OR 568/SYST568 FINAL PROJECT REPORT

PREDICTIVE ANALYTICS

STATISTICAL ANALYSIS AND VISUALIZATION OF GOOGLE PLAY STORE DATA

By,

Deepak Sadayampatti (G01212219)

Ankita Tapadia (G01207031)

Dilpreet Singh Bedi (G01224028)

Rushika Bejjanki (G01176754)

Prateek Chitpur (G01163985)

1 INTRODUCTION:

1.1 ABOUT GOOGLE PLAY STORE

Google Play, formerly Android Market, is an American digital distribution service operated and developed by Google. It serves as the official app store for the Android operating system, allowing users to browse and download applications developed with the Android software development kit (SDK) and published through Google. [1]

Google Play Store, as we know it is used by millions of people on their android phone to download apps from different genre like gaming, fitness, health, tools, food and drink, education etc., to meet the daily needs. This data was scraped from Google Play Store and put into the csv file using jQuery and made available in Kaggle. The inspiration behind using this dataset is to understand the potential of the Apps to drive the app making business to success. Besides that, the developers can make sense of what makes an Application to be a hit; so as to capture the Android Market.

Applications success or rating depends on it being user friendly(reviews) by different age category users. Application updates depends on its popularity (number of installs), size of the application, reviews and rating already provided to the application by the user(s) and the category the application belongs to.

1.2 DATASET:

The dataset contains detailed attributes of approx. 10,800 apps present in Google App store as of 05/14/2020. The dataset was obtained from Kaggle and contains 13 attributes describing the App such as Genre, Rating and Review given by the users etc.,

1.3 SCHEMA:

Variable	**Data Type**	**Description**
App	Nominal	Name of the Application
Category	Nominal	The Category of the application
Rating	Ratio	Rating given to the application
Reviews	Ratio	Number of Reviews for the application
Size	Interval	Size of application in Bytes
Installs	Ratio	Number of times the application is installed
Type	Ordinal	Whether the application is Free or Paid.
Price	Ratio	If the app is paid, then the cost of the application
Content Rating	Ordinal	Rating is done by which category person like teen or adult
Genres	Ordinal	Genre of the application Category
Last Updated	Interval	The date when the application was last updated
Current Ver	Interval	Current version of the application available
Android Ver	Interval	Works on which Android versions

Table 1: Data Type and Schema

1.4 DATA PRE-PROCESSING:

The dataset had certain columns/attributes which were not of relevance for the analysis such as “Genre” and “Last Updated Date” denoting when the application had a software update. Also, the dataset contained duplicate records. As a first step these variables were dropped, and the duplicates were removed. Post this the dataset contained a lot of missing values around 1000**.** The missing values were imputed by the following methodology

Attribute/Variable: Rating of the Application

For this variable, the missing value was replaced by taking the mean of the Category as it is a continuous variable. For instance, the app named “EASY AND FAST RECIPES” had missing values for rating. It was replaced with the mean of the Category it belonged to i.e., “FOOD AND DRINK”.

Attribute/Variable: Size of the Application

The size of the application was denoted in terms of MB and KB for different applications. The first step here was to convert the size into a uniform value to enable apples to apples comparison. Hence the data was converted to KB by multiplying it with 1000 (i.e., MB TO KB conversion). For this variable, the missing value was replaced by taking the mean of the Category as it is a continuous variable. For instance, the app named “USED CARS AND TRUCKS FOR SALE” had missing values for rating. It was replaced with the mean of the Category it belonged to i.e., “AUTO AND VEHICLES”.

Correlation Plot

Below is the correlation plot of all the numerical variable of the data set. It is visible that the correlation between the Reviews and Installs variable is the strongest with a value of 0.63. Installs and Reviews variable are positively correlated.

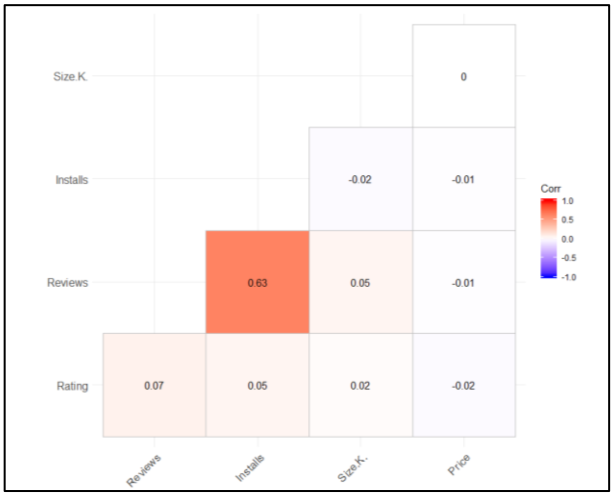


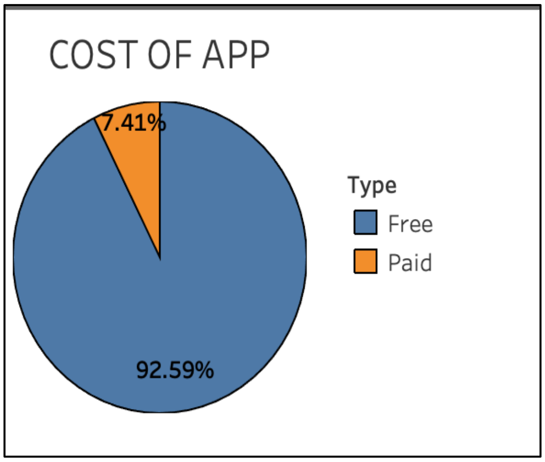
Figure 1: Correlation Plot

2 EXPLORATORY DATA ANALYSIS:

There are many lingering questions we have based on our daily usage of the App store/Play store that can be answered based on data visualizations such as

1. **Which kind of apps are downloaded the most; paid or free**?

   The Free app dominates the Market with 6873 apps contrary to the 550 Paid Apps. It is seen that approx. 93% of the applications are Free compared to approx. 7% of paid apps.



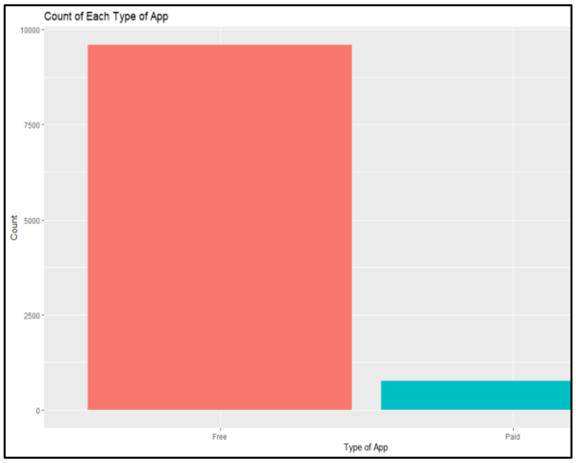


Figure 2: Cost of an Application

Figure 3: Histogram of Application Type

2. **Which App Category is the most popular in the market?**

The Family related apps are observed to be more popular when compared to other categories in the market, having 1,943 apps in that category.

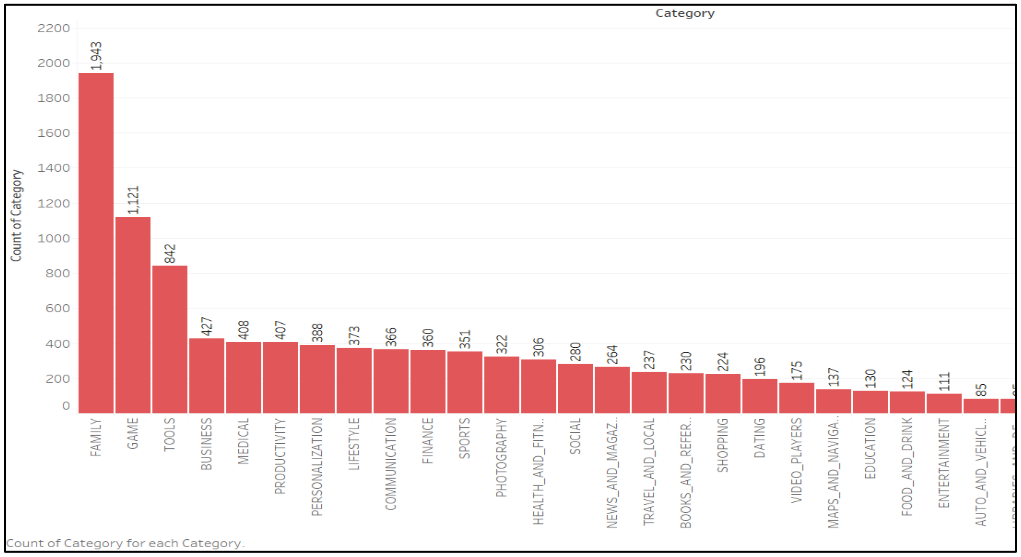


Figure 4: Most popular APP Category in the Market

3. Which App Category is more popular based on the Installs?

Based on the number of installs, we can observe from this visualization, that the apps related to gaming category are more popular than the other categories.

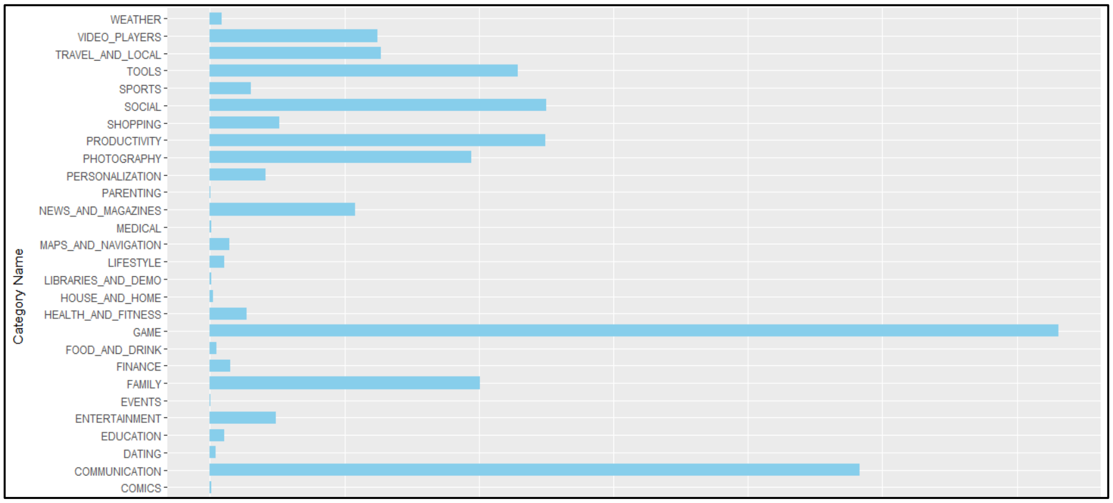


Figure 5: App Category based on Installs

4. Which App Category is more popular based on the Reviews?

Based on the number of reviews, we can observe that the apps related to gaming category are popular. From Figure 3 and Figure 4 it is observed that, Game category is the most popular category having maximum number of installs and maximum number of reviews.

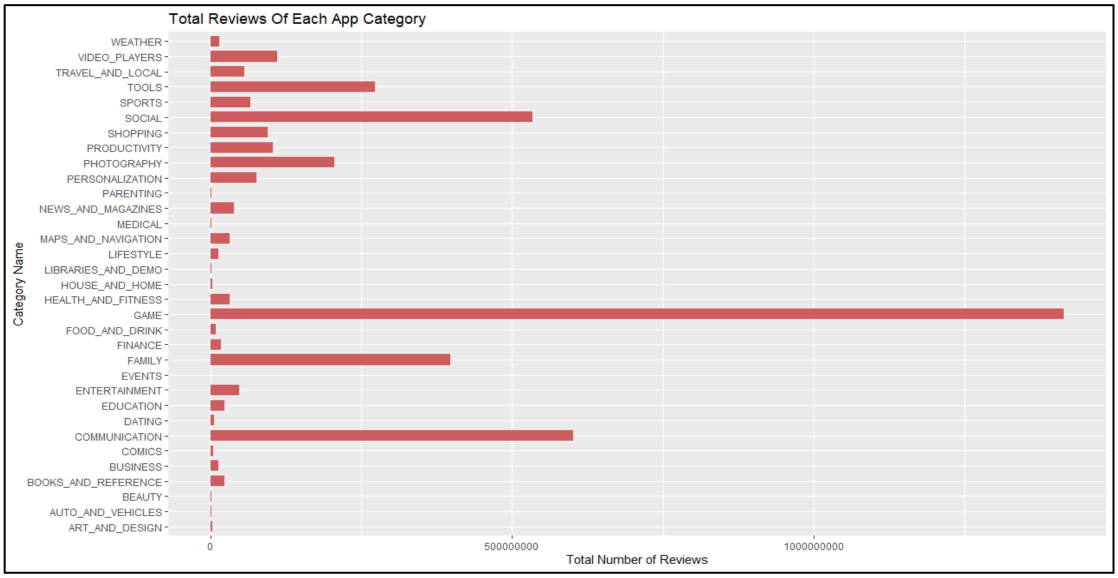


Figure 6: App Category based on Reviews

5. Which App is the most popular based on the app Reviews?

Figure 5 shows the apps based on the number of reviews. It is observed that Instagram is the most popular app among the users having 199,664,676 Reviews, followed by Facebook with 156,286,514 Reviews and Subway Surfers with 138,606,606 Reviews.

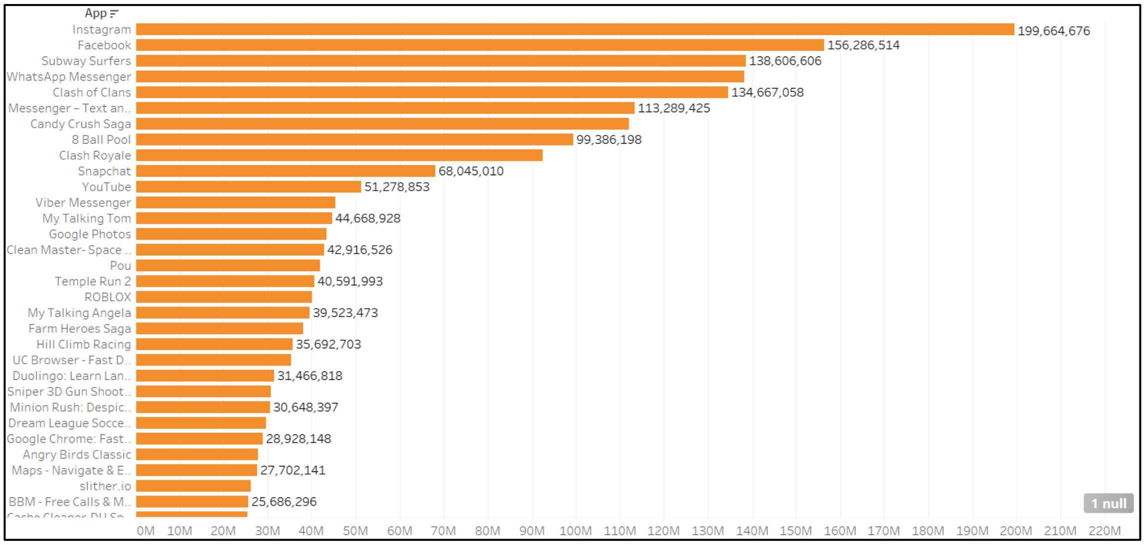


Figure 7: Most popular APP in the Market

3 RESEARCH QUESTIONS

- **Can we build a model to predict the Application rating based on the numerous attributes associated with that particular Application?**
  The response variable here is Rating which is a Continuous Variable. The predictor variable for the model would be the following variables – Number of Installs, Reviews, Category, Size etc. Models that can be built here are **Linear Regression, Random forest regression and SVM**. After model building, select the model(s) which gives the best predictive accuracy and lowest error.

- **Can we build a model to predict the number of application installs based on the numerous attributes associated with that particular Application?**
  The response variable here is Number of App installs which is a Continuous Variable. The predictor variable for the model would be the following variables – Number of Installs, Reviews, Category, Size etc. Models that can be built here are **Linear Regression, Random forest regression and SVM**. After model building, select the model(s) which gives the best predictive accuracy and lowest error.

- **Can we build a classification model to classify the apps based on the Content Rating?**
  Here, the K – means clustering model can be used to classify the model based on similar content rating like Teen, Everyone, Adults etc., Also the Support Vector Machine can be used to predict the different classes.

3.1 REGRESSION MODEL BASED ON APPLICATION'S RATING

Rating is an attribute of an application that denotes the Rating given to the application on a continuous scale of 1 to 5. With 1 being the lower rating and 5 being the highest. Below on the left is the histogram of Rating variable from the original data after imputing the missing value with mean by app category. On the right is the histogram of the rating variable after applying Box-cox transformation and changing the left skewed variable to a normalized variable with a lamda value of 4.1.

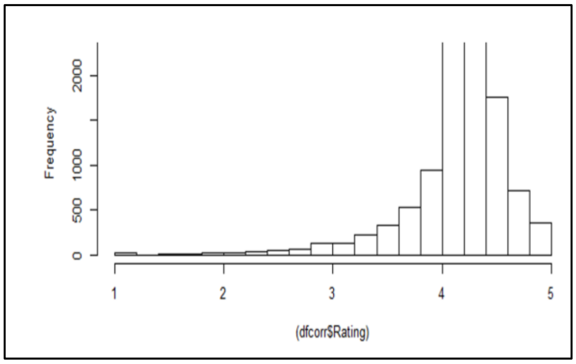


Figure 8: Histogram of Rating variable

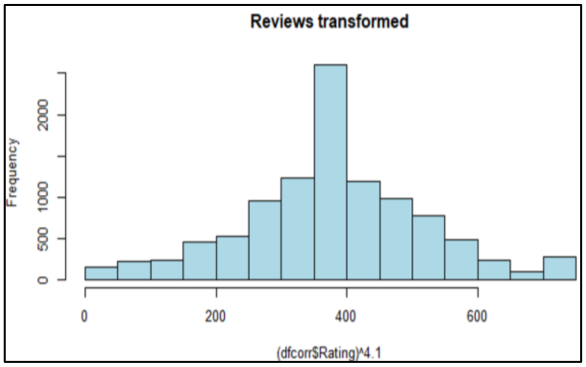
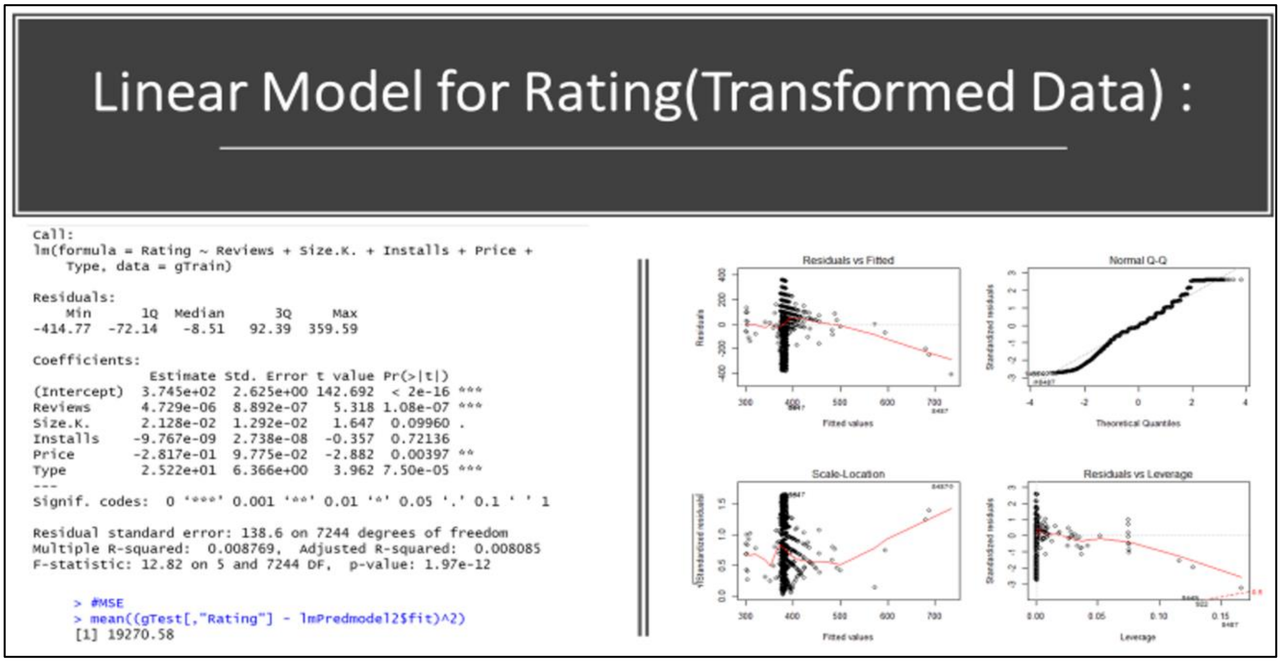


Figure 9: Histogram of Transformed Rating variable

The Review attribute is transformed and normalized. Using this we tried answering our first question i.e. Can we build a regression model to predict the applications Rating based on the numerous attributes associated with that application?

- **LINEAR REGRESSION:**
  The data is split into train and test data using a split ratio of 70 and 30. A multiple linear model is built using Rating as the Target Variable. Using Reviews, Installs, Size.K, Type and Price as predictors. Once the model is built, we used summary function to see the significance of the variables. Using Backward step also generated the same significance and below results:
  From the above summary result we can observe that Reviews and Type variable are statically significant less than the p value. Though the variables are significant for the model, but the variability of the model is very less i.e. 0.008. we can say this with only 8% confidence. Hence the model built is not a good model in predicting the Rating of the apps.



Linear Model for Rating(Transformed Data) :

```
Call:
lm(formula = Rating ~ Reviews + Size.K. + Installs + Price +
    Type, data = gTrain)

Residuals:
    Min      1Q  Median      3Q     Max
-414.77  -72.14   -8.51   92.39  359.59

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.745e+02  2.625e+00 142.692  < 2e-16 ***
Reviews      4.729e-06  8.892e-07   5.318 1.08e-07 ***
Size.K.      2.128e-02  1.292e-02   1.647  0.09960 .
Installs    -9.767e-09  2.738e-08  -0.357  0.72136
Price       -2.817e-01  9.775e-02  -2.882  0.00397 **
Type         2.522e+01  6.366e+00   3.962 7.50e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 138.6 on 7244 degrees of freedom
Multiple R-squared:  0.008769,  Adjusted R-squared:  0.008085
F-statistic: 12.82 on 5 and 7244 DF,  p-value: 1.97e-12

    > #MSE
    > mean((gTest[,"Rating"] - lmPredmodel2$fit)^2)
    [1] 19270.58
```

Figure 10: Summary Statistic of Linear Model

Also, from the right-hand side plots, we could state that the assumption of linearity is not at all satisfied. From the residuals and fitted graph plots we could see that it does not satisfy all the linearity assumptions. In the Normal Q-Q plot, the points are starting to deviate from the center and is completely deviated at both the tails. In the cook's distance leverage graph, there are many outliers, but not all outliers should be removed. In the Skewed plot, the data points are spread across. In the 1st residual vs fitted value plot; the red line is not straight. Hence, we could see that the linear model does not fit well. From the numeric values also, we could figure it out.

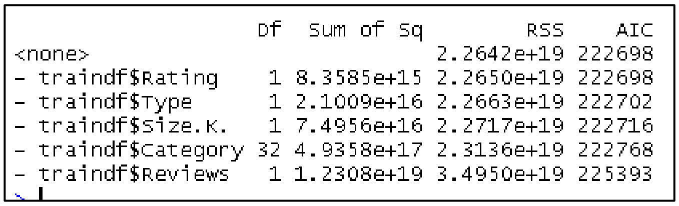
Running the model on test data gave an extremely high MSE of 19270.58. Hence, from the correlation plot, linear model assumptions, MSE and Adjusted R^2, we could say that this model is not a fit model to predict the applications Rating.

3.2 REGRESSION MODELS BASED ON THE NUMBER OF APPLICATION ISTALLS

- **LINEAR REGRESSION**

Application install denotes number of installs made for a particular App. It is a numerical discrete data type. Starting from 0 for no install to 1 billion installs.

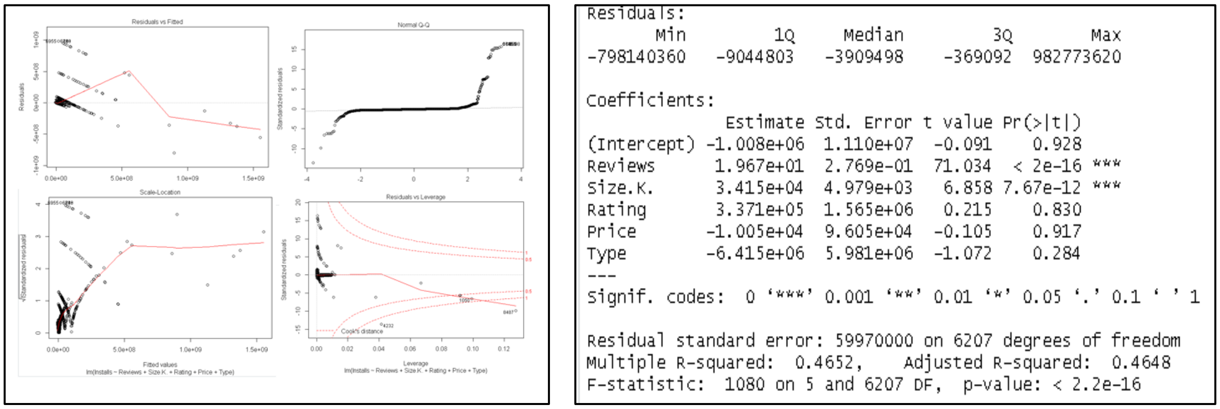
The data is split into train and test data using a split ratio of 70 and 30. A multiple linear model is built using Installs as the Target Variable. Using Category, Reviews, Size.K, Rating, Type and Price as predictors. Once the model is built, we used summary function to see the significance of the variables. Using Backward step also generated the same significance and below results:



```
                    Df  Sum of Sq        RSS    AIC
<none>                              2.2642e+19 222698
- traindf$Rating     1 8.3585e+15 2.2650e+19 222698
- traindf$Type       1 2.1009e+16 2.2663e+19 222702
- traindf$Size.K.    1 7.4956e+16 2.2717e+19 222716
- traindf$Category 32 4.9358e+17 2.3136e+19 222768
- traindf$Reviews    1 1.2308e+19 3.4950e+19 225393
>
```

Figure 11: Variable Significance

Category was not significant, so we dropped it and rebuild the model, we got an improved model. We can see that the Reviews are the most significant and Rating, Price, and Type are not.
For the first trial we model we got R-square value of 0.3828 and Residual standard error of 60550000, this high RSE is because of the target value which has 0 to 1billion values, but the low 0.3828 is low. So, after doing feature selection we rebuild our model with removing the Category. We saw a significant improve in our model with reduce in RSE value and increase in R-square to 0.46 which is better that the previous value of 0.3828.



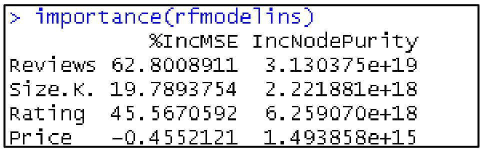
```
Residuals:
      Min        1Q    Median        3Q       Max
-798140360  -9044803  -3909498   -369092  982773620

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.008e+06  1.110e+07  -0.091    0.928
Reviews      1.967e+01  2.769e-01  71.034  < 2e-16 ***
Size.K.      3.415e+04  4.979e+03   6.858 7.67e-12 ***
Rating       3.371e+05  1.565e+06   0.215    0.830
Price       -1.005e+04  9.605e+04  -0.105    0.917
Type        -6.415e+06  5.981e+06  -1.072    0.284
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 59970000 on 6207 degrees of freedom
Multiple R-squared:  0.4652,    Adjusted R-squared:  0.4648
F-statistic:  1080 on 5 and 6207 DF,  p-value: < 2.2e-16
```

Figure 12: Summary Statistic of Linear Model

The units of residual standard error are precisely those of the response or outcome or dependent variable, that in our case is Install (which has 0 to 1 billion values), so the large value of RSE is 64285581 which for this data is quite normal. The R – square value will justify the RSE value, which in our case is week.
Also, Reviews and Size has impact on Installs. It can be noticeable that Install increases 1.967e^1 unit when Reviews is one unit and Increases 3.415e^4 unit with 1 unit increase in Size.

- **RANDOM FOREST**

Random Forest is an ensemble learning method for regression and classification. A random forest regression model is built for the target variable Installs, considering Reviews, Size.K., Rating and Price as predictor variables. The dataset is split into training and test data in the ration 70:30. The total number of trees built are 500. The percent of variance obtained over train data is 78.8%, which is better result. It is observable that Reviews and Size.K. are the important predictors in predicting target variable.



```
> importance(rfmodelins)
          %IncMSE IncNodePurity
Reviews 62.8008911  3.130375e+19
Size.K. 19.7893754  2.221881e+18
Rating  45.5670592  6.259070e+18
Price   -0.4552121  1.493858e+15
```

The root mean squared error (RMSE) and residual square (R2) obtained in this case are 37995025 and 0.65 respectively. The model can predict the target variable with an accuracy of 65%.
Below is the plot of Out-of-bag error and the number of trees. The error decreases with the increase in the number of trees built in the model.

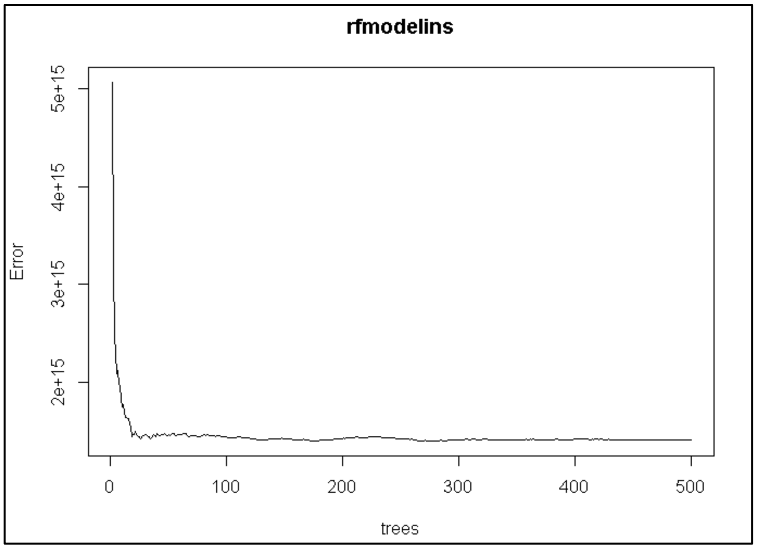
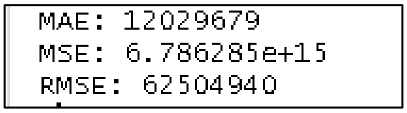


Figure 13: Out-of-bag error vs number of trees

- **SUPPORT VECTOR MACHINE**

A Support Vector Machine (SVM) is a supervised machine learning algorithm, which can perform both regression and classification. An SVM regression model is built considering Installs as the target variable and Reviews, Size.K., Rating, Price are the predictor variables. The data is spit into proportions of 0.70 and 0.30 as training and validation data, respectively.



```
MAE: 12029679
MSE: 6.786285e+15
RMSE: 62504940
```

The mean squared error (MSE) and root mean squared error (RMSE) obtained are 6.786e+15 and 62504940 respectively, which are significantly high. The number of support vectors built in this case are 597. Below is the plot which shows the predicted data points.

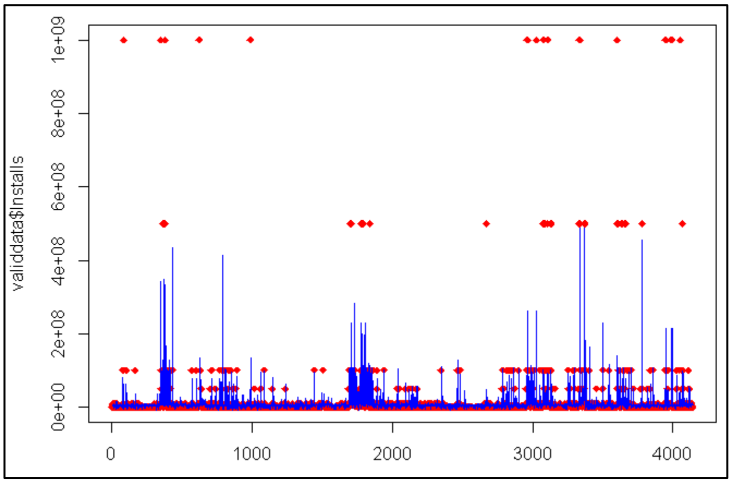


Figure 14: Data points of predicted values

3.3 CLASSIFICACATION MODELS BASED ON APPLICATION'S CONTENT RATING

Content Rating is an attribute of an application that denotes the age category the App is suitable for. The Attribute has the following Values. Can we build a classification model to classify the apps based on the Content Rating? The following Models were built to predict and classify the Content Rating

Content Rating	**Description**
Everyone	Content is generally suitable for all ages
Everyone 10+	Content is generally suitable for ages 10 and up
Mature 17+	Content is generally suitable for ages 17 and up
Adults only 18+	Content suitable only for Adults 18+
Teen	Content is generally suitable for ages 13 and up
Unrated	Rating is pending

Table 2: Content Rating Values

- **K -MEANS CLUSTERING:**

  K Means algorithm is one of the simplest unsupervised Machine Learning algorithms. It aims to cluster a collection of data points based on certain similarities. The task here is to group the Applications based on similar Content rating determined by the individual attributes of the Apps. The dataset contains 6 different levels.

  As a first step in order to determine the number of optimal clusters the elbow graph was plotted. The **elbow method** runs **k-means** clustering on the dataset for a range of values for **k** (say from

1-10) and then for each value of **k** computes an average score for all clusters. Based on the graph shown below the optimal number of clusters was determined to be 3.

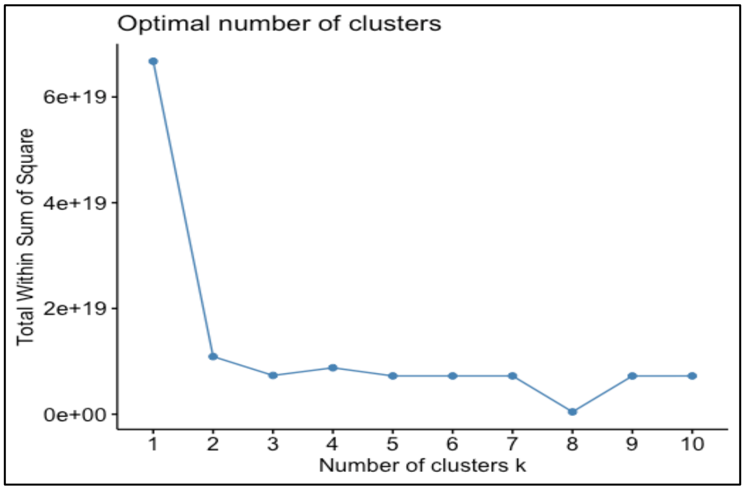


Figure 15: Optimal Number of Clusters

Cluster	Adults only 18+	Everyone	Everyone 10+	Mature 17+	Teen	Unrated
1	3	7858	320	415	1007	2
2	0	446	50	30	115	0
3	0	77	7	2	24	0

Table 3: Cluster Data

INFERENCE:

K-Means has split the data into the following clusters as outlined in the Table 3. It is seen that there are good number of observations in each cluster with Cluster 1 having the highest number of apps. There are no similarities of dissimilarities in each cluster indicating that the model is not suitable for this dataset or classification.

- **SUPPORT VECTOR MACHINE:**

SVM is a machine learning model that can be used for regression and classification tasks. The objective of the support vector machine algorithm is to find a hyperplane in a dimensional space (N — the number of features) that distinctly classifies the data points.

The SVM model being built here was to classify the 6 classes of Content rating by building n number of hyperplanes. As a next step, the data was split into test and training based on a 70:30 ratio split.

In this model, based on trial and error methods the following arguments were given

a) **Cost (C):** C here defines the cost of misclassification. A large C gives low bias and high variance. A small C gives higher bias and lower variance. For this model the Cost parameter was set at 10

b) **Gamma:** Intuitively, the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors. For this model, the gamma value is 0.1 [2]

Actual/Predicted	**Adults only 18+**	**Everyone**	**Everyone 10+**	**Mature 17+**	**Teen**	**Unrated**
Adults only 18+	0	2	0	0	0	0
Everyone	0	2078	0	1	15	0
Everyone 10+	0	77	11	0	1	0
Mature 17+	0	115	0	2	1	0
Teen	0	273	0	0	12	0
Unrated	0	1	0	0	0	0

Table 4: Actual Vs Predicted Class

The total number of Support vectors built was 4032. The table above summarizes the Actual and Predicted Class for the 6 different classes. The SVM model predicted the Content Rating with an accuracy of 80%.

4 CONCLUSION:

The regression models built for predicting Installs using Linear regression and Support Vector Machine did not significant results in estimating dependent variable. However, Random Forest gave better accuracy of 65% in predicting Installs.

The most important variables for prediction were the Size of the Application and the number of reviews given by the users.

For the Research question 3 stated above, The K means clustering model did not yield positive results with the data points between the different cluster not being similar or dissimilar in any ways. On the other hand, the SVM model predicted the Content rating with an accuracy of 80% between the Actual and Predicted class.

5 REFERENCES:

[1] “Google Play.” *Wikipedia*, Wikimedia Foundation, 10 May 2020, en.wikipedia.org/wiki/Google_Play.

[2] “RBF SVM Parameters¶.” *Scikit*, scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html.

[3] Donges, Niklas. “Data Types in Statistics.” *Medium*, Towards Data Science, 8 Oct. 2019, towardsdatascience.com/data-types-in-statistics-347e152e8bee.

[4] Gupta, Lavanya. “Google Play Store Apps.” *Kaggle*, 3 Feb. 2019, www.kaggle.com/lava18/google-play-store-apps.