**Project Report**

**On**

**Predicting Application Rating of Google Play Store**

### Developed By: -

Astha Patel(15012121012) Parth Patel(15012121018) Kavan Patel(16012122004)

### Guided By:-

Prof. Umang Shukla (Internal)

Mr. XYZ (External)

**Submitted to**

**Department of Computer Science & Engineering Institute of Computer Technology**

****

**Year: 2020**



# CERTIFICATE

This is to certify that the Application Development Project work entitled **“Predicting Application Rating of Google Play Store”** by Astha Patel(Enrolment No.15012121012), Parth Patel(Enrolment No.15012121018) and Kavan Patel (EnrolmentNo.16012122004) of Ganpat University, towards the fulfillment of requirements of the degree of Bachelor of Technology – Computer Science and Engineering, carried out by them in the CSE(CBA/BDA/CS) Department. The results/findings contained in this Project have not been submitted in part or full to any other University / Institute for award of any other Degree/Diploma.

Name & Signature of IBM Mentor Name & Signature of Internal Guide

**Place:**

### Date:

**ACKNOWLEDGEMENT**

Application Development project is a golden opportunity for learning and self-development. I consider myself very lucky and honored to have so many wonderful people lead me through in completion of this project. First and foremost, I would like to thank **Dr. Hemal Shah,** Head of Department, Computer Science and Engineering, who gave us an opportunity to undertake this project. My grateful thanks to **Prof. abc & Mr.xyz (Internal & External Guides)** for their guidance in project work **Predicting Application Rating of Google Play Store**, who despite being extraordinarily busy with academics, took time out to hear, guide and keep us on the correct path. We do not know where would have been without his/her help. CSE department monitored our progress and arranged all facilities to make life easier. We choose this moment to acknowledge their contribution gratefully.

### PARTH PATEL (Enrollment No:15012121018)

**ABSTRACT**

A cursory glance at the newspaper reveals the fact that every year, the ratio of the mobile application is rapidly rising. We strongly believe that Data Science can be used for good, that’s why we decided to make this contribution. We are considering Google Play Store data-set and with the help of this data-set we will make approximately prediction of application rating; after that, we will create UI based application, which helps Google Play Store mobile application developer while developing the application.

**INDEX**

### Title PageNo

**CHAPTER 2:PROJECT SCOPE 03-04**

**CHAPTER1:INTRODUCTION 01-02**

**CHAPTER 3: SOFTWARE ANDHARDWAREREQUIREMENT 05-06**

**CHAPTER 4:PROCESS MODEL 07-08**

[CHAPTER 5: PROJECT PLAN 09-10](#_TOC_250009)

* 1. [List ofMajorActivities 10](#_TOC_250008)
  2. [Estimated Time Durationin Days 10](#_TOC_250007)

[CHAPTER 6: IMPLEMENTATION DETAILS 11-5](#_TOC_250006)2

* 1. [FlowchartofImplementation 12](#_TOC_250005)
     1. [DataCollection 12](#_TOC_250004)
     2. [UnderstandingData 13](#_TOC_250003)
     3. [DataVisualization 13](#_TOC_250002)
        1. Understand Original ApplicationInformation

Data 13

6.1.3.1 Understand Original ApplicationReviews Data 18

* + 1. [Data Cleaning andSentimentAnalysis 19](#_TOC_250001)
       1. Data Cleaning of ApplicationInformationData 19
       2. Visualization of Application Information CleanData 33
       3. [Data Cleaning and Visualization of Original 37](#_TOC_250000)

Application Review Data

* + - 1. **Web Scraping of NewReviews 37**
      2. **Visualization of Scraped Reviews40**
      3. **Merging and Visualization of Original Application 41**

**Reviews Data and Scraped Reviews**

* + - 1. **Cleaning, Sentiment Analysis with Visualization of41**

**Merged Reviews**

* + - 1. **Merging Application Reviews and Information Data43**

**and its Visualization**

* + - 1. **Feature Selection in Final Clean Data46**

**6.1.4.10 Prediction of Rating on Clean Data47**

**6.1.4.11 Selection of Best Fit Model with Visualization 52**

**6.1.4.12 UI Creation 53**

**CHAPTER 7: CONCLUSION AND FUTURE WORK 54-55**

**CHAPTER8:REFERENCE 56-57**

**CHAPTER: 1 INTRODUCTION**

**CHAPTER 1 INTRODUCTION**

In today’s world, every people loves to use application in their mobile phones so the day-by-day ratio of the mobile application is increasing one of the best examples is Google play store where in July 2013 total number of the application was 1 million which hiked to 2.6 million in December 2019 [[1]](https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/). It is very helpful for us if we can know approximately rating of application before developing application; here is an example to support this point; first of all let’s consider that if you want to develop and publish your application so before developing it you will decide its properties like my application will be free and its category will be gaming after that you will predict rating on the bases of that properties so by this predicted rating you can decide that you should keep this same property for your new application or you need to change it to get a better rating from users.

By considering this above scenario we decided to make a UI based application which helps to prediction rating by taking few parameters in input; to make this application we have first scrap data from Google play store and then perform data cleaning operations on that; after getting clean data we have done feature selection and extraction; then in the last step we applied various model for prediction of rating and then select best fit model.

The main focus of this project is to utilize historical data and make effective use of this data in the development of successful application using various new technologies. Below is the list of the tools and technologies which we have used in thisproject:-

* Smarten for datacleaning
* Tableau for datavisualization
* Anaconda for web scraping (Python3.6)
* R Studio for rating prediction by creating various models in this we used (nameof libraries which we used) library (Rprogramming)
* Trello for project management

**CHAPTER: 2 PROJECT SCOPE**

### CHAPTER 2 PROJECT SCOPE

The project is limited to only Google Play Store because data which is considered for rating prediction is of Google Play Store.

# CHAPTER: 3 SOFTWARE AND HARDWARE REQUIREMENTS

### CHAPTER 3 SOFTWARE AND HARDWARE REQUIREMENTS

**Minimum Hardware Requirements**

|  |  |
| --- | --- |
| **Processor** | 2.0 GHz |
| **RAM** | 4GB |
| **HDD** | 40GB |

*Table 3.1 Minimum Hardware Requirements*

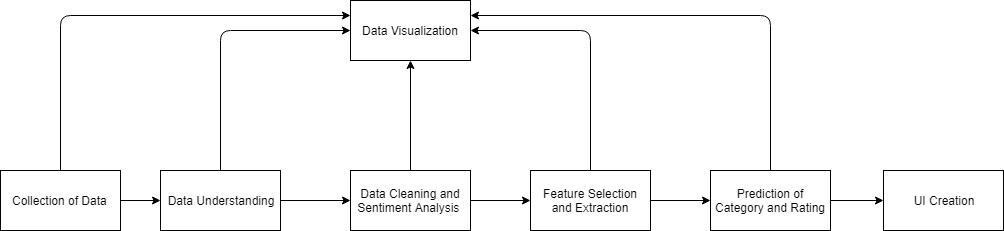
**Minimum Software Requirements**

|  |  |
| --- | --- |
| **Operating System** | Any operating system which can support an internet browser. |
| **Programming language** | - |
| **Other tools & tech** | Internet browser |

*Table 3.2 Minimum Software Requirements*

**CHAPTER: 4 PROCESS MODEL**

### CHAPTER 4 PROCESS MODEL



*Figure 4.1 Process Model of Project*

## CHAPTER: 5 PROJECT PLAN

### CHAPTER 5 PROJECT PLAN

### List of MajorActivities

Task: - 1 Data Understanding using Visualization

Task: - 2 Pre-Processing of Application Information Data and Visualization Task: - 3 Web Scraping of Reviews and its Visualization

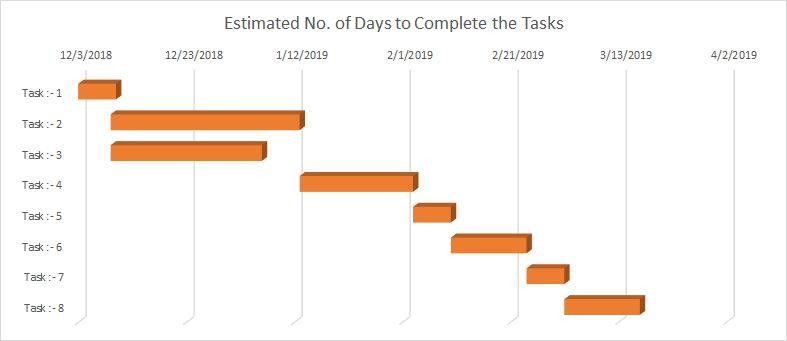
Task: - 4 Pre-processing of Reviews Data with Sentiment Analysis and its Visualization Task: - 5 Feature Selection and Extraction along with Visualization

Task: - 6 Implement a various model Rating prediction

Task: - 7 Selection of Best Fit Model

Task: - 8 UI Creation

### Estimated Time Duration inDays

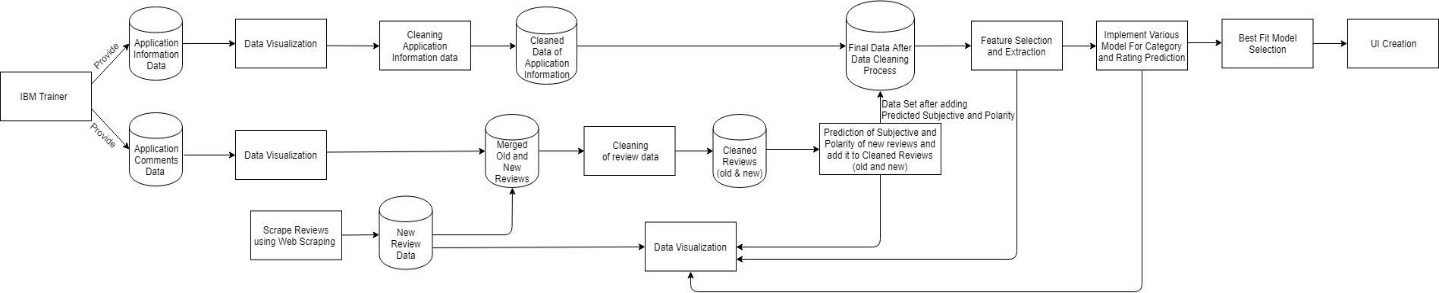


*Figure 5.1 Task Completion Estimated Time Duration in Days*

# CHAPTER: 6 IMPLEMENTATION DETAILS

### CHAPTER 6 IMPLEMENTATION DETAIL

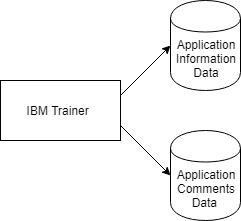
### Flowchart ofImplementation



*Figure 6.1 Project Implementation Flowchart*

### DataCollection

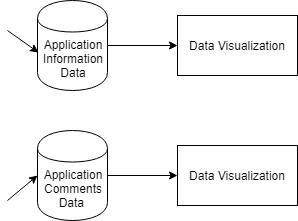
We are provided two different data (Application Information Data and Application Comments Data) from IBM trainer.



*Figure 6.2 Method of Data Collection*

### UnderstandingData

In this Phase, we have understood the structure of data-set with the help of visualization. In outcome we get to know that in application information data there are total 13 columns which are application name, category, rating, no of reviews, size, installs, type, price, content rating, genres, last update, current version and android version; talking about application comments there are total 5 columns which are application name, review, sentiment, polarity and subjectivity.

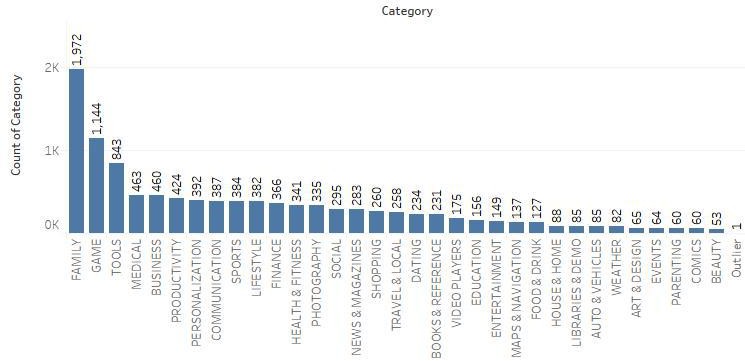


*Figure 6.3 Method of Data Understanding*

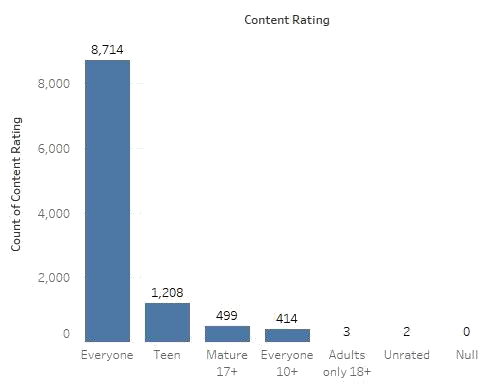
### DataVisualization:-

In this phase we have implemented various graph of different phases; phases which are considered are understood data, web scraping of comments, data cleaning, prediction of subjective and polarity, Feature selection and extraction, Model selection for rating prediction. Following are the things which we have implemented in this phase:-

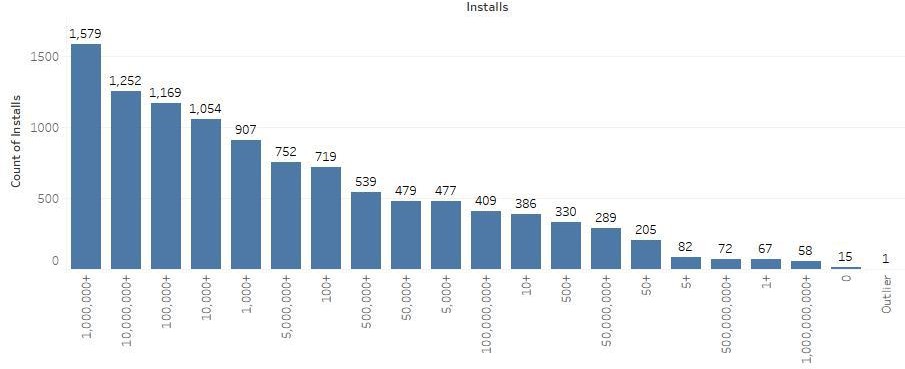
### Understand Original Application InformationData:-

By making graphs we have to understand the value of every column in the original data set. Below mention is the visualization of every column except android version and last update.

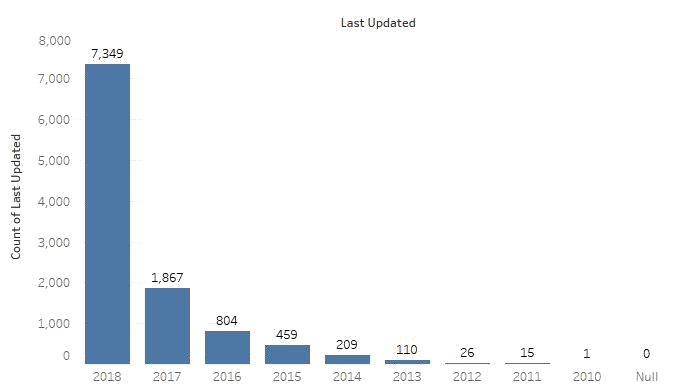
*Figure 6.4 Description of Category Column in Application Information Original Data*



*Figure 6.5 Description of Content Rating Column in Application Information Original Data-set*



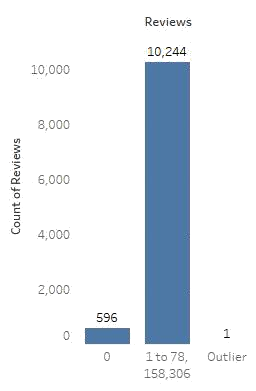
*Figure 6.6 Description of Installs Column in Application Information Original Data-set*



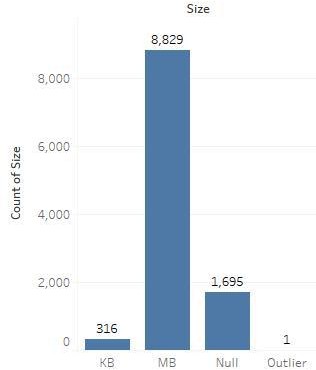
*Figure 6.7 Description of Last Updated Column in Application Information Original Data-set*



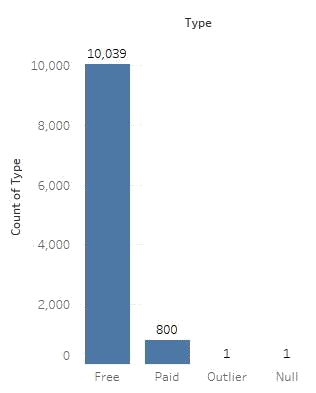
*Figure 6.8 Description of Rating Column in Application Information Original Data-set*



*Figure 6.9 Description of Reviews Column in Application Information OriginalData-set*



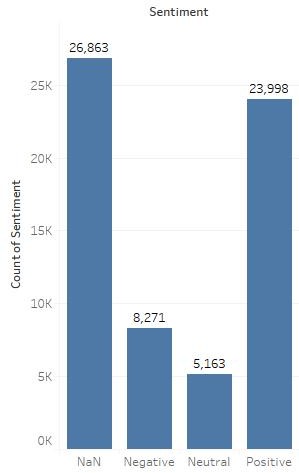
*Figure 6.10 Description of Size Column in Application Information OriginalData-set*



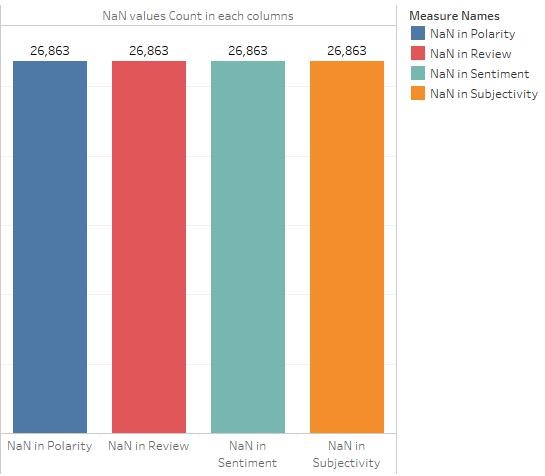
*Figure 6.11 Description of Type Column in Application Information Original Data-set*

### Understand Original Application ReviewData:-

By making graphs we have to understand the value of every column in the original data set of reviews. Below mention is the visualization from which we have carry out meaningful information which help us in pre-processing of this data-set.

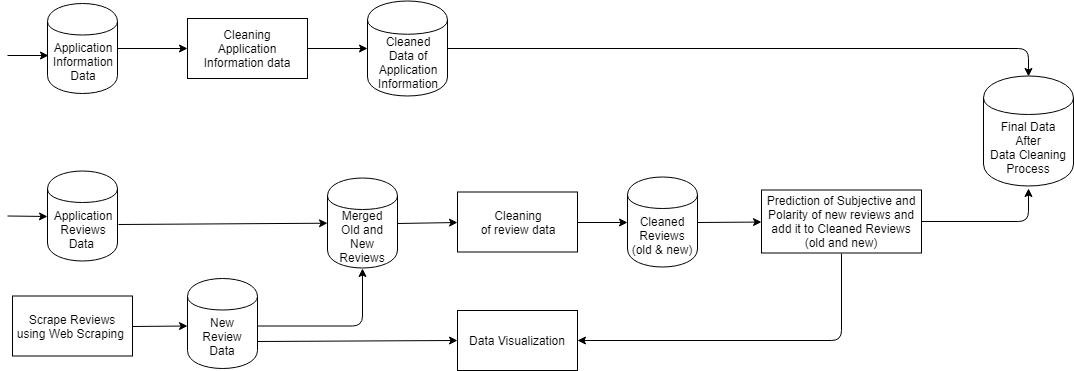


*Figure 6.12 Description of Sentiment Column in Application Reviews Original Data-set*



*Figure 6.13 Description of Total NaN values in all Column of Application Reviews Original Data-set*

### Data Cleaning and SentimentAnalysis:-



*Figure 6.14 Method of Data Cleaning and Sentiment Analysis*

* + - 1. **Data Cleaning of Application InformationData:-**

Following task are done in this phase using Smarten:-

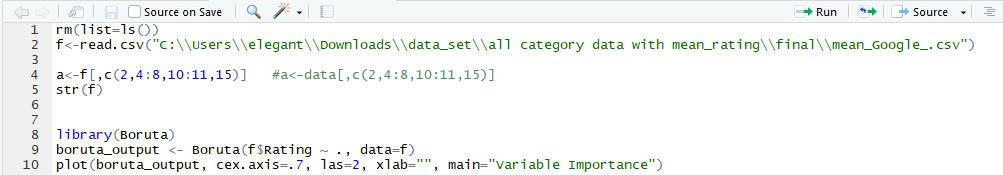
* Remove all rows with all samevalues.
* Find out NaN values in every column by table graph insmarten
* Remove rows which are not appropriate like category name 1.9 row
* Remove rows who’s rating was NaN, installs were 0 and review was0
* Select size column which only has values in Mb; remove M from it and then convert it into kb
* Place NaN in size column where there was varies withdevice
* In review, column replace NaN where there is varies withdevice
* Remove + from Installscolumn
* Remove $ from pricecolumn
* Remove and up from android version column and round value to by 1after.
* Separate data categorywise
* In the size column, we place mean of specific categories field where there was NaN value Rating column is our Output parameter so we decided to predict the value of NaN for that, due to that we can get high accuracy -> in category separated file first we place mean of specific category where there was NaN value and then we do the same thing for medianand make new CSV file for everycategory.
* Feature selection for modelcreation.[[3]](https://www.listendata.com/2017/05/feature-selection-boruta-package.html)[[4]](https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/)[[5]](https://www.machinelearningplus.com/machine-learning/feature-selection/)

Selected algorithm: - Boruta feature Selection

We have used Boruta feature selection because of various reasons which are listed below:-

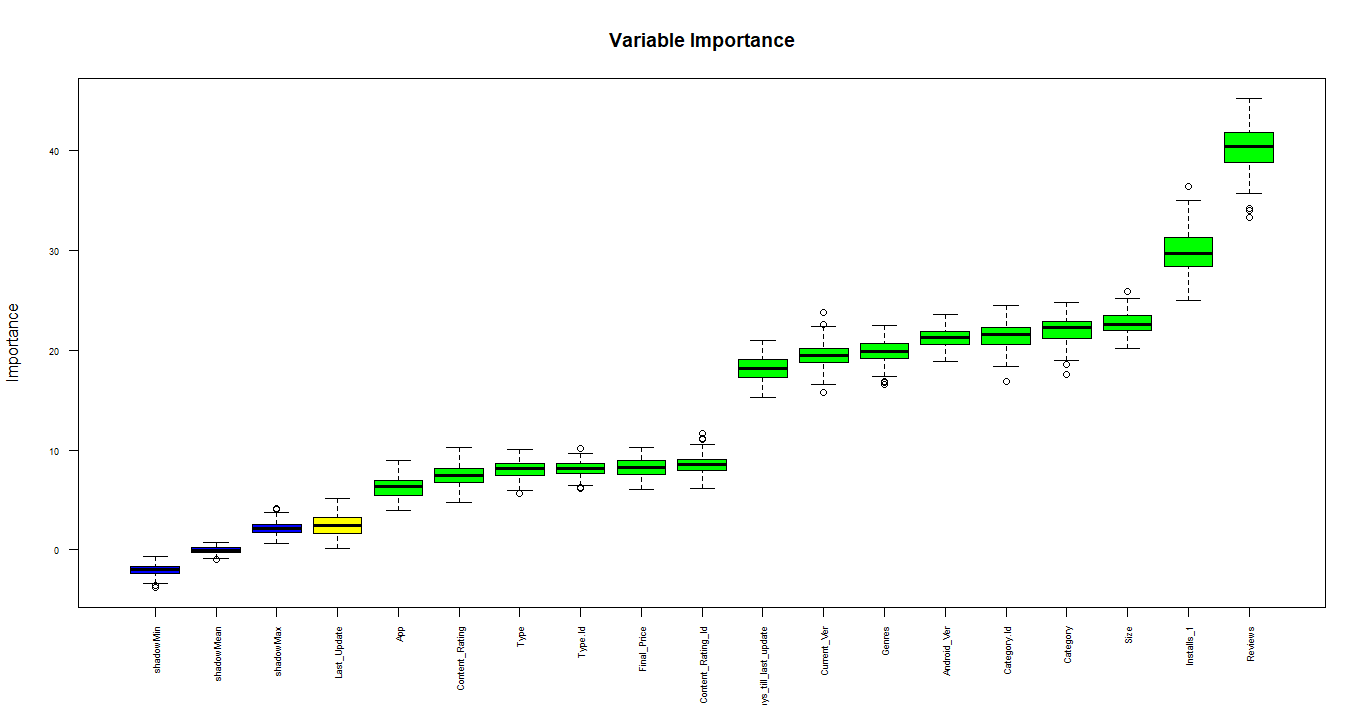
* It works well for both regression and classification problem.
* It takes multi-variable relationship into consideration.
* This algorithm is an advance improvement on random forest variable importance measure andthis method is getting popular in today’s world.
* It use all the features which are relevant to the output variable which is also known as all-relevant variable selection method. Where other feature selection methods only considersubset of features which yields a minor error so due to that other algorithm are minimaloptimal method as compare to boruta.
* Interactions between variables are also handled.

Code:-



*Figure 6.15 Code of Feature Selection*

Output:-



*Figure 6.16 Plot of Feature Selection*

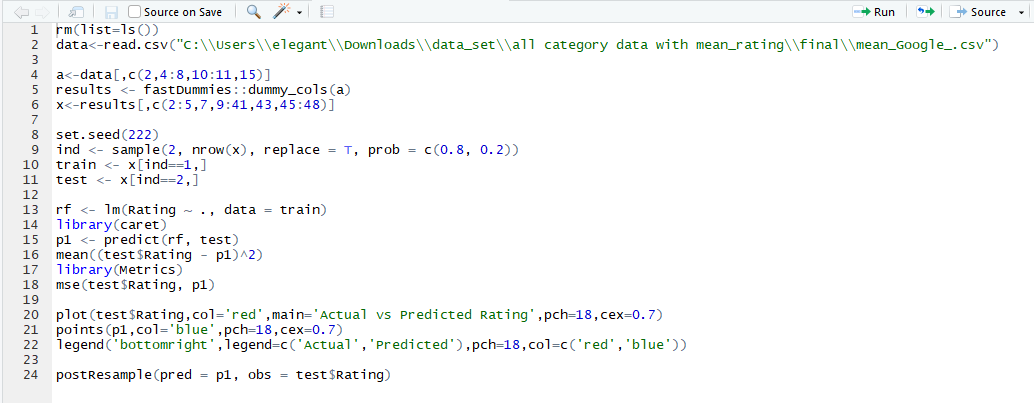
Selected variable for all model are:-

* Category Name
* Reviews
* Size
* Installs
* Type
* Final Price
* Content Rating
* No of days till last update
* Prediction of NaN value of rating by applying a machine learning model. (Imputation)[2]

Given below are the code and outcome of machine learning models applied on mean and median

data-set which are used for rating prediction.

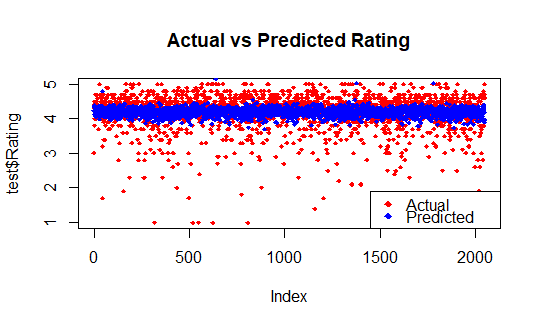
* Multiple Linear Regression



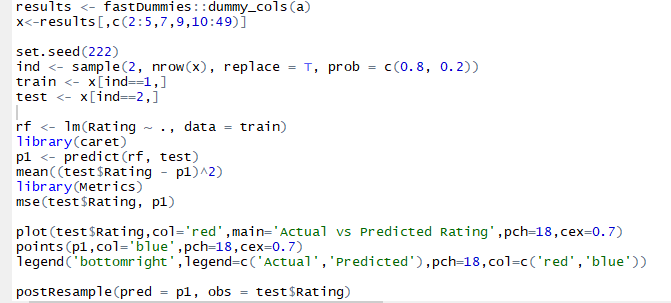
*Figure 6.17 Multiple Linear Regression on Mean Data*

C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Mean\multiple linear regression mean square error.PNG

*Figure 6.18 Mean Square Error of Multiple Linear Regression on Mean Data*



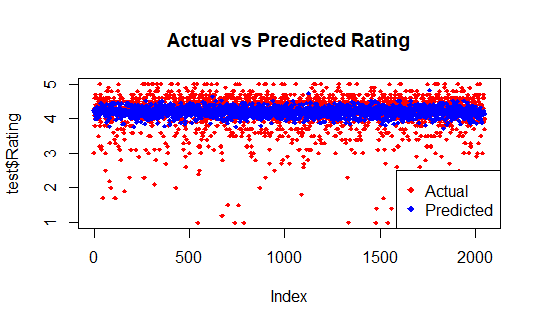
*Figure 6.19 Plot of Multiple Linear Regression on Mean Data*



*Figure 6.20 Multiple Linear Regression on Median Data*

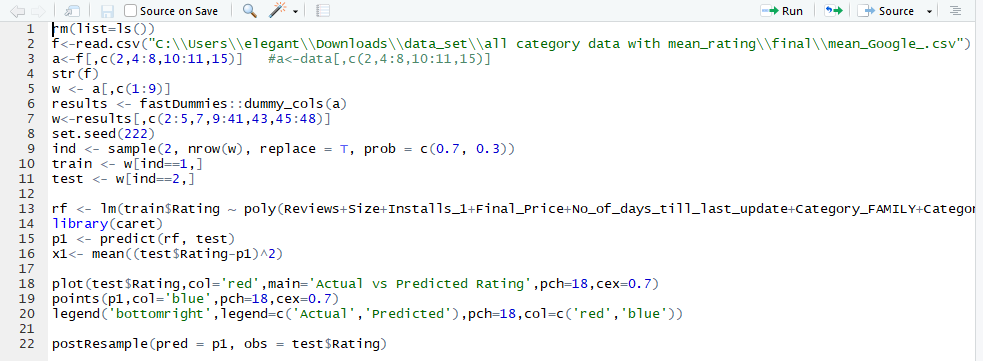
C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Median\Multiple Linear Regression Mean Square Error.PNG

*Figure 6.21 Mean Square Error of Multiple Linear Regression on Median Data*



*Figure 6.22 Plot of Multiple Linear Regression on Median Data*

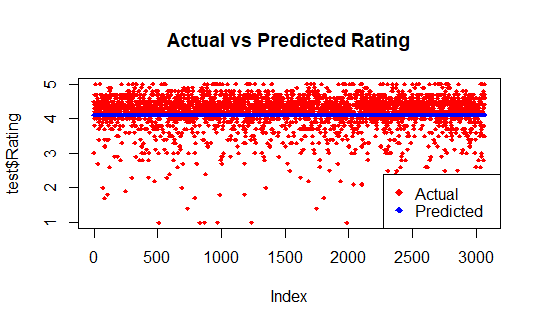
* Multiple Polynomial Regression



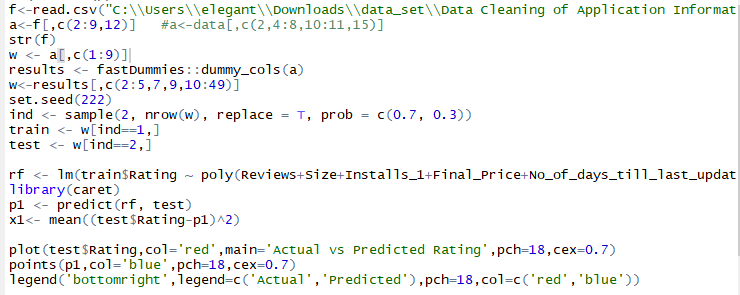
*Figure 6.23 Polynomial Regression on Mean Data*

C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Mean\poly regression mean square error.PNG

*Figure 6.24 Mean Square Error of Polynomial Regression on Mean Data*



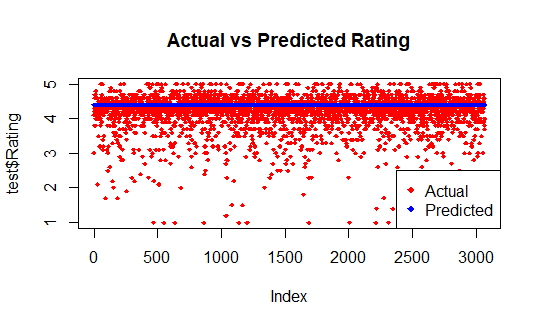
*Figure 6.25 Plot of Polynomial Regression on Mean Data*



*Figure 6.26 Polynomial Regression on Median Data*

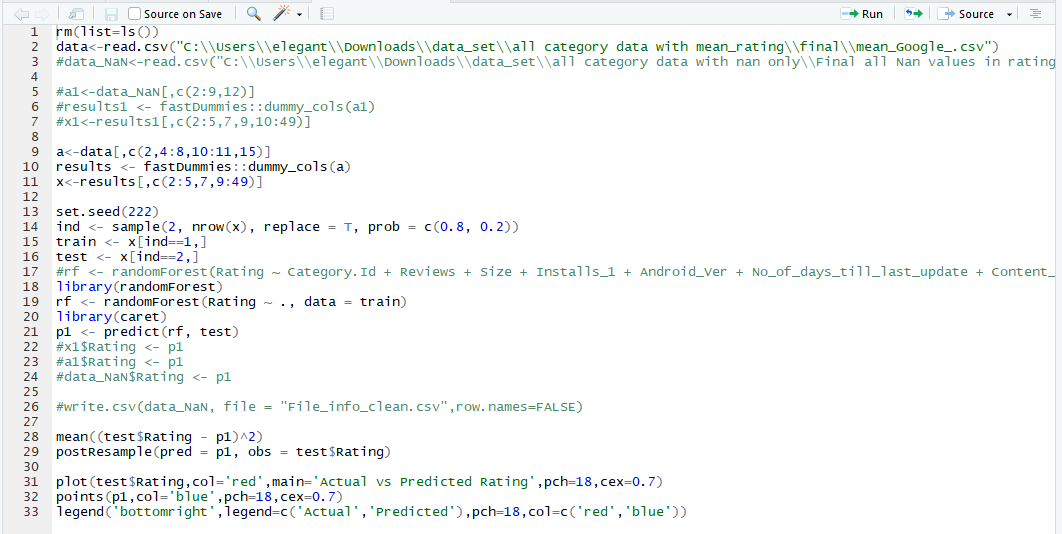
C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Median\Multiple Polynomial Regression Mean Square Error.PNG

*Figure 6.27 Mean Square Error of Polynomial Regression on Median Data*



*Figure 6.28 Plot of Polynomial Regression on Median Data*

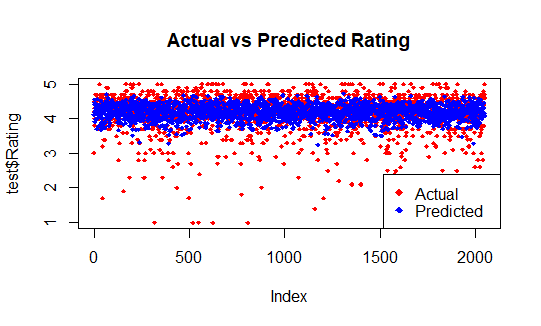
* Random Forest



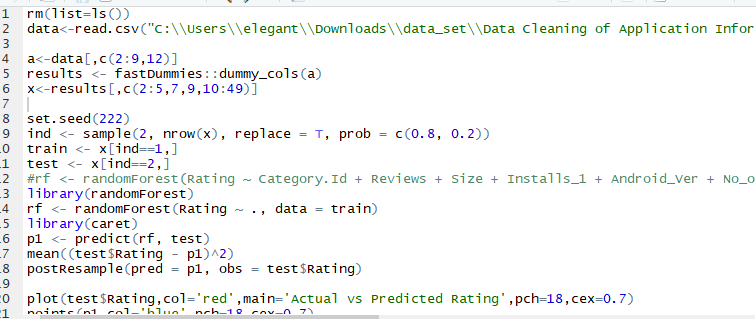
*Figure 6.29 Random Forest on Mean Data*

C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Mean\random forest mean square error.PNG

*Figure 6.30 Mean Square Error of Random Forest on Mean Data*



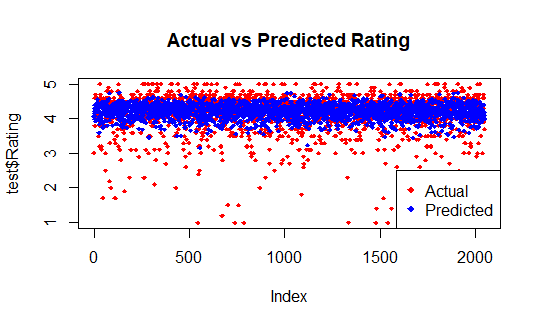
*Figure 6.31 Plot of Random Forest on Mean Data*



*Figure 6.32 Random Forest on Median Data*

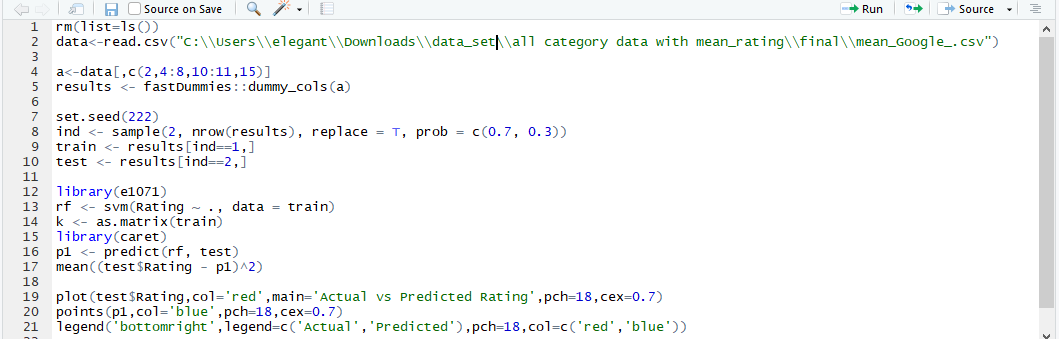
C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Median\Random Forest mean square error.PNG

*Figure 6.33 Mean Square Error of Random Forest on Median Data*



*Figure 6.34 Plot of Random Forest on Median Data*

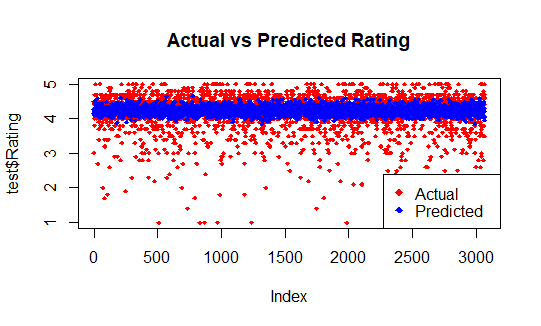
* Support Vector Machine



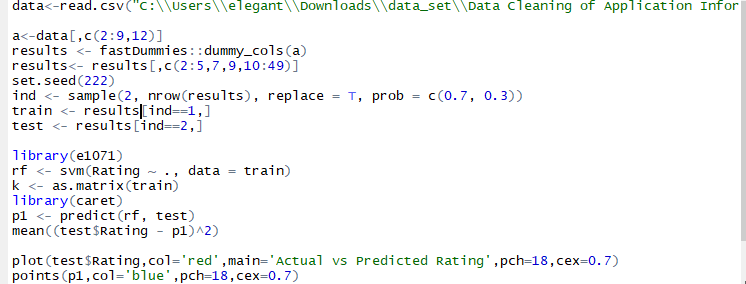
*Figure 6.35 Support Vector Machine on Mean Data*

C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Mean\SVM mean square error.PNG

*Figure 6.36 Mean Square Error of Support Vector Machine on Mean Data*



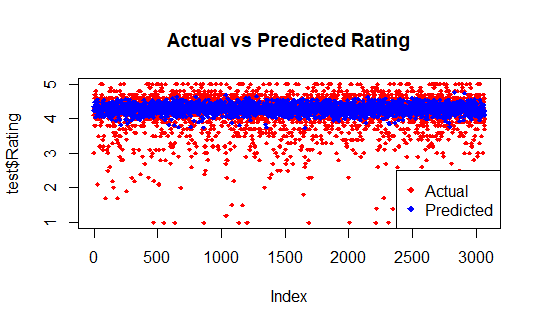
*Figure 6.37 Plot of Support Vector Machine on Mean Data*



*Figure 6.38 Support Vector Machine on Median Data*

C:\Users\admin\Documents\Ganpat\sem 8\IBM Project\Report\Review 2\Final Changes\code ss\Median\mean square error SVM.PNG

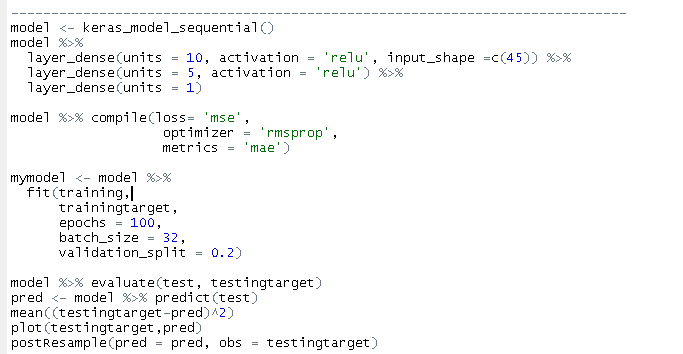
*Figure 6.39 Mean Square Error of Support Vector Machine on Median Data*



*Figure 6.40 Plot of Support Vector Machine on Median Data*

* Neural Network

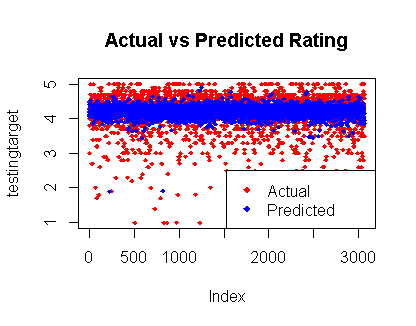




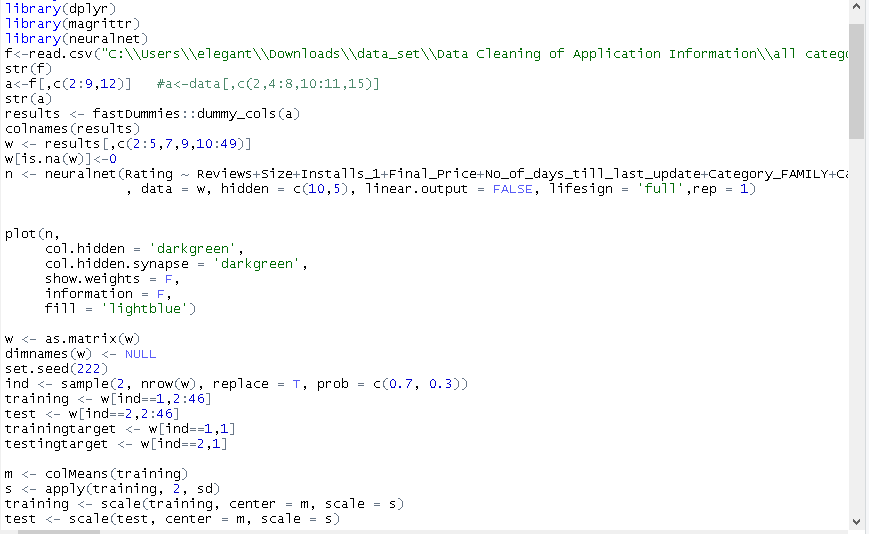
*Figure 6.41 Neural Network on Mean Data*

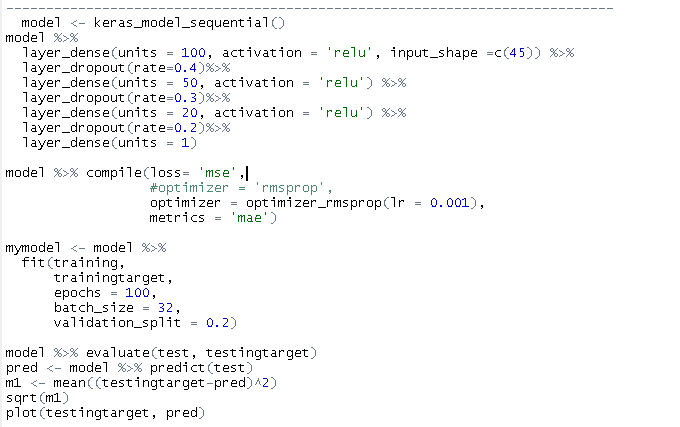
mse_mean_data_nn

*Figure 6.42 Mean Square Error of Neural Network on Mean Data*

**

*Figure 6.43 Plot of Neural Network on Mean Data*

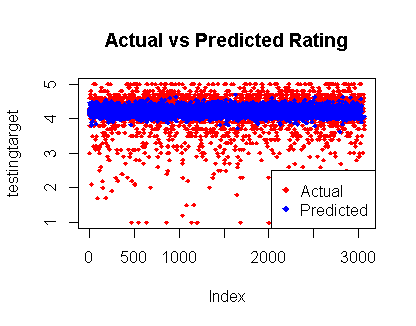




*Figure 6.44 Neural Network on Median Data*

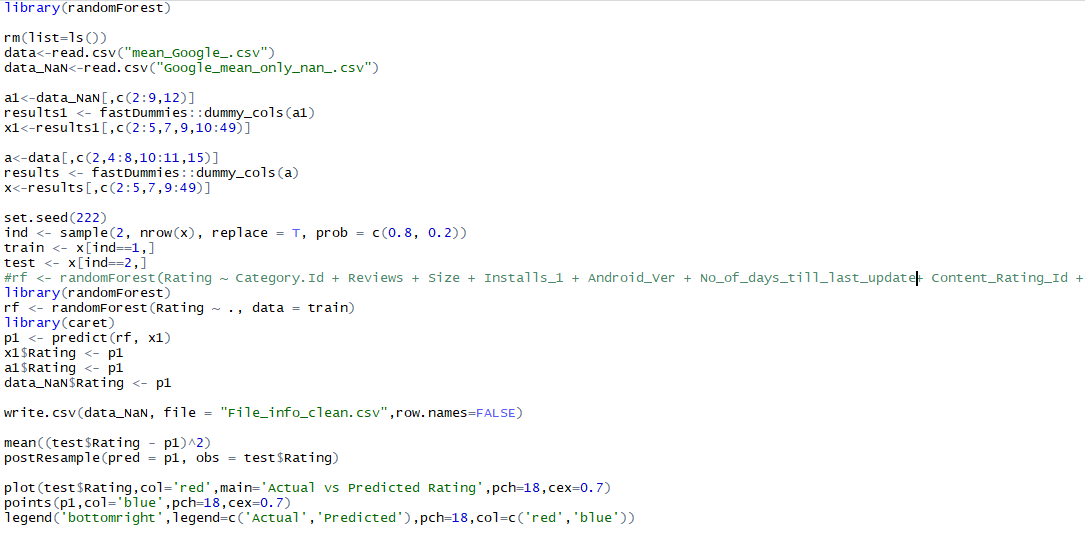
median_data_mean_square_error_nn

*Figure 6.45 Mean Square Error of Neural Network on Median Data*

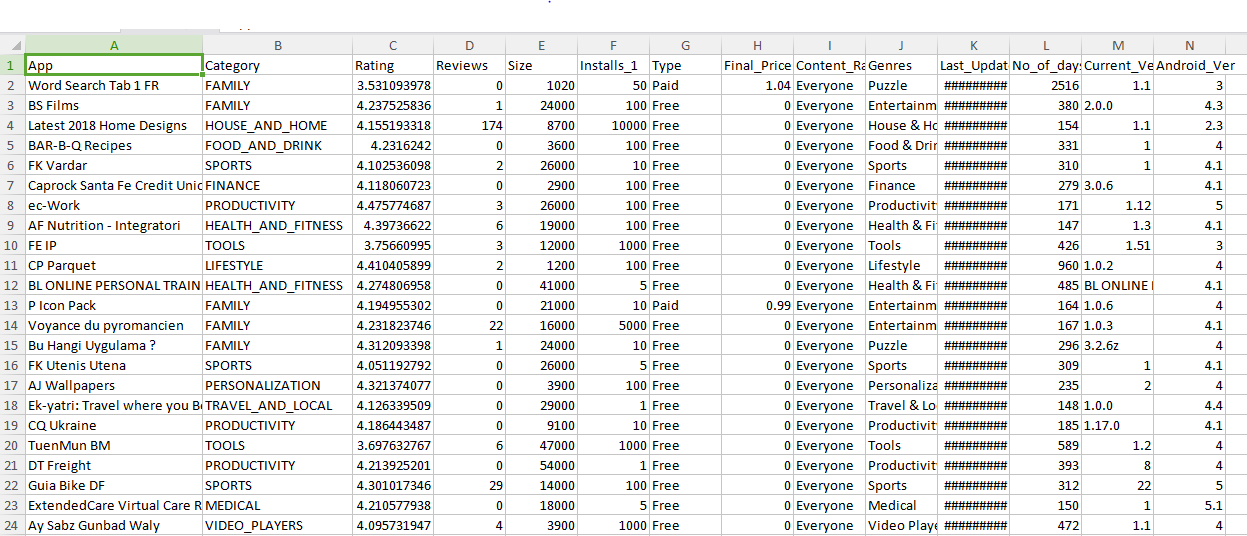


*Figure 6.46 Plot of Neural Network on Median Data*

* Replace NaN of rating with the help of the model which get low error; while using random forest on mean replaced data set we got less error as compare to other algorithm. Given below is the code for it:-



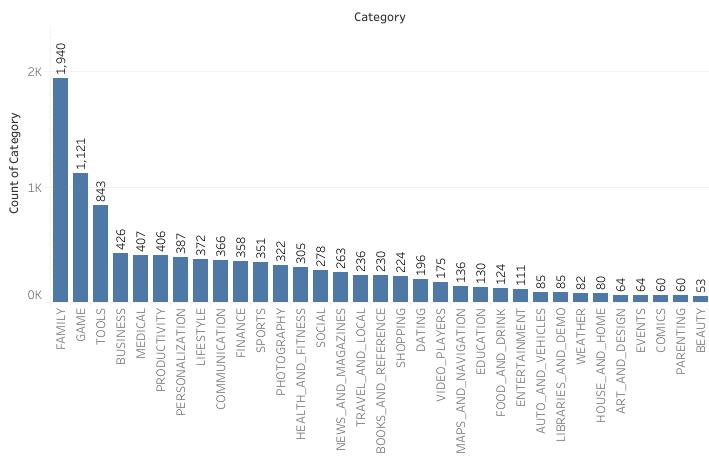
*Figure 6.47 Random Forest for NAN Values of Rating*



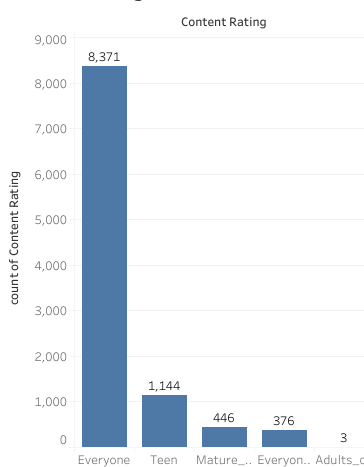
*Figure 6.48 NaN Rating Prediction Using Random Forest*

* + - 1. **Visualization of Application Information CleanData:-**

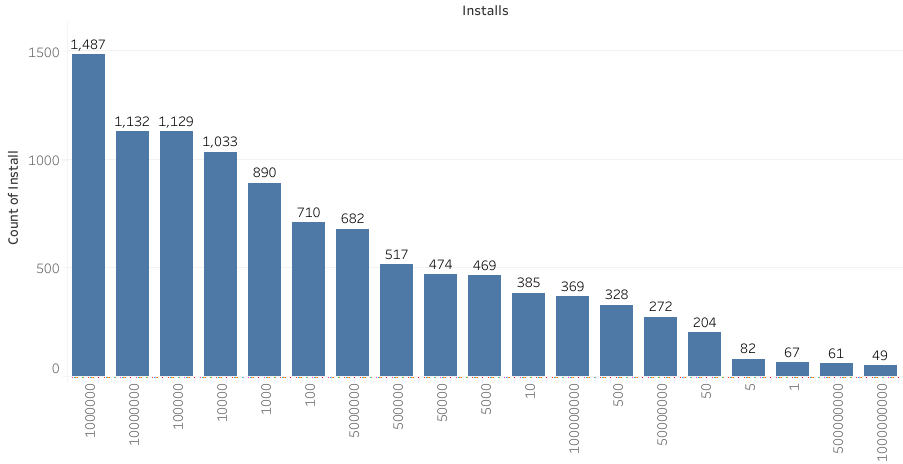
By making graphs we can compare both application information data before cleaning and after cleaning; by seeing a visualization of application information clean data we can also cross verify that if there is any unclean or noisy data is still there or not.



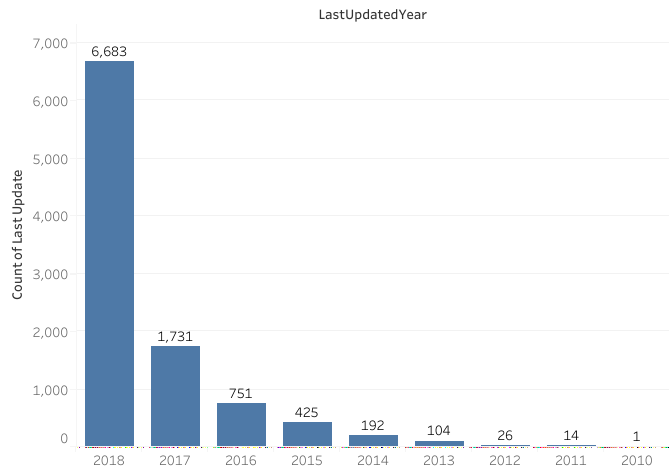
*Figure 6.49 Description of Category Column in Application Information of Clean Data*



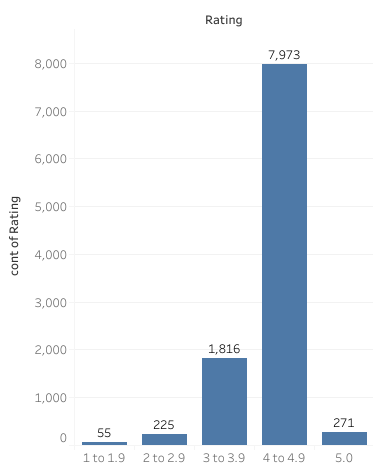
*Figure 6.50 Description of Content Rating Column in Application Information of Clean Data-set*

**

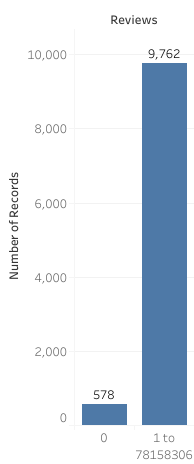
*Figure 6.51 Description of Installs Column in Application Information of Clean Data-set*



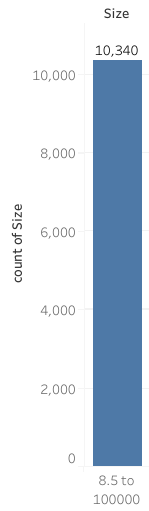
*Figure 6.52 Description of Last Updated Column in Application Information of Clean Data-set*



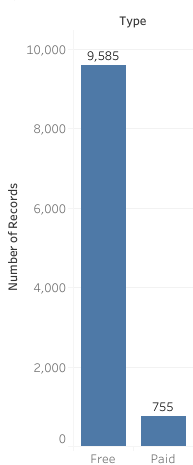
*Figure 6.53 Description of Rating Column in Application Information of Clean Data-set*



*Figure 6.54 Description of Reviews Column in Application Information of Clean Data-Set*

**

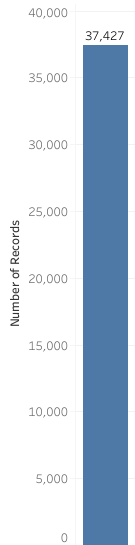
*Figure 6.55 Description of Size Column in Application Information of Clean Data-set*



*Figure 6.56 Description of Type Column in Application Information of Clean Data-set*

### 6.1.4.3 Data Cleaning and Visualization of Original Application Review Data:-

The original reviews data was having NaN values in columns of Translated Review, Sentiment, Polarity and Subjectivity. As it was not useful for sentiment analysis so we removed it and give below is its visualization.



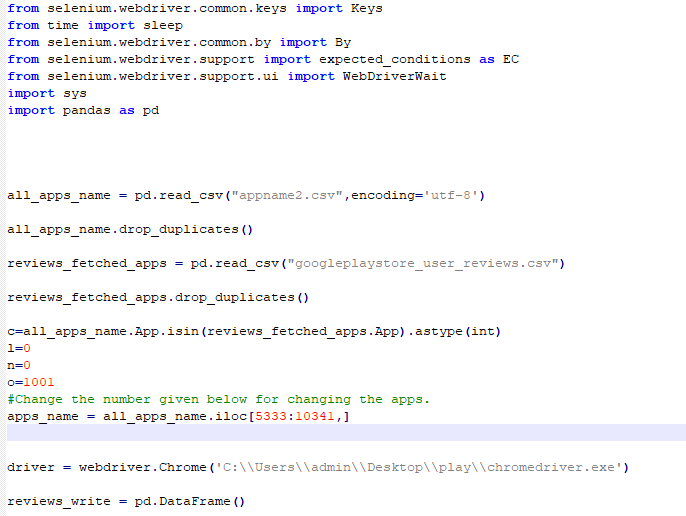
*Figure 6.57: Number of OriginalReviews after removing NaN data*

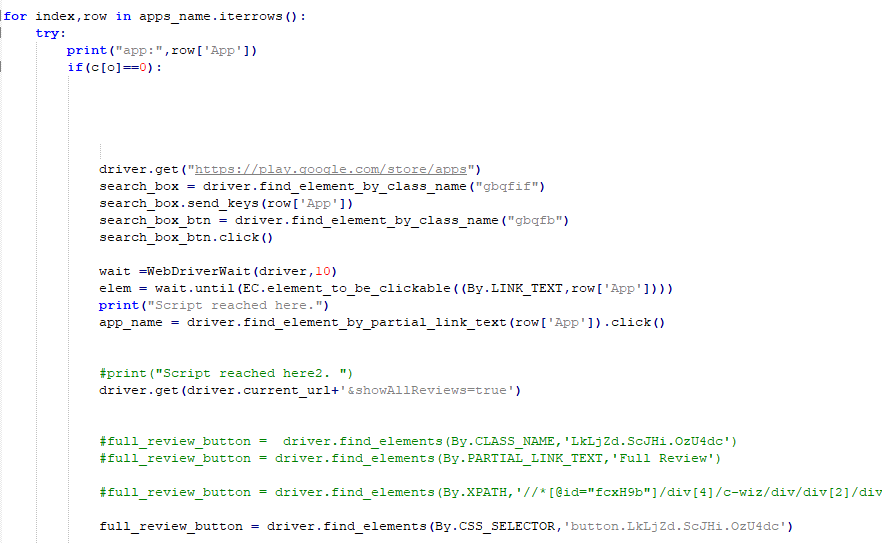
### 6.1.4.4 Web Scraping of NewReviews:-

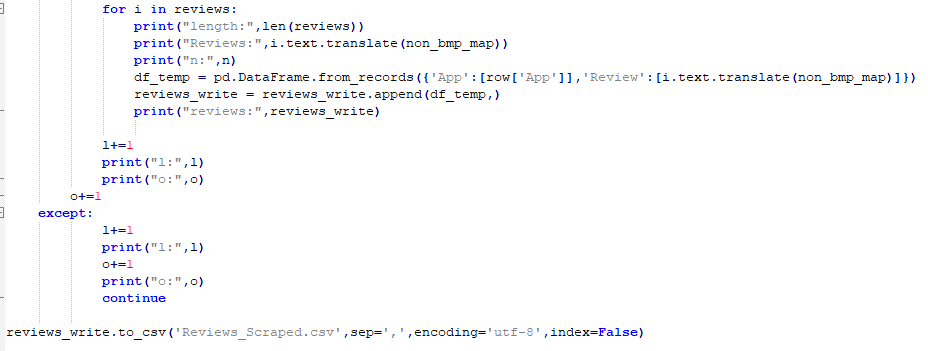
Now we are having only 37,427 no of reviews so due to that we decided to scrap the review of application from play store for that we have written a python script using selenium and pandas for scraping comments from play store; following is the process, which we have implemented through python code:-

* First, it will go to <https://play.google.com/store/apps>
* Then in the search bar it will search a name of theapplication
* After searching application name it will click on application name which matches to the text which was searched in search bar by clicking on it application page will beopen.
* In the next step, it will click on full review button in the comments section due to that we can see the whole comment.
* Finally, in the last step we fetch the first few comments by considering the most helpfulfirst filter.

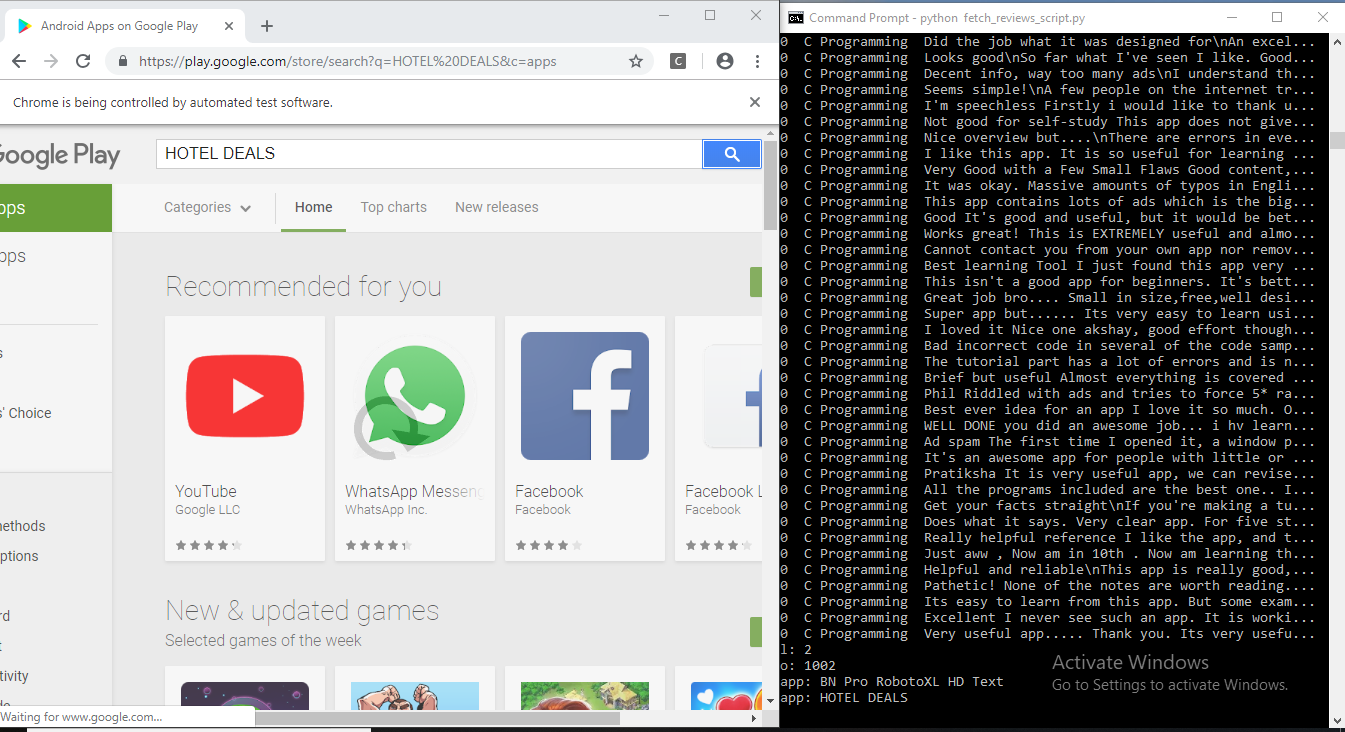
Given below are the screenshot of code and outcome:-







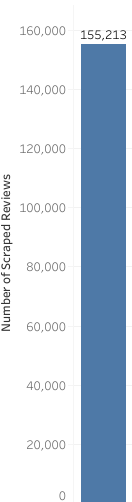
*Figure 6.58 Code for Review Scraping from Play Store*



*Figure 6.59 Result of Review Scraping from Play Store*

### 6.1.4.5 Visualization of Scraped Reviews:-

By this visualization we get clear idea that number of reviews which we have scraped are 155,213.

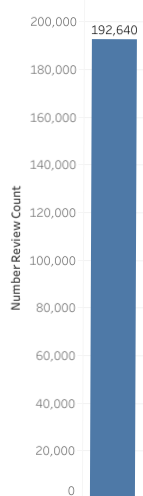


*Figure 6.60: Number of Scraped Reviews*

**6.1.4.6Merging and Visualization of Original Application Reviews Data and Scraped Reviews:-**

We have merge Original Application Review and Scraped Reviews dataset directly in excel only.

By this visualization we get to know that total number of reviews are 192,640 after adding both Scraped Reviews and Original Reviews clean review.

****

*Figure 6.61: Number of Reviews After merging Original Application Reviews and Scraped Reviews*

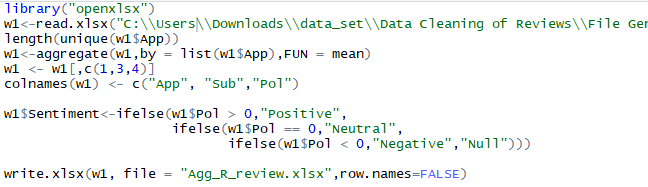
**6.1.4.7 Cleaning, Sentiment Analysis with Visualization of Merged Reviews:-**

### For both Cleaning and sentiment analysis we have used tm library in R. Given below is the sample code for it. We have used tm.plugin.sentiment package. First Step is to calculated sentiment subjectivity and sentiment polarity for all reviews. Given below is its code.

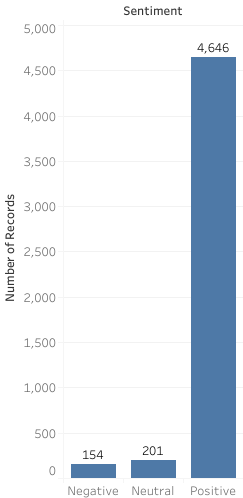
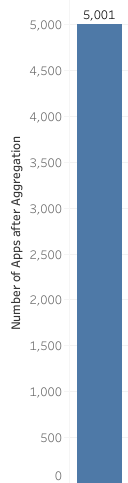
### *Finad Subjectivity and Polarity*

*Figure 6.62: Code to Calculating Sentiment Polarity and Subjectivity*

### Next Step is to calculated average polarity for each application and then assigned sentiment as positive, negative and neutral on the basis of polarity score. If polarity score is 0 then sentiment is neutral, polarity score is greater than 0 and less than or equal to 1 then sentiment is positive and if polarity score is between -1 to 0 then sentiment is negative. Given Below is its code and visualization.

**

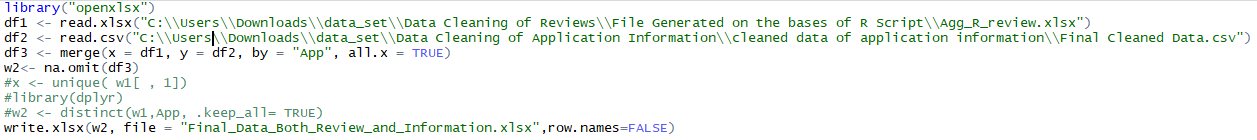
*Figure 6.63: Code to Calculating Average Polarity and Assigning Sentiment based on Application Name*

**

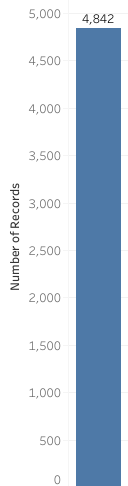
*Figure 6.64: No. of Application after AggregationFigure 6.65: Overall Sentiment of Applications*

**6.1.4.8 Merging Application Reviews and Information Data and its Visualization:-**

Now we are having both clean data of application information and its reviews so to make final prediction we have join both the dataset on the bases of application name. Given below is its code and its visualizations.

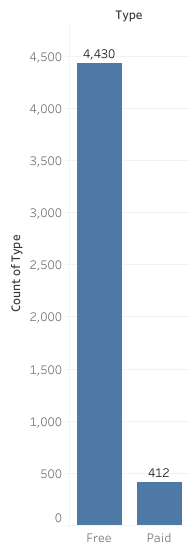
**

*Figure 6.66: Code to join Cleaned Application Review and Information Data based on Application Name*

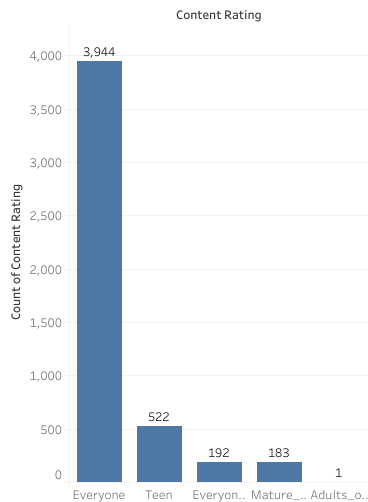


*Figure 6.67: Number of Applications in Final Clean Data*

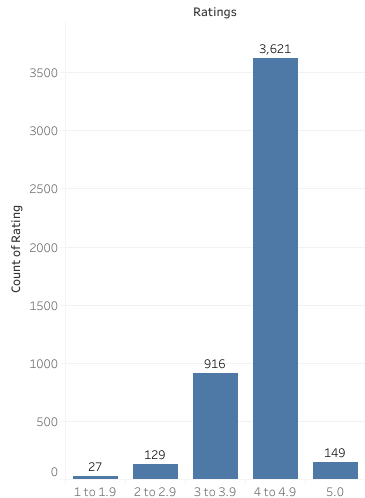
In this number of application get reduced as compare to number of application in clean review dataset because there are some application which are removed from application information during cleaning process and for scraping we have taken application name from original dataset which was given us.

****

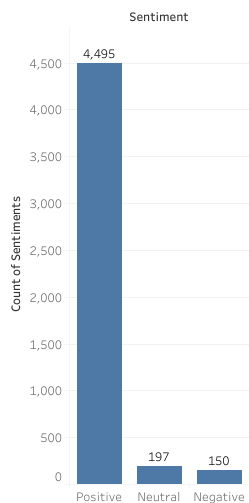
*Figure 6.68: Number of Free and Paid Application in Final Clean Data*

**

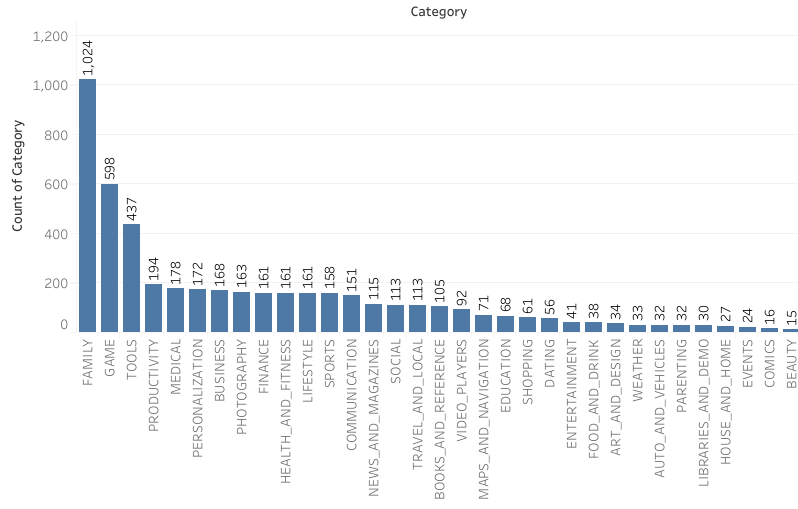
*Figure 6.69: Content Rating Wise Number of Application in Final Clean Data*

****

*Figure 6.70: Rating wise Number of Applicationin Final Clean Data*

****

*Figure 6.71: Sentiment wise Number of Applicationin Final Clean Data*

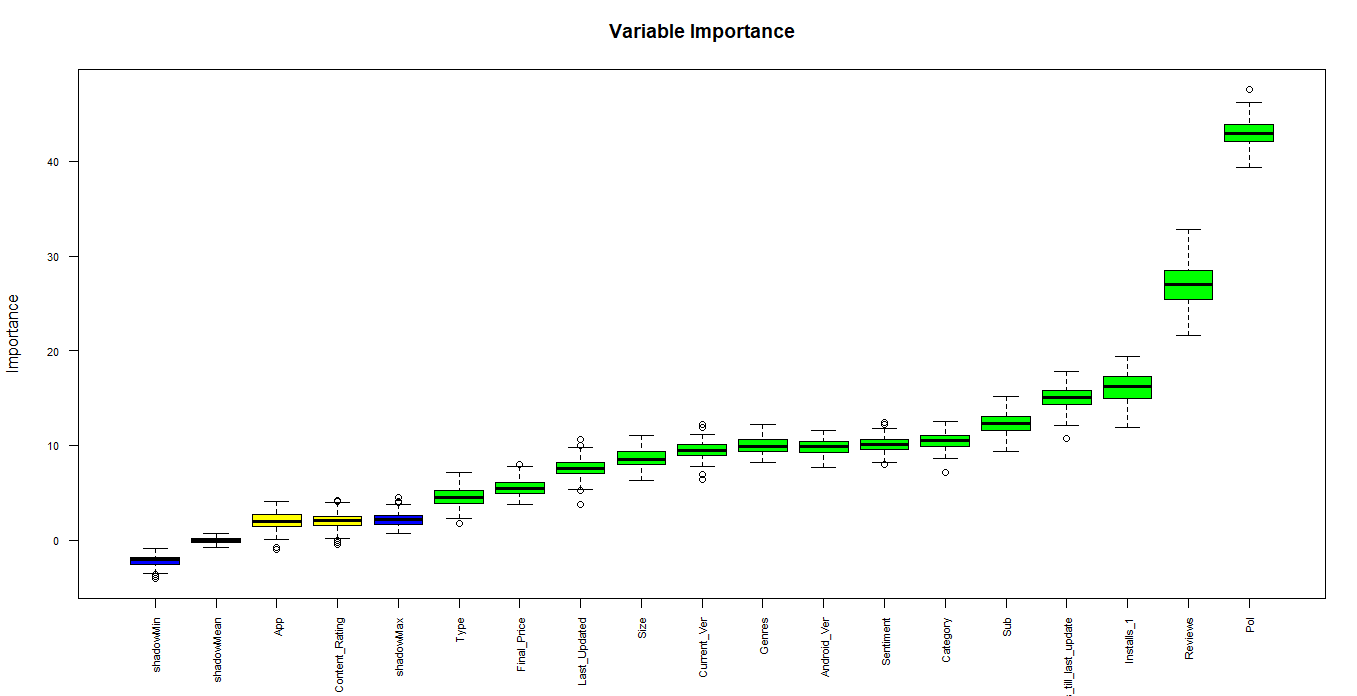


*Figure 6.72: Category Wise Number of Application in Final Clean Data*

**6.1.4.9 Feature Selection in Final Clean Data:-**

Selected algorithm: - Boruta feature Selection

The reason for using Boruta feature selection algorithm and it code is given in section 6.1.4.1 and figure 6.15, respectively so kindly, refer it for more information on Boruta feature selection algorithm. Given below is the outcome of this feature selection

**

*Figure 6.73: Outcome of Feature Selection based on Final Clean Data*

Selected features for all model are:-

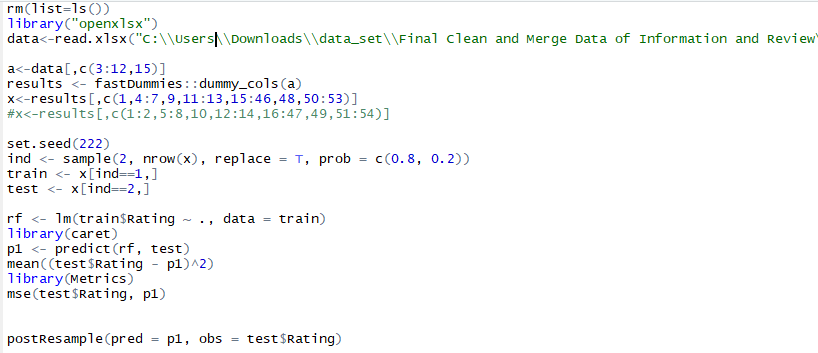
* Category Name - Installs - Content Rating - Sentiment
* Reviews - Type - Polarity
* Size - Final Price - No of days till last update

**6.1.4.10 Prediction of Rating on clean data:-**

Here, we have used total 5 algorithms for prediction of rating and selected a best fit algorithm on the basis of low mean square error; features which are confirmed through the Boruta algorithm and our basic logic are used as an input parameters for prediction.

Following are the list of algorithms along with its code and outcome:-

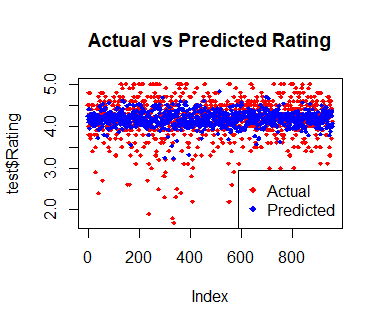
* Multiple Linear Regression



*Figure 6.74: Multiple Linear Regression Code for final Clean Data*

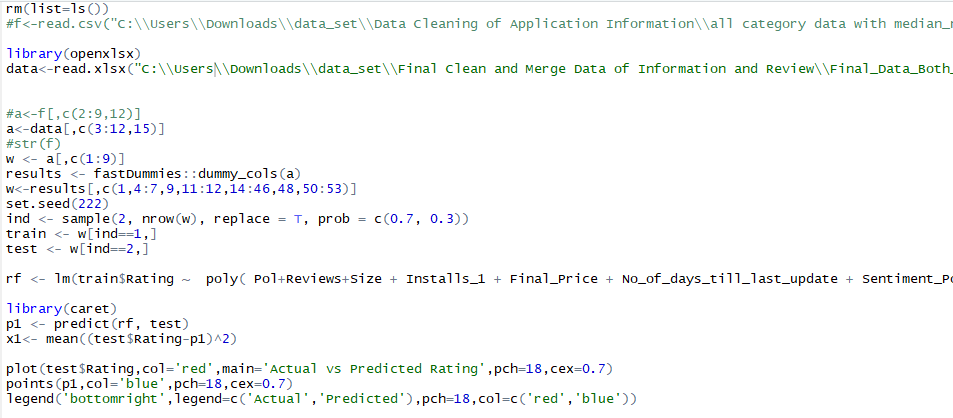
Multiple Linear Regression Mean Square Error

*Figure 6.75: Mean Square Error of Multiple Linear Regressionfor Final Clean Data*



*Figure 6.76: Plot of Multiple Linear Regression for Final Clean Data*

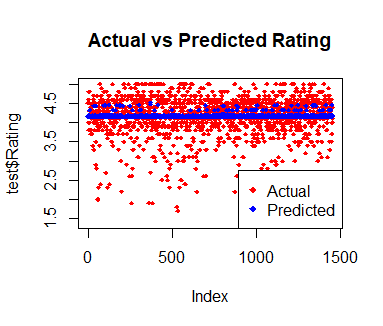
* Multiple Polynomial Regression



*Figure 6.77: Polynomial Regression Codefor Final Clean Data*

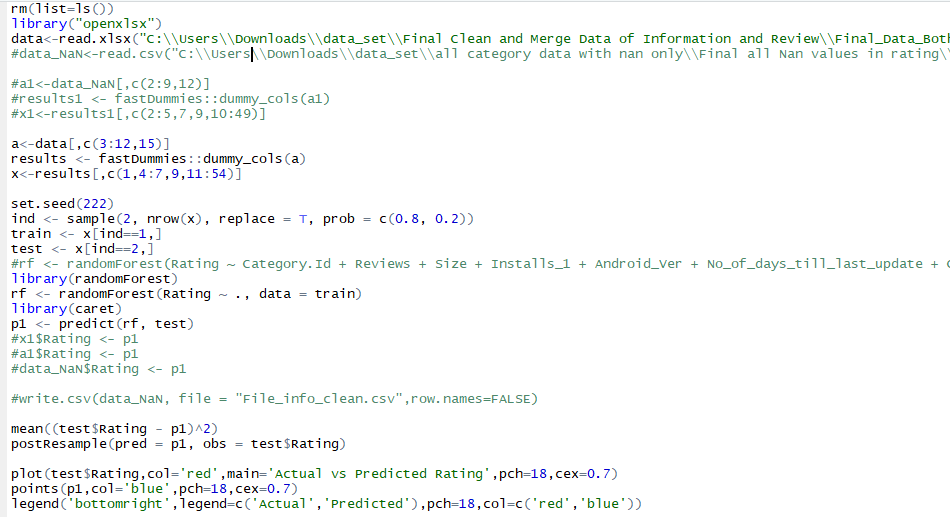
Multiple Polynomial Regression Mean Square Error

*Figure 6.78: Mean Square Error of Polynomial Regressionfor Final Clean Data*



*Figure 6.79: Plot of Polynomial Regressionfor Final Clean Data*

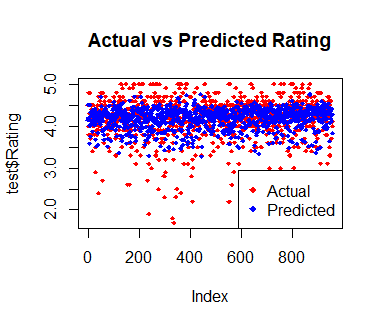
* Random Forest



*Figure 6.80: Random Forest Codefor Final Clean Data*

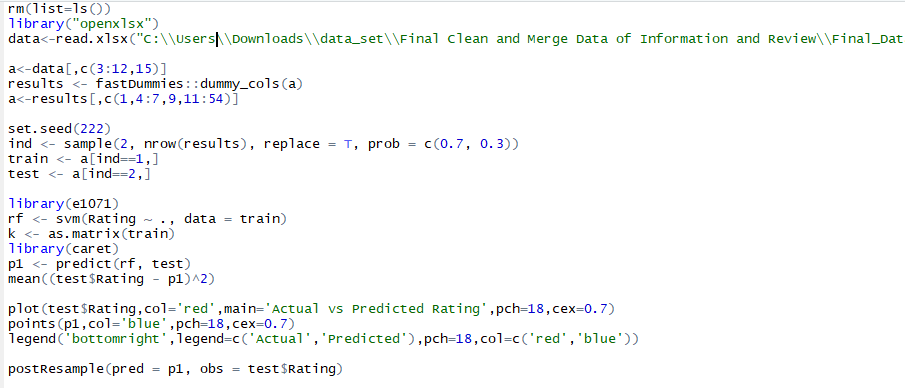
Random Forest Mean Square Error

*Figure 6.81: Mean Square Error of Random Forestfor Final Clean Data*



*Figure 6.82: Plot of Random Forestfor Final Clean Data*

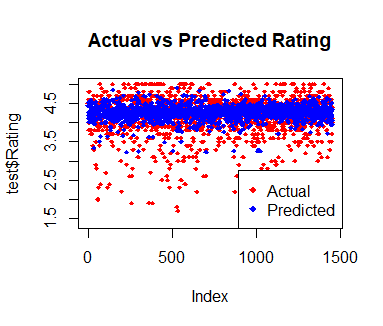
* Support Vector Machine



*Figure 6.83: Support Vector Machine Codefor Final Clean Data*

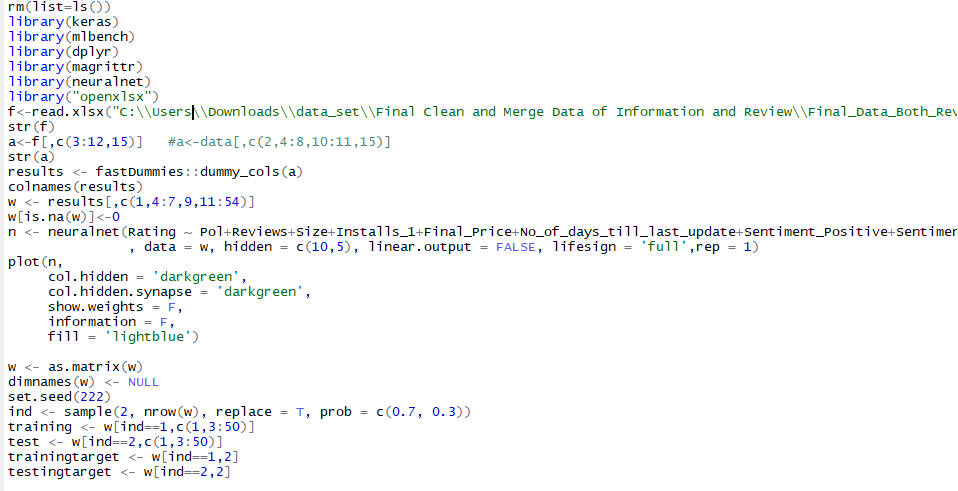
SVM Mean Square Error

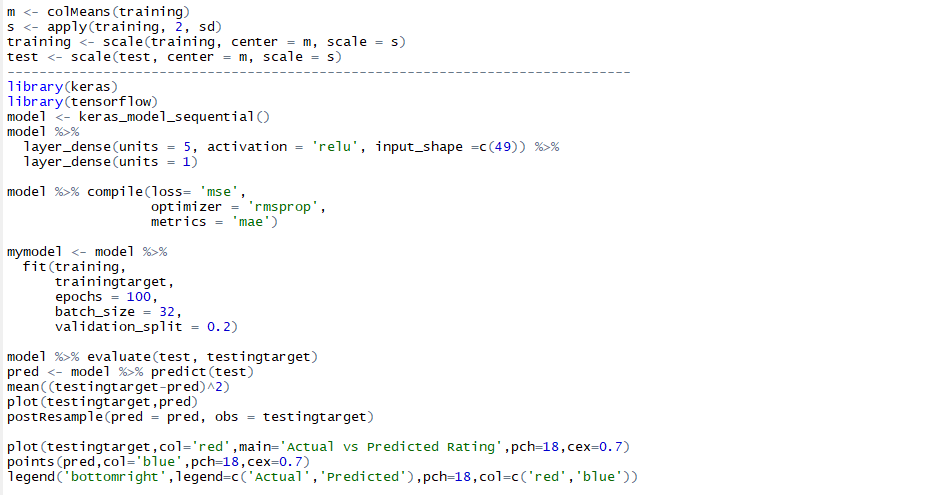
*Figure 6.84: Mean Square Error of Support Vector Machinefor Final Clean Data*



*Figure 6.85: Plot of Support Vector Machinefor Final Clean Data*

* Neural Network

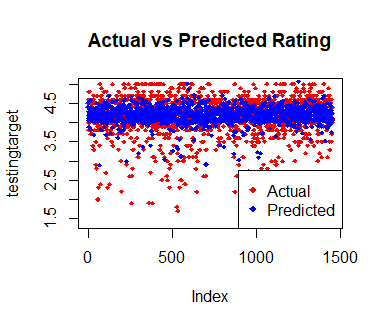




*Figure 6.86: Neural Network Codefor Final Clean Data*

Neural Network Mean Square Error

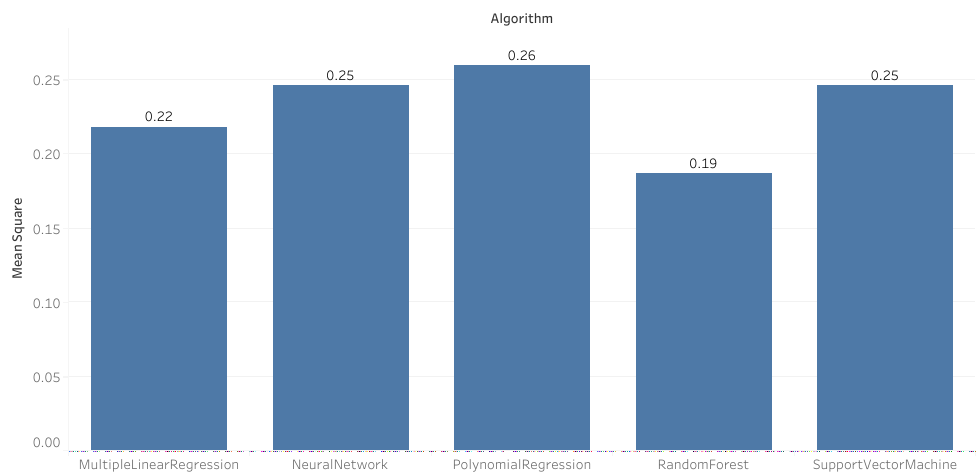
*Figure 6.87: Mean Square Error of Neural Networkfor Final Clean Data*

**

*Figure 6.88: Plot of Neural Networkfor Final Clean Data*

**6.1.4.11 Selection of Best Fit Model with Visualization:-**

We have predict rating by various model which are discuss above but in random forest we have very less mean square error as compare to other mode which is clearly seen in below visualization so we have consider random forest as our final rating prediction.



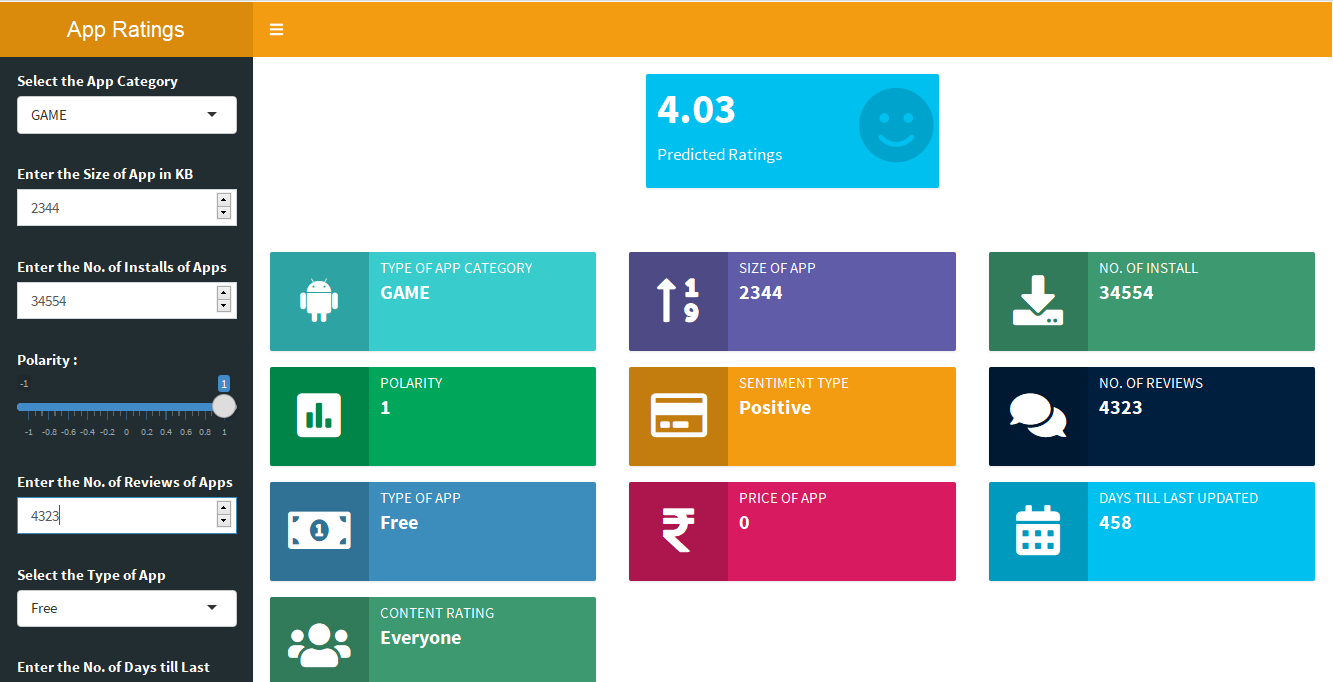
*Figure 6.89: Comparison of All Algorithm Mean Square Error*

**6.1.4.12 UI Creation**

We prepared a UI using R. This shiny application is dynamic and validate the input.This UI is able to take input from user and predict ratings according to input. It is available on (deployed link).



*Figure 6.90: Shiny UI code*



*Figure 6.91: Shiny Application*

**CHAPTER: 7 CONCLUSION AND FUTURE WORK**

### CHAPTER 7 CONCLUSION AND FUTURE WORK

**Conclusion**

To reiterate my views, as we all know that the ratio of mobile application development has rapidly increased in the last few years. Before developing a mobile application if we can know the approximate rating with the help of its properties which we are going to consider while development then by this we can conclude that will our application will become a successful app in future? If not then we can also know what kind of properties will help to make it popular in the real world. We are developing a UI based application which will predict approximate rating with that help of properties of application and this UI based application will work only on the application which is going to launch on Google PlayStore.

### Future work

Till now we have clean application properties data and on the other side, we have also scrap reviews from Google Play Store. Moreover, we have also done data visualization for original data-set. Following are the task which we are going to implement now.

* Pre-processing ofreviews
* Sentimentanalysis
* Merging of cleaned application properties and reviewsdata.
* Feature selection

# CHAPTER: 8 REFERENCES

### CHAPTER 8 REFERENCES

1. <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>
2. <https://towardsdatascience.com/the-tale-of-missing-values-in-python-c96beb0e8a9d>
3. <https://www.listendata.com/2017/05/feature-selection-boruta-package.html>
4. <https://www.analyticsvidhya.com/blog/2016/03/select-important-variables-boruta-package/>
5. <https://www.machinelearningplus.com/machine-learning/feature-selection/>