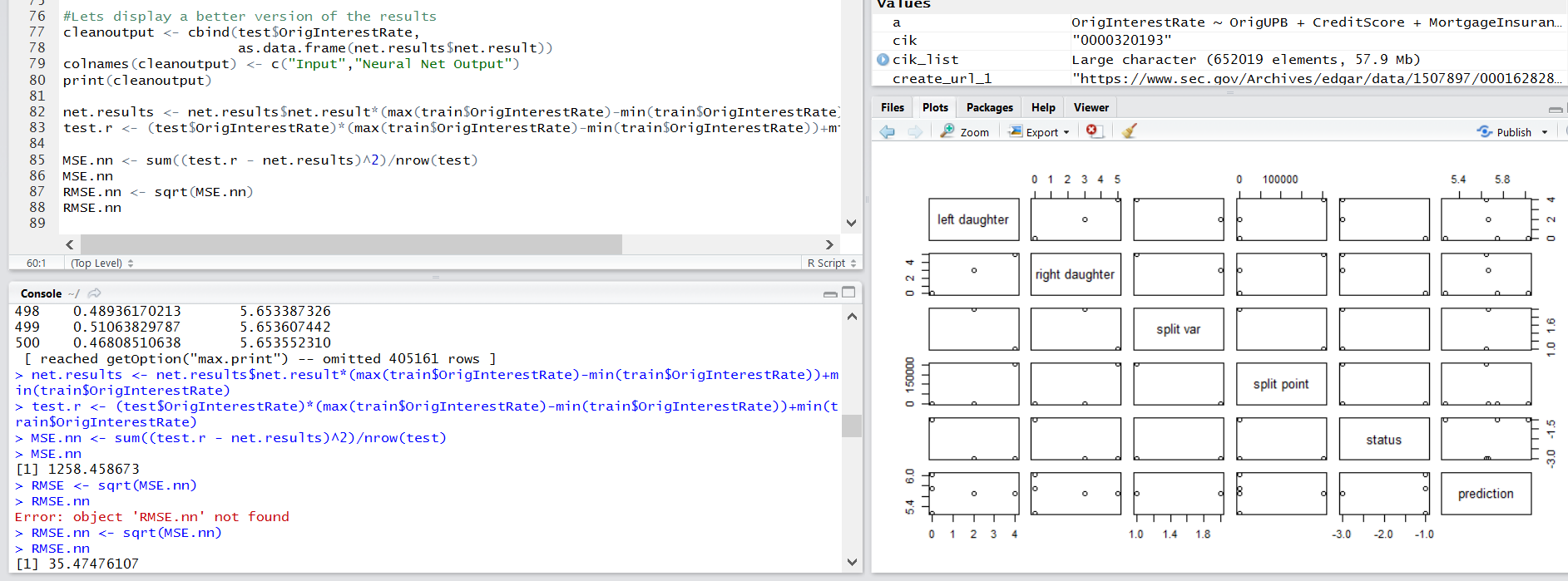
In neural network the function used is neuralnet() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



The Output is given below:

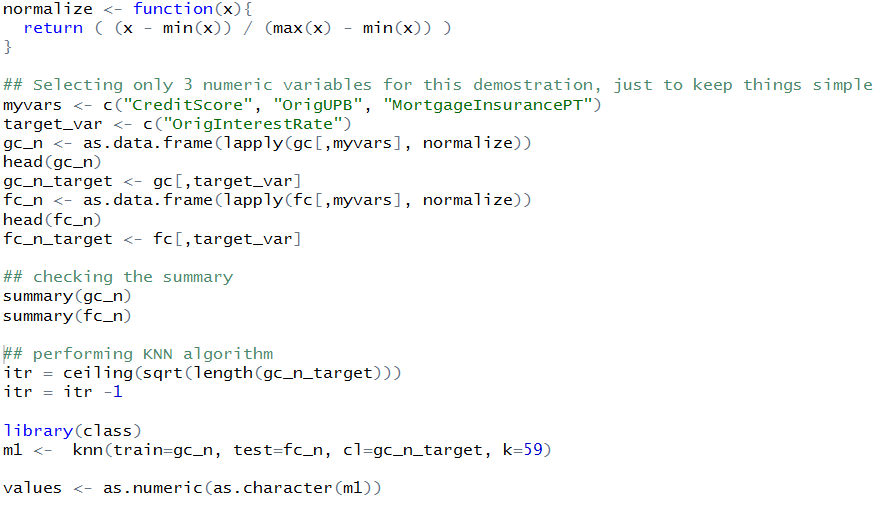




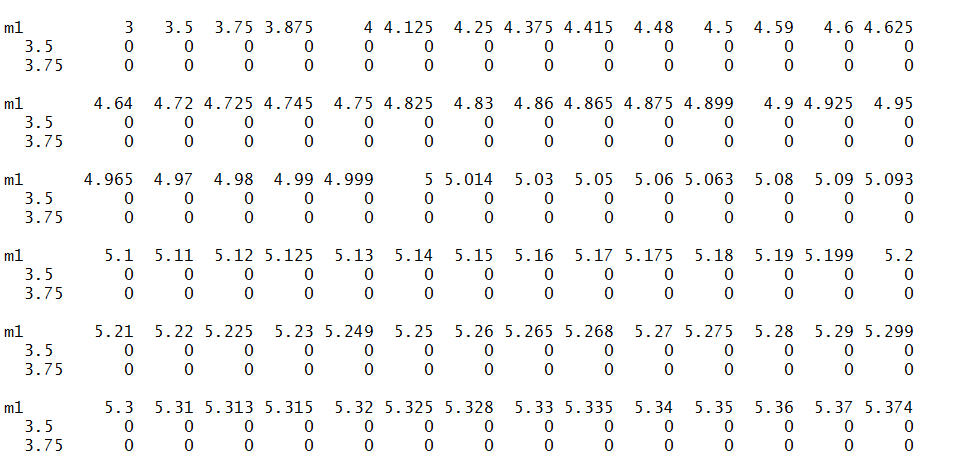
**Algorithm 3: KNN algorithm**

In KNN algorithm the function used is knn() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



The Output is given below:



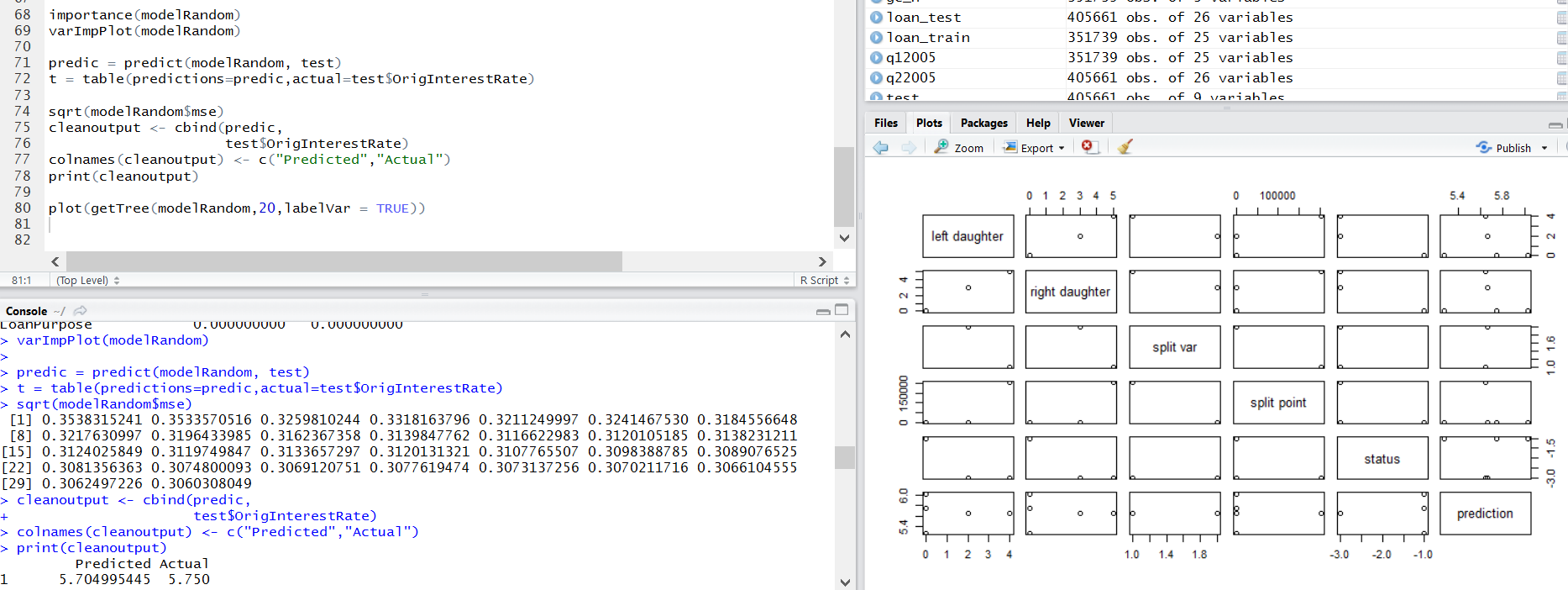
**Algorithm 4: Random Forest**

In Random Forest the function used is randomforest() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



The Output is given below:

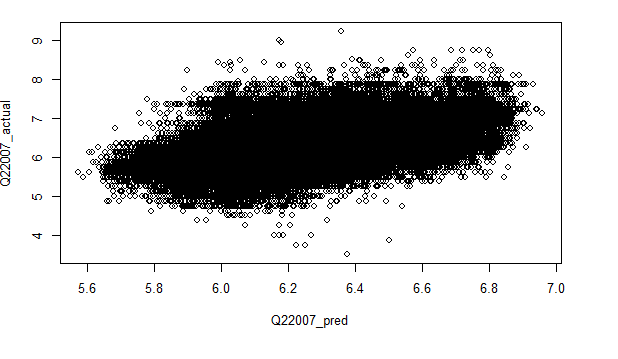


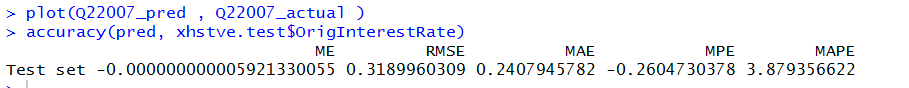
**Algorithm selection**

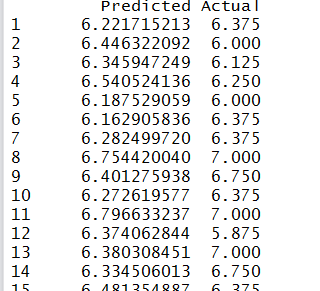
After cross validation on the algorithms on the basis of RMS value generated by each model. It was concluded that the RMS value of Linear Regression has the least value. Now on this the trained algorithm different data set are executed.

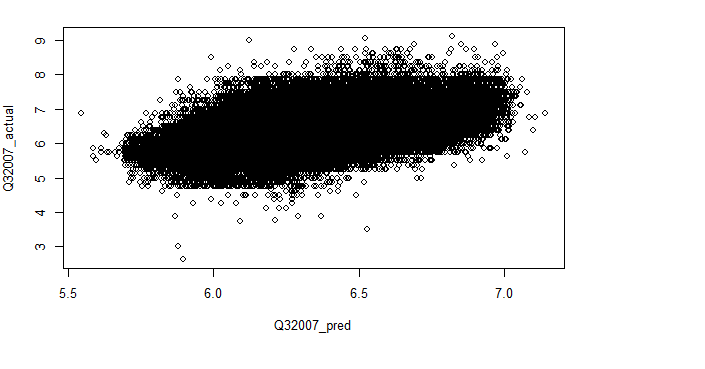
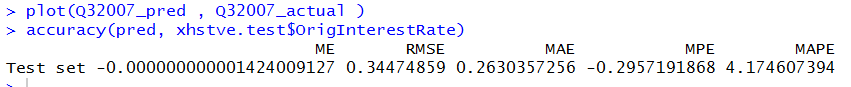
Below are the 2007 dataset that was required to run:

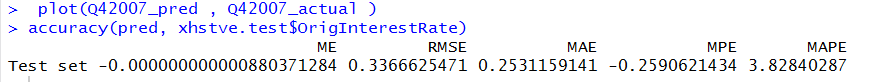
Q12007 used to predict Q22007

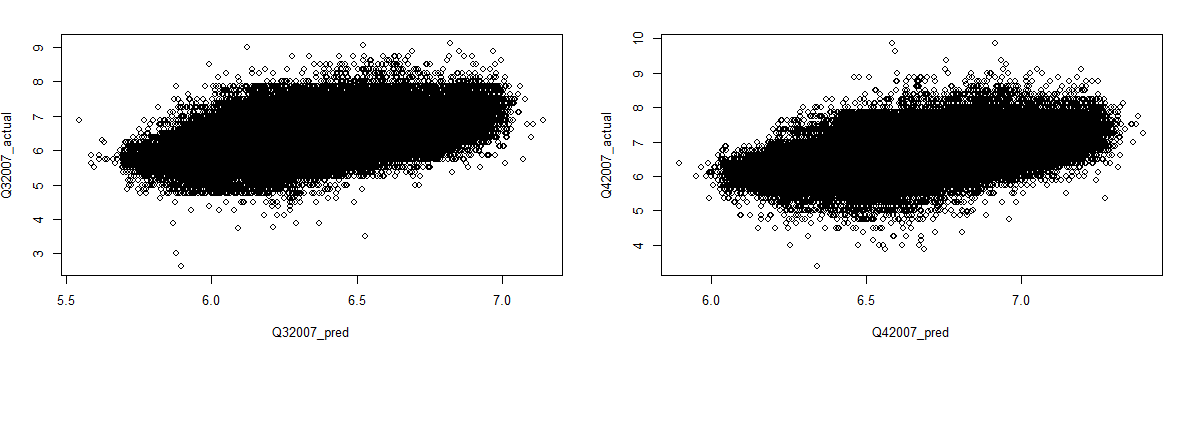


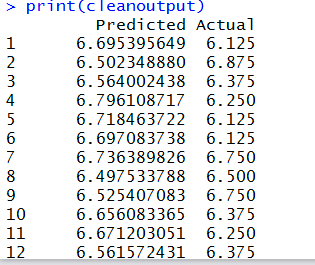


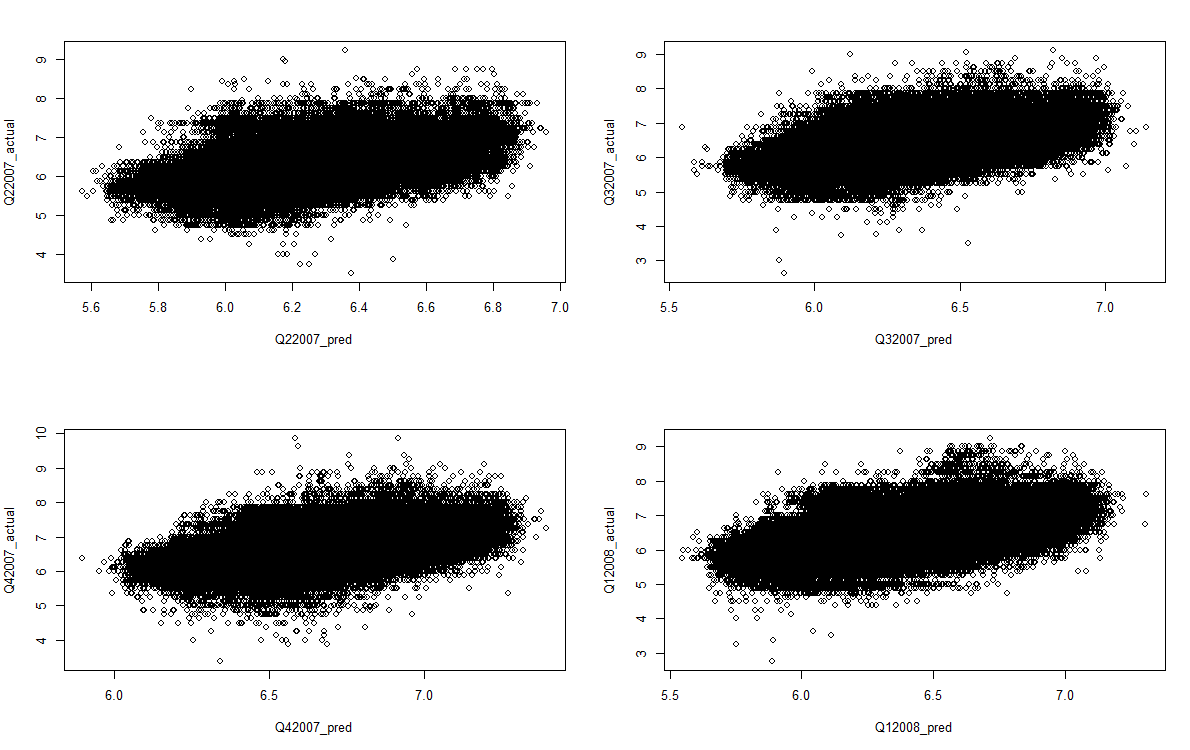


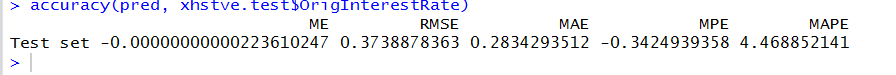




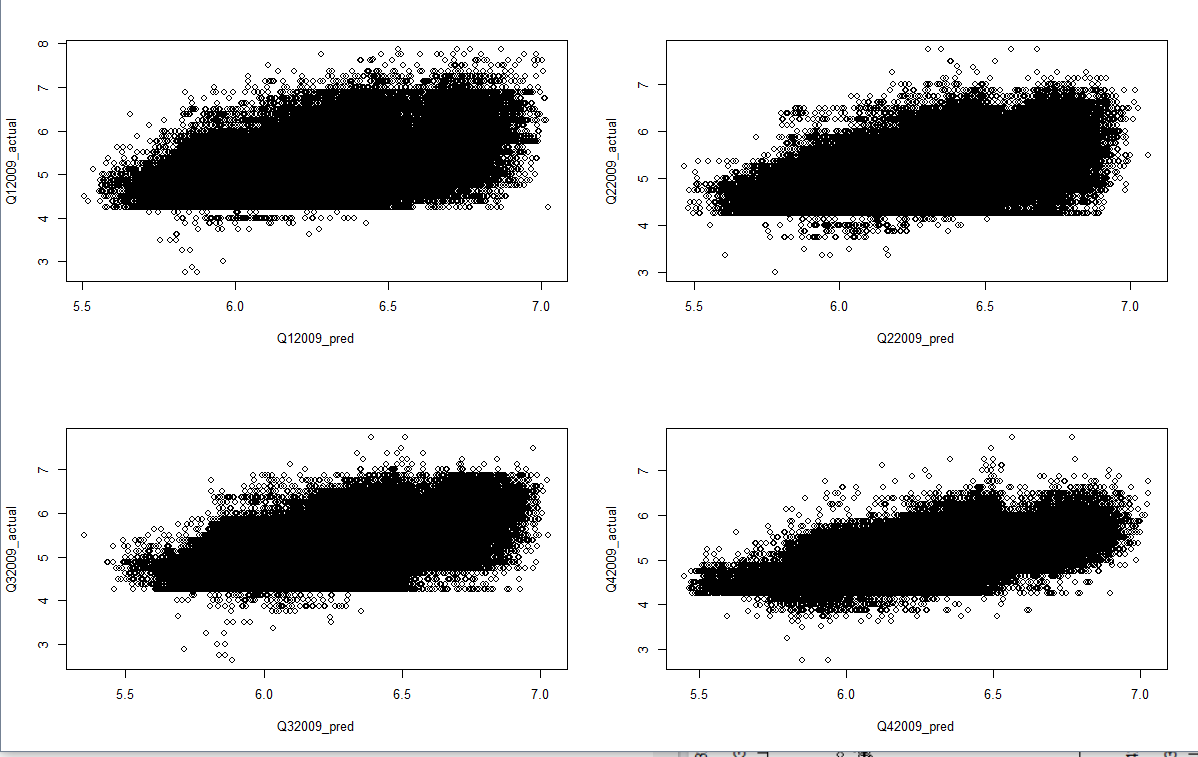






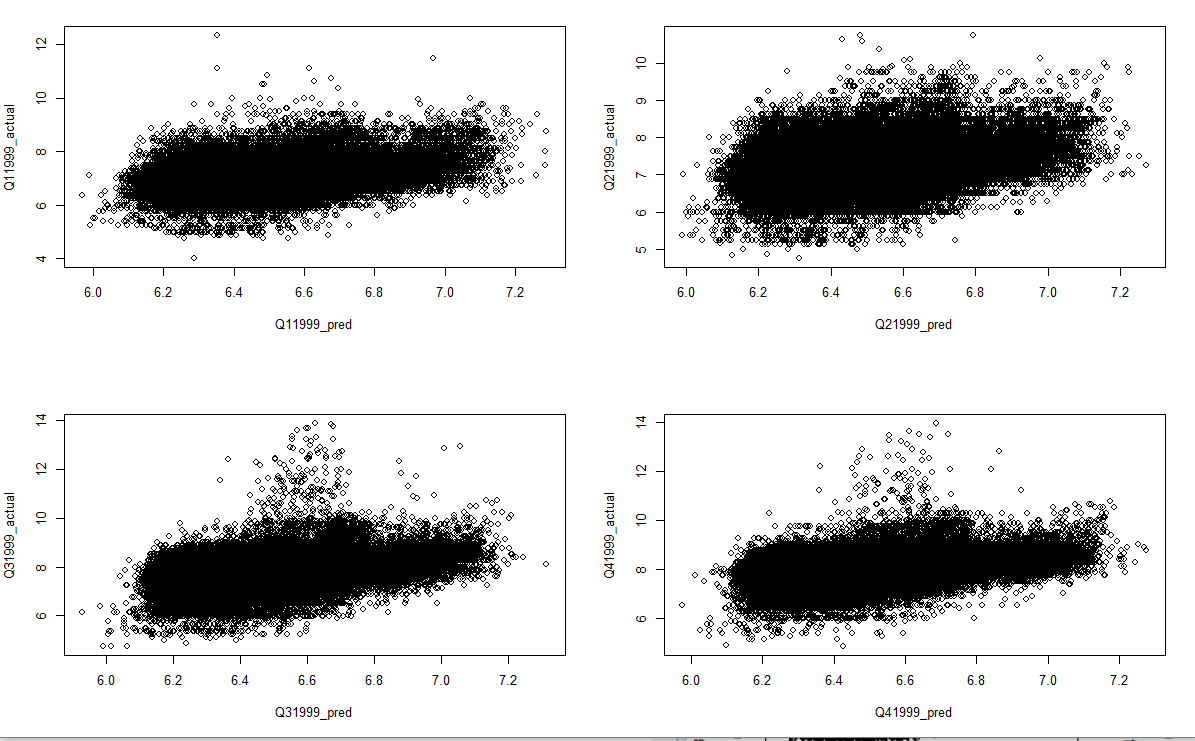


Predictions for 2009:





Economic Boom:





c Boom 1999

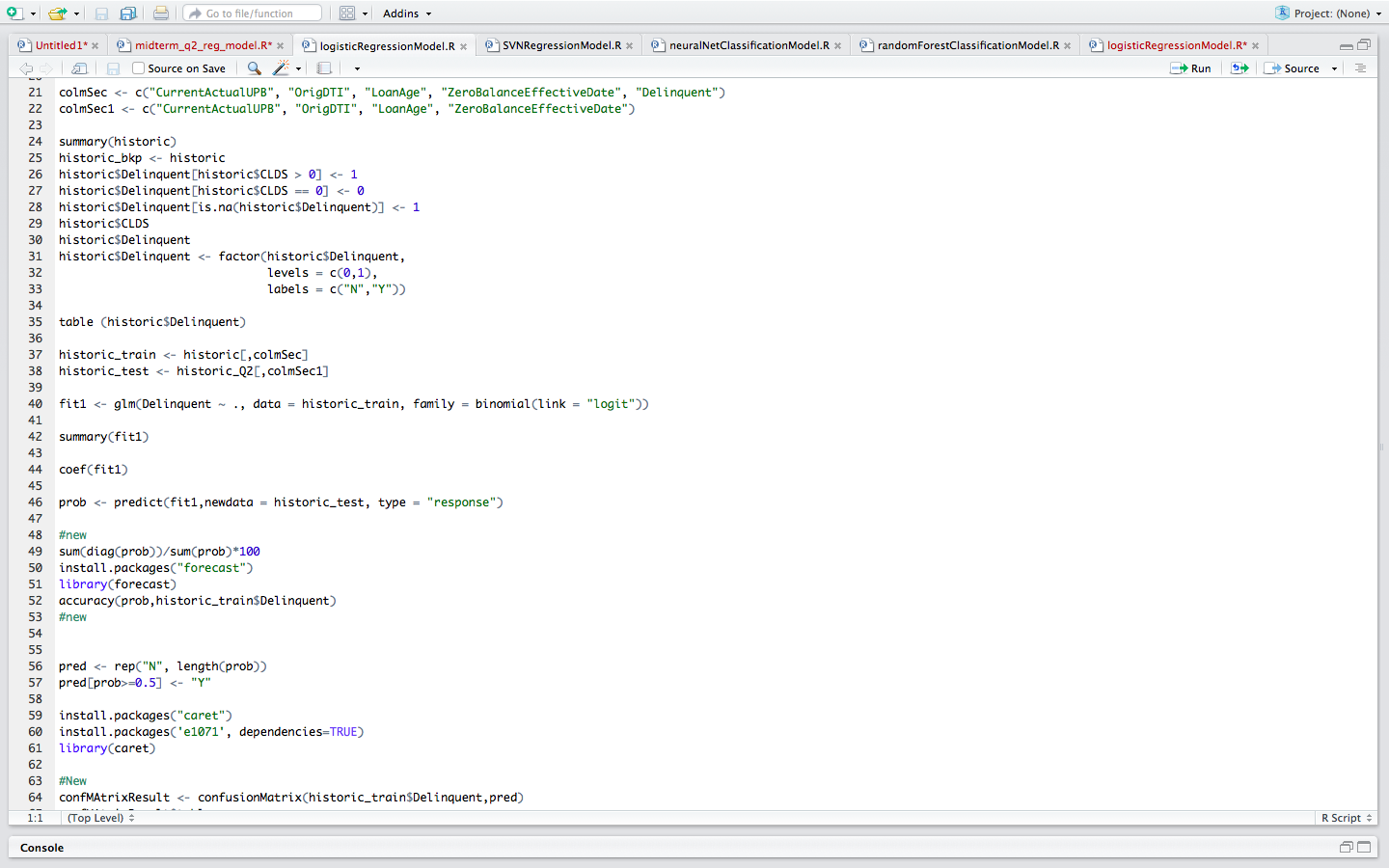
**1.B Classification - Part 1 (Algorithm selection)**

This part deals with classification of data that are required to make prediction on that class:

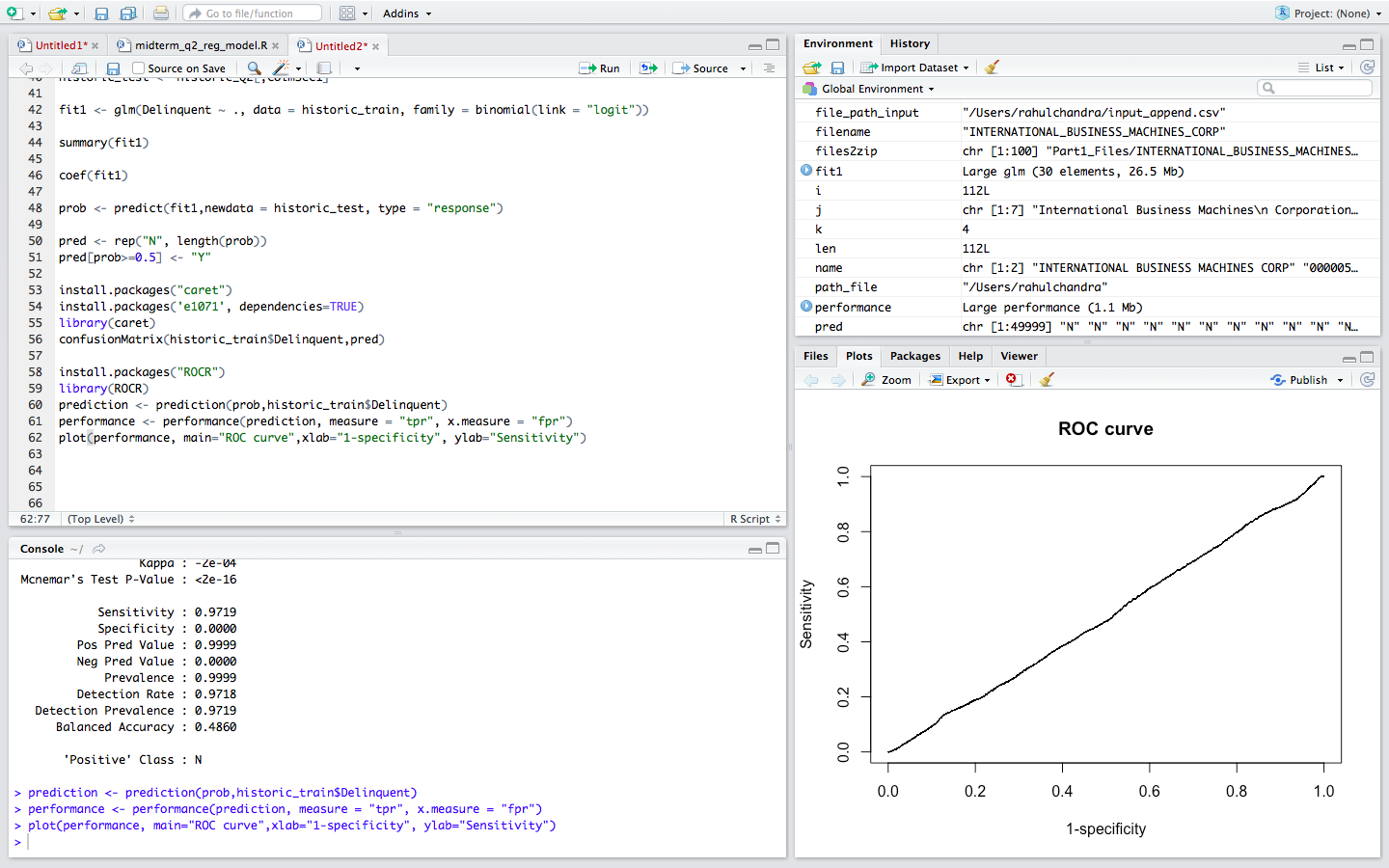
**Algorithm 1: Logistic Regression**

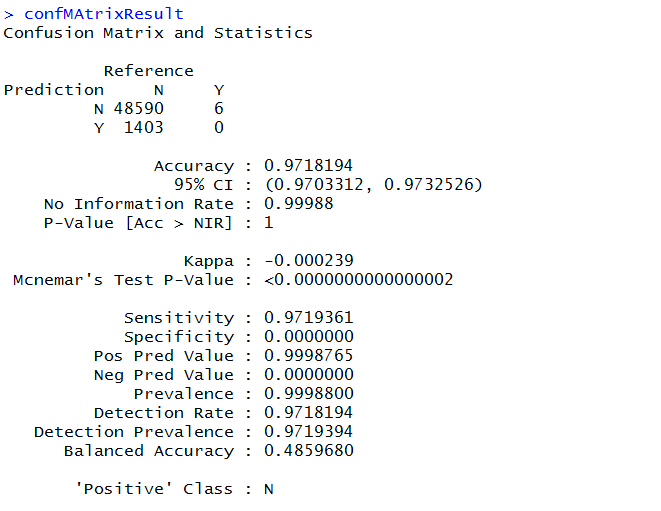
In Logistic Regression the function used is glm() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



The Output is given below:

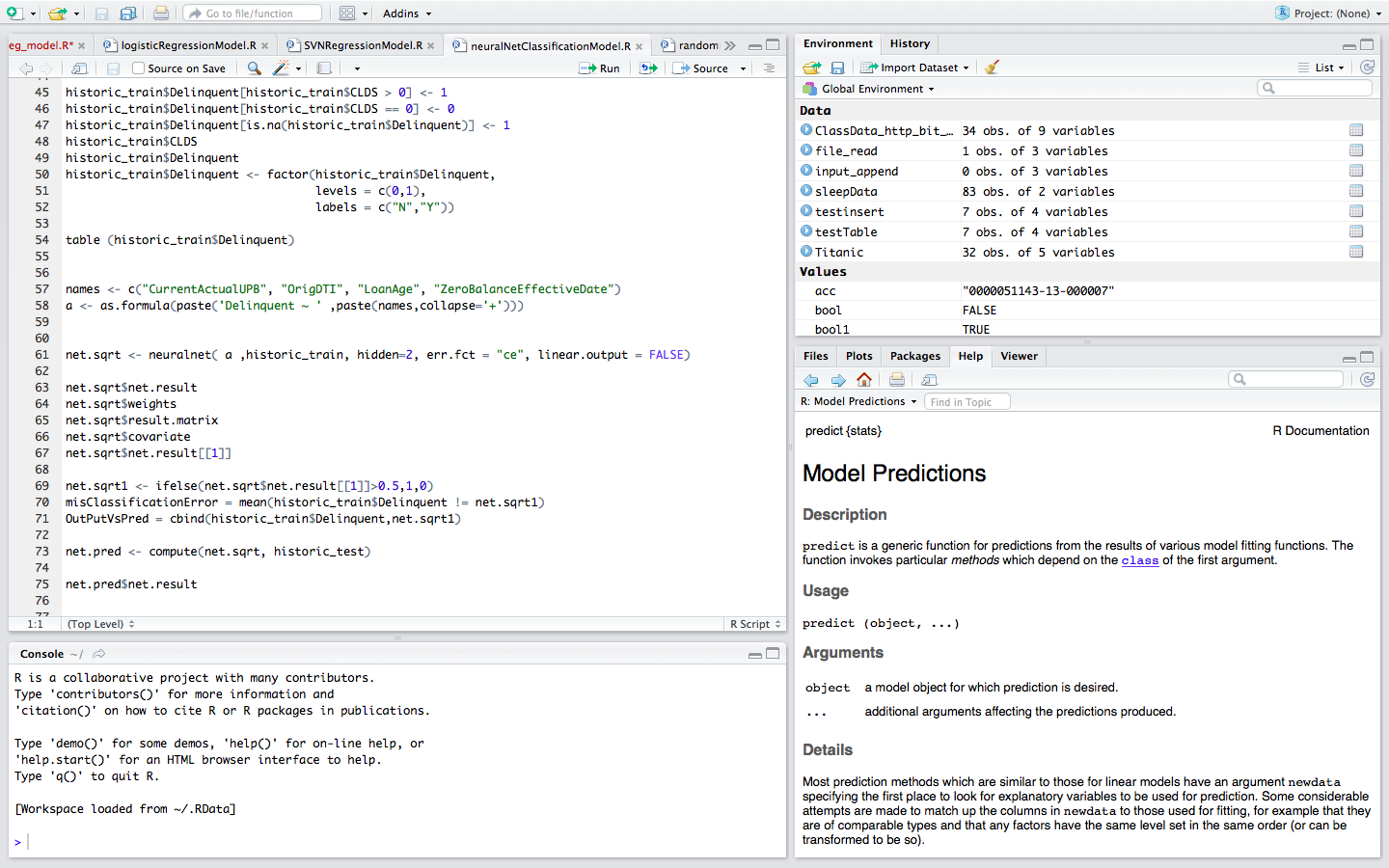




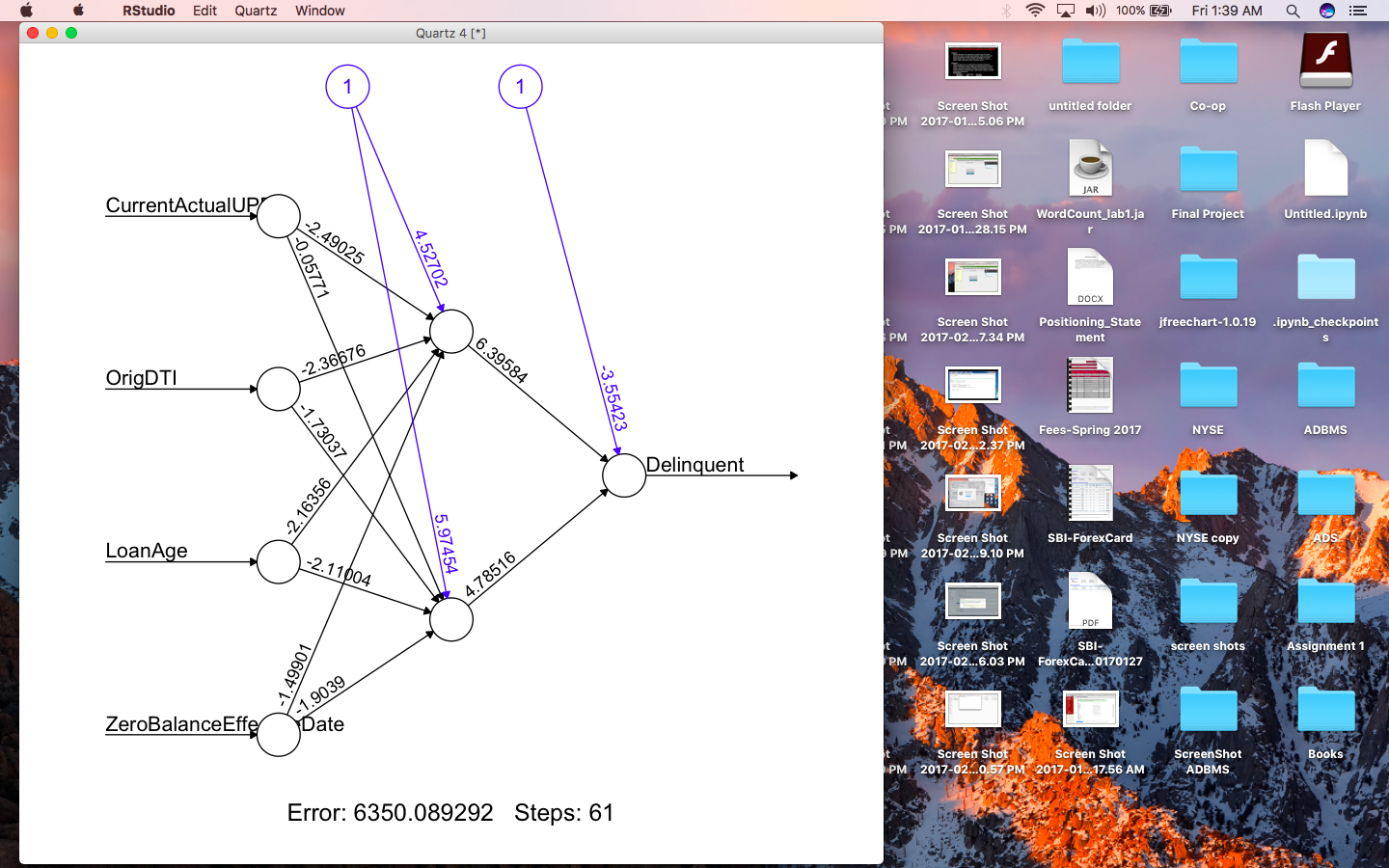
**Algorithm 2: Neural Network**

In neural network the function used is neuralnet() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



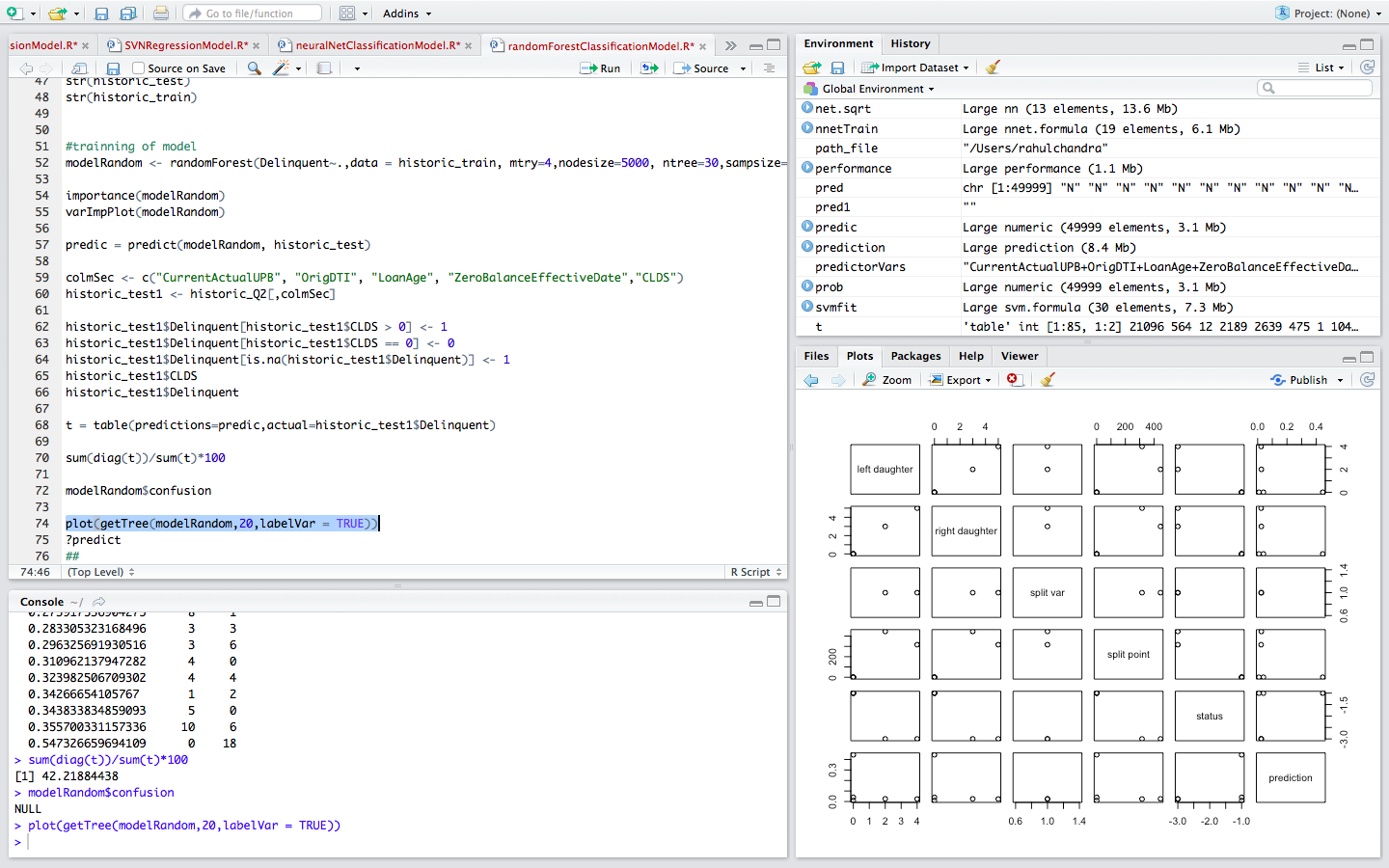
The Output is given below:



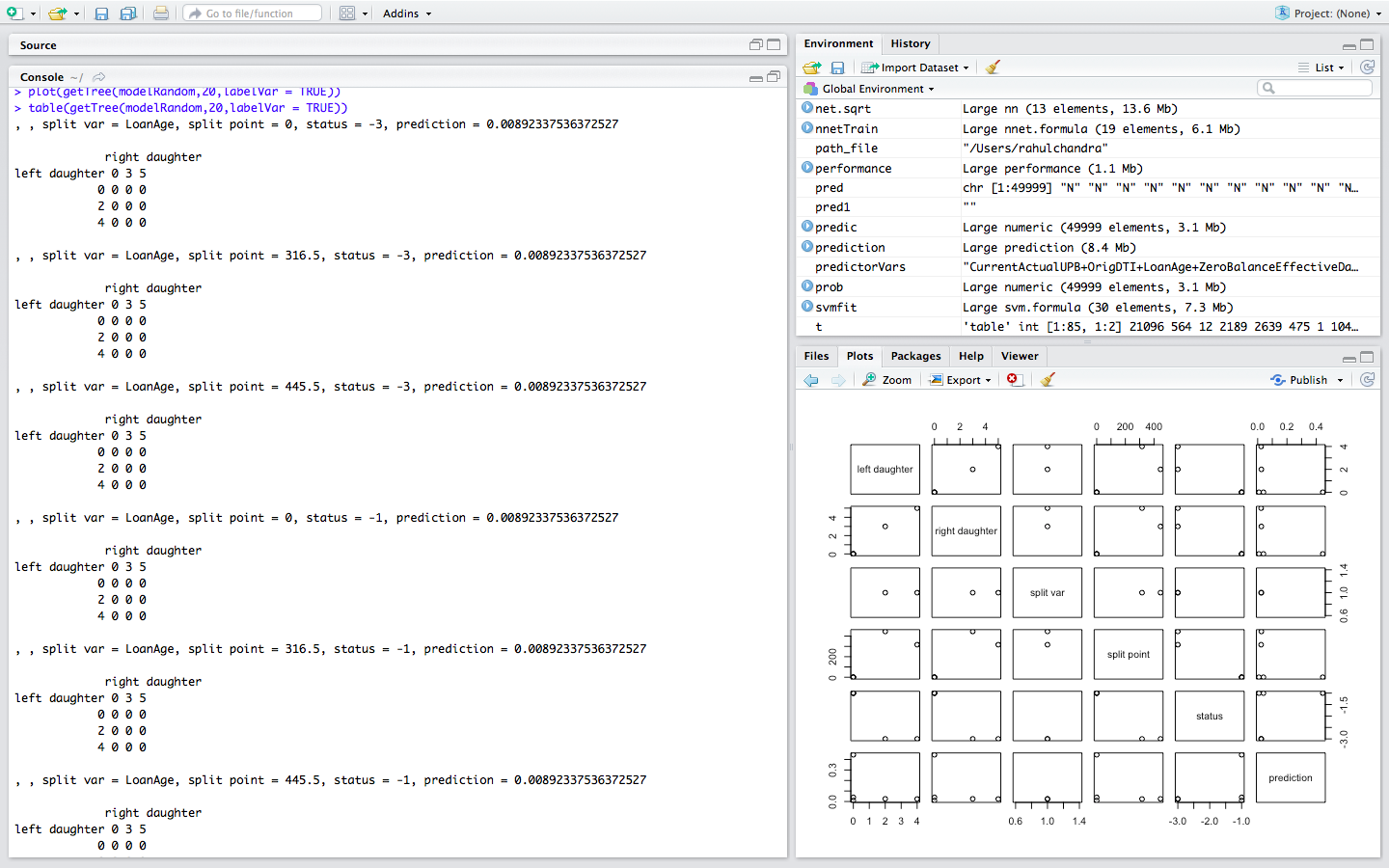
**Algorithm 4: Random Forest**

In Random Forest the function used is randomforest() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



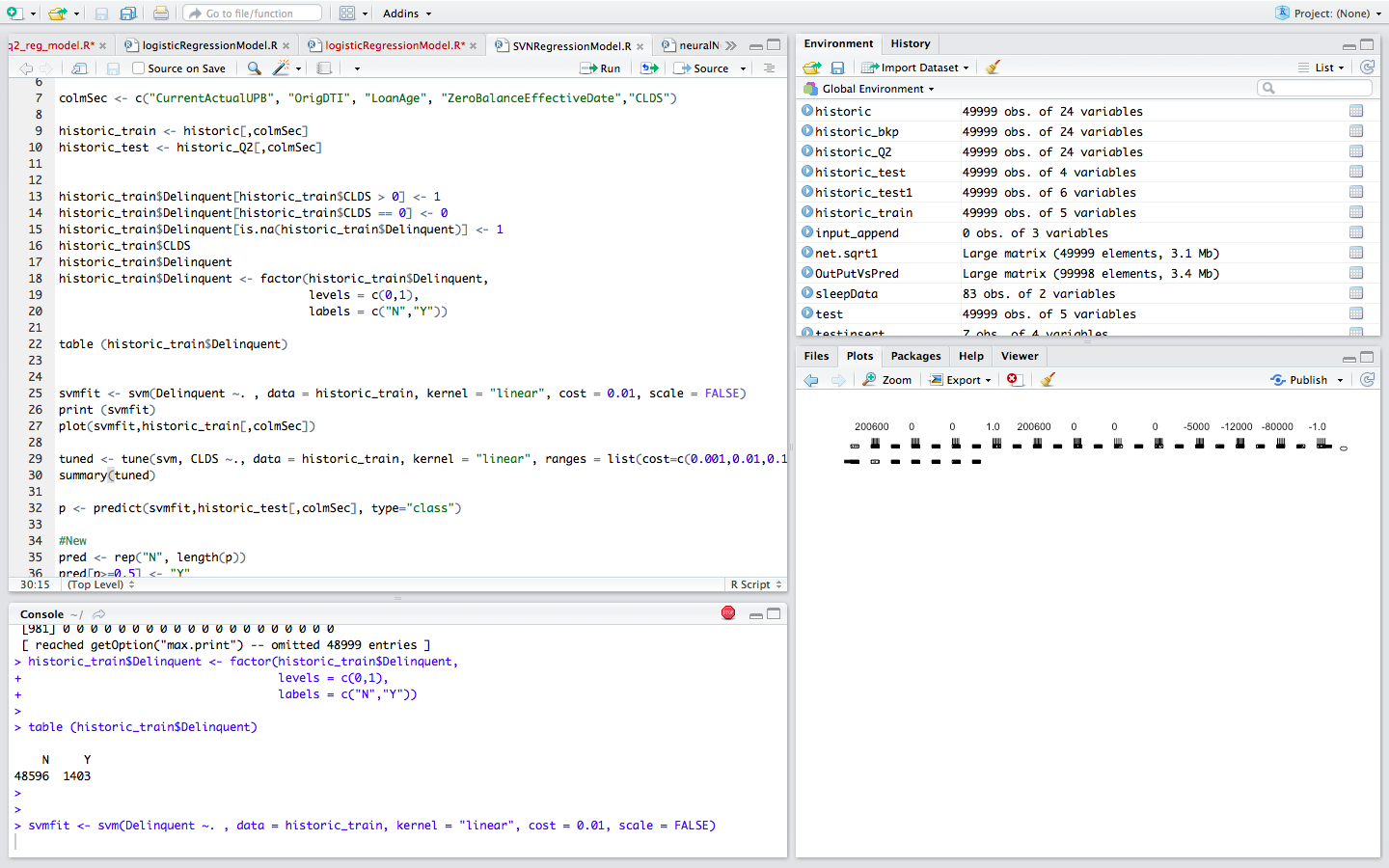
The Output is given below:



**Algorithm 3: SVN algorithm**

In SVN algorithm the function used is svn() to train and hence determine the output. In this the model is trained and then tested with testing data.

The code is given below:



2005 confusion matrix summary in tabular format:

