32513 Machine Learning

Assignment 2: Practical Machine Learning Project

Yue Xiao 13130760

Haoran Li 13122852

Dataset: <https://github.com/HaoranLi6/UTS_ML2019_ID13122852/blob/master/googleplaystore.csv>

Python Code:

<https://github.com/HaoranLi6/UTS_ML2019_ID13122852/blob/master/ML2019_assig2_coding.ipynb>

**Content**

[Introduction 2](#_Toc20252618)

[1. Exploration 3](#_Toc20252619)

[1.1 Data Description 3](#_Toc20252620)

[1.2 Data Features 3](#_Toc20252621)

[2. Methodology 7](#_Toc20252622)

[2.1 Data Processing 7](#_Toc20252623)

[2.2 KNN 9](#_Toc20252624)

[2.3 Random Forest 10](#_Toc20252625)

[3. Evaluation 13](#_Toc20252626)

[4. Ethical 13](#_Toc20252627)

[Conclusion 14](#_Toc20252628)

# **Introduction**

Since the mobile devices are invaded into people's daily life now, the number of app users are increased rapidly, which cause the growth of app developers correspondingly. The dominant Android operating system gains approximately 80% market shares worldwide, and most apps are housed in the Google Play Store, which reflects that the Google Play Store is the biggest and famous app stores. Therefore, how to develop an app that can attract more android users are crucial for all app developers to consider. This paper focuses on predicting the apps' rating data and figure out how the rating data have been influenced by other attributes, which will assist developers to get an insight of app's details and make more wise decisions when developing an app.

The data source named Google Play Store Apps that we obtained from Kaggle, which contains about 10k apps data. The data of Each app includes basic information about app price, rating scores, category, etc. There is a total of 13 attributes and 10,841 entries contained in this dataset. To figure out the rating data, first, the dataset features are analyzing by using data visualization tools, such as pie chart and histogram. Then based on the understanding of whole dataset shape, the KNN and Random Forest algorithms are applied to do the modeling, and the results are compared in the evaluation section.

# **Exploration**

## 1.1 Data Description

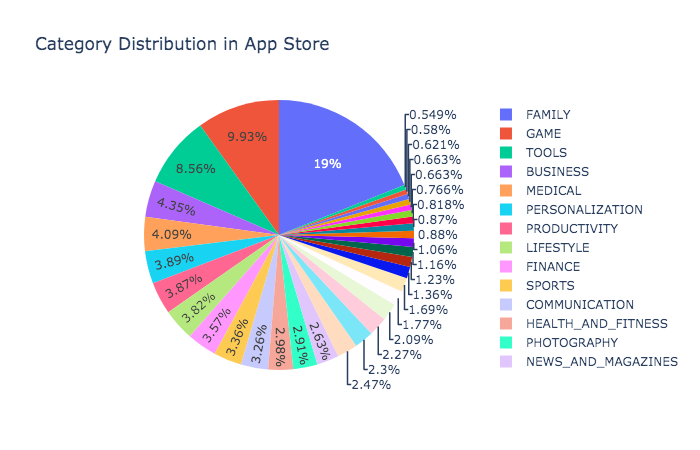
This dataset contains information of applications in google app store. The attributes of the application include App (name of application), category, rating, reviews, size, installs, type, price, content rating, genres, last updated, current version and android version. The dataset contains information of 10,841 applications in google store.

|  |  |
| --- | --- |
| **Attribute** | **Description** |
| APP | The name of google store applications. |
| Category | Different categories of applications. |
| Rating | The overall rates of users for the app. |
| Reviews | The number of reviews by users. |
| Size | The size of the application. |
| Installs | The amount of installments by users. |
| Type | Whether the application is free or not. |
| Price | The price of the application. |
| Content Rating | Age group the app is targeted at - Children / Mature 21+ / Adult |
| Genres | An app can belong to multiple genres (apart from its main category). |
| Last Updated | Date when the app was last updated on Play Store (as when scraped) |
| Current Ver | Current version of the app available(as when scraped) |
| Android Ver | Current android version of the app(as when scraped). |

# 

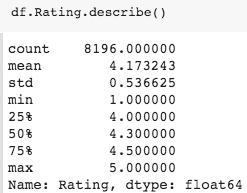
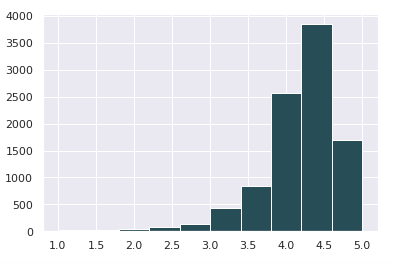
## 1.2 Data Features

Before analyze the number of apps in the store per category, the total number of category have been defined and a category named ‘1.9’ was deleted to make the result more accurately. Form the figure 1, it can be seen the top 5 category is family apps, game, tools, business and medical apps.



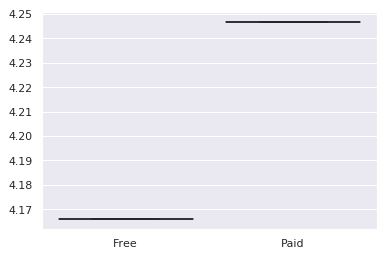
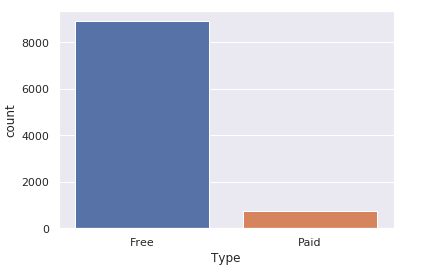
*(Figure 1)*

Looking at the rating attribute, there are 1463 null values included in rating. The null values are replaced by the mean value of their respective category rather than replaced them with the mean of rating, due to the null values rate are quite high and the result will be distorted. Then, it can be seen that the average rating of the apps in store is approximate around 4 to 4.5, which is high.



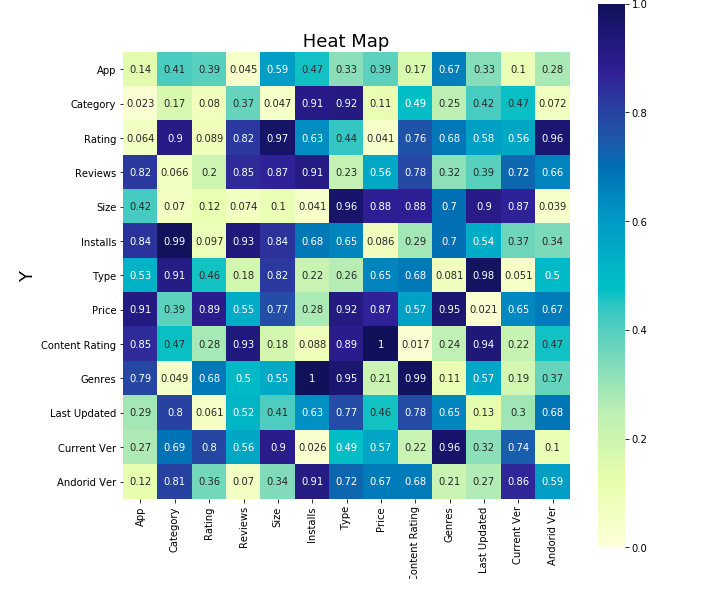
*(Figure 2)*

From the figure 3, it can be seen that the apps are divided into two types, free and paid. There are 8,902 apps are free and only 756 apps need to be paid. the boxplot demonstrates the rating mean of each type of the apps. The paid apps have the higher rating than free apps.



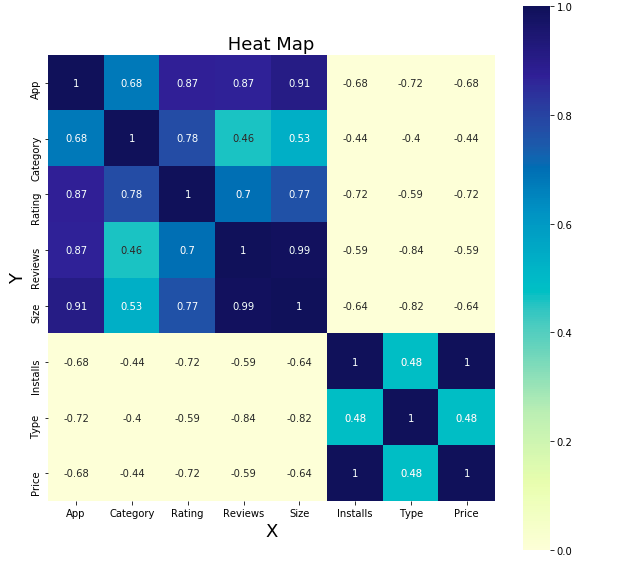
*(Figure 3)*

The figure below shows the correlations among these 13 attributes, which produced based on the random data matrix. To ensure the accuracy of the correlation results, the input data has been cleaned that replaced the null values and some abnormal values to decrease the negative influences of outliers.



*(Figure 4)*

The figure 5 demonstrates the correlation that produced based on the correlation coefficient of each attributes. The blue square blocks present the positive relations between the attributes and the negative correlations are shown by the light yellow blocks. Based on the analysis of correlation, it can be found that how the rating data related with others.



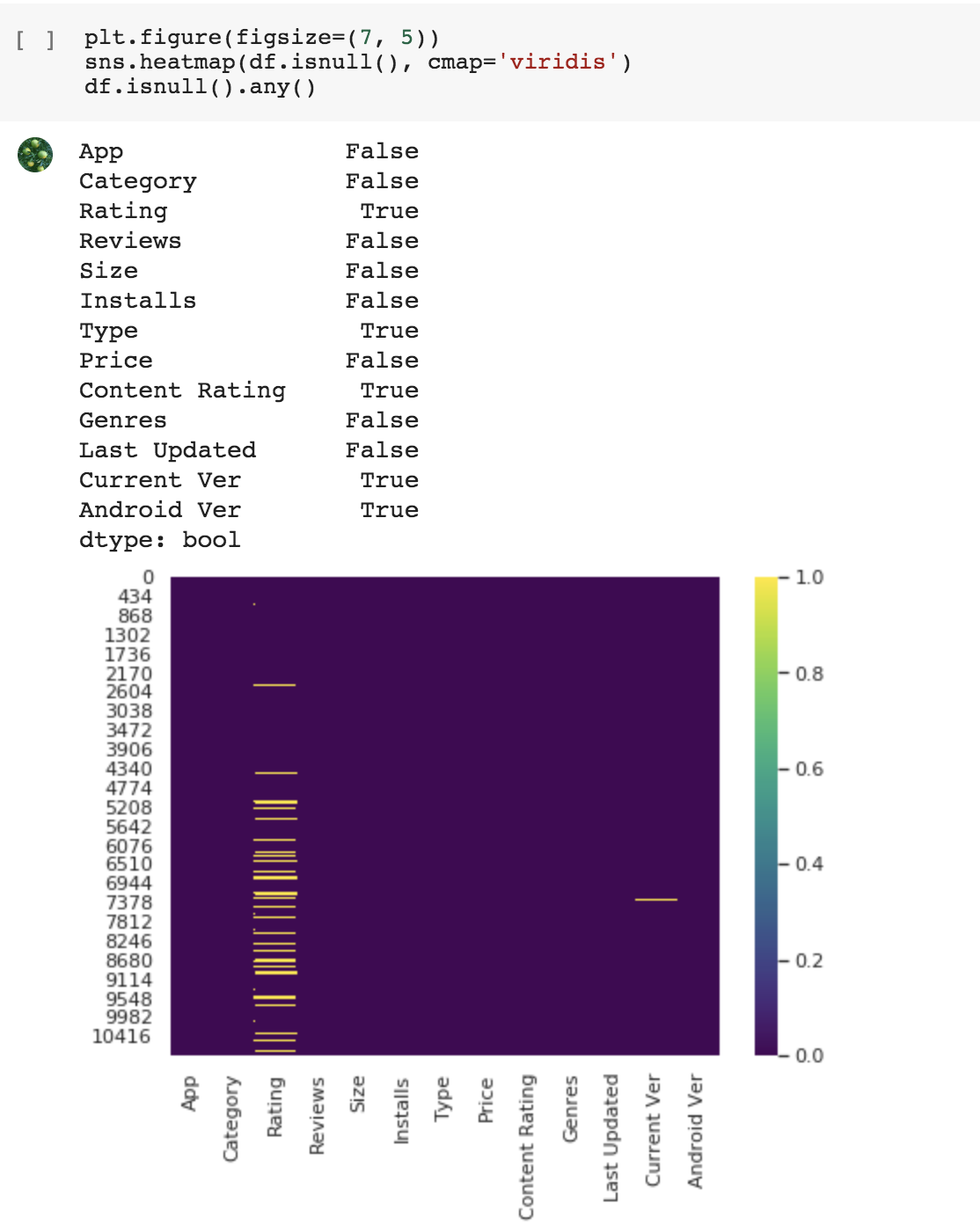
*(Figure 5)*

# **Methodology**

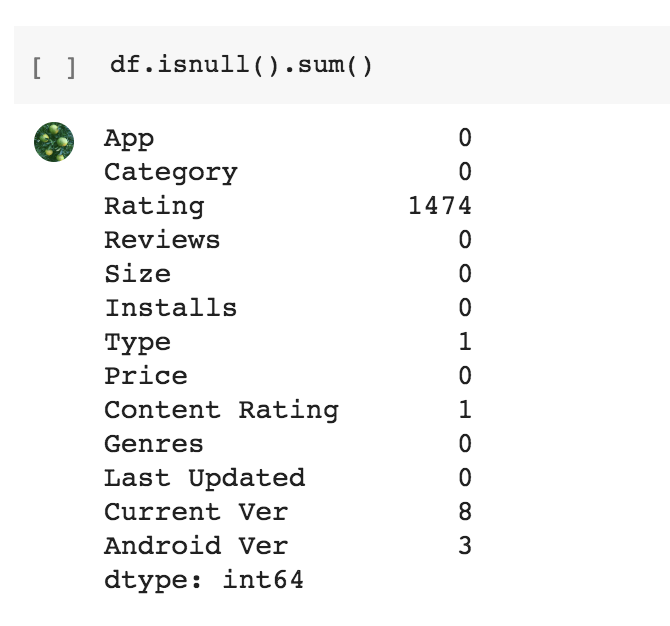
## 2.1 Data Processing

This research choses two algorithms, K-nearest neighbor and Random Forest to predict the rating of applications. In the dataset, there are 13 attribute. Missing data of each attribute should be examine and cleaned before we apply the algorithm to the dataset.

The figure 6 shows the code of python to check whether there is missing value or NaN within the dataset. It is obvious that the column rating, type, content rating, current version and android version have missing value. Figure 7 is the codes of python to find out the number of missing values in each attribute. We can see that rating has the most missing value which equals to 1474.

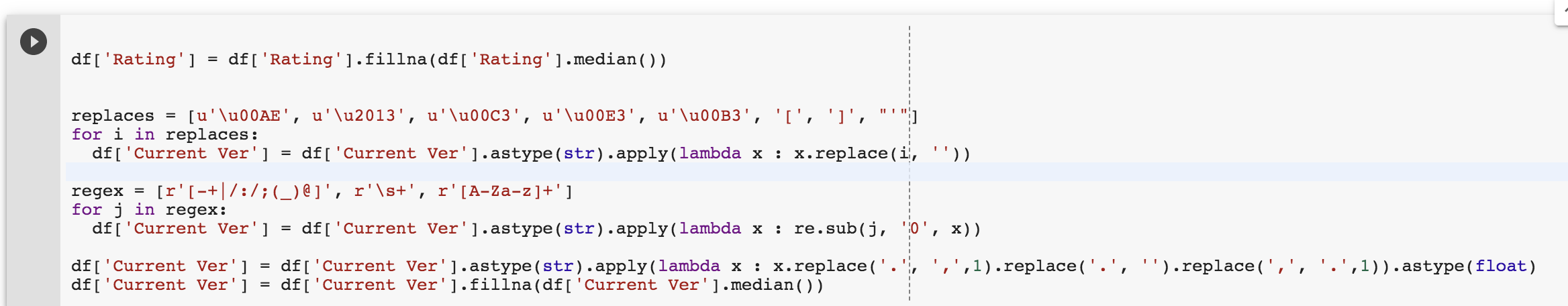


*(figure 6)*

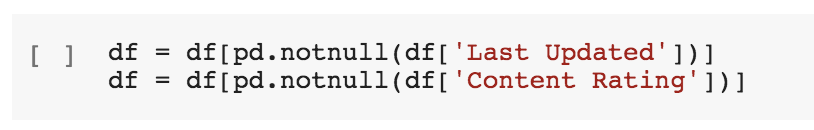


*(figure 7)*

To deal with missing value, the method we use in this case is to replace the missing value by the mean value of each attribute. Besides, some non-numeric attributes which have missing should be cleaned before the replacement, and then remove the missing value. Figure 8 shows the codes of the data cleaning step, while figure 9 is the code of removing NaN values.

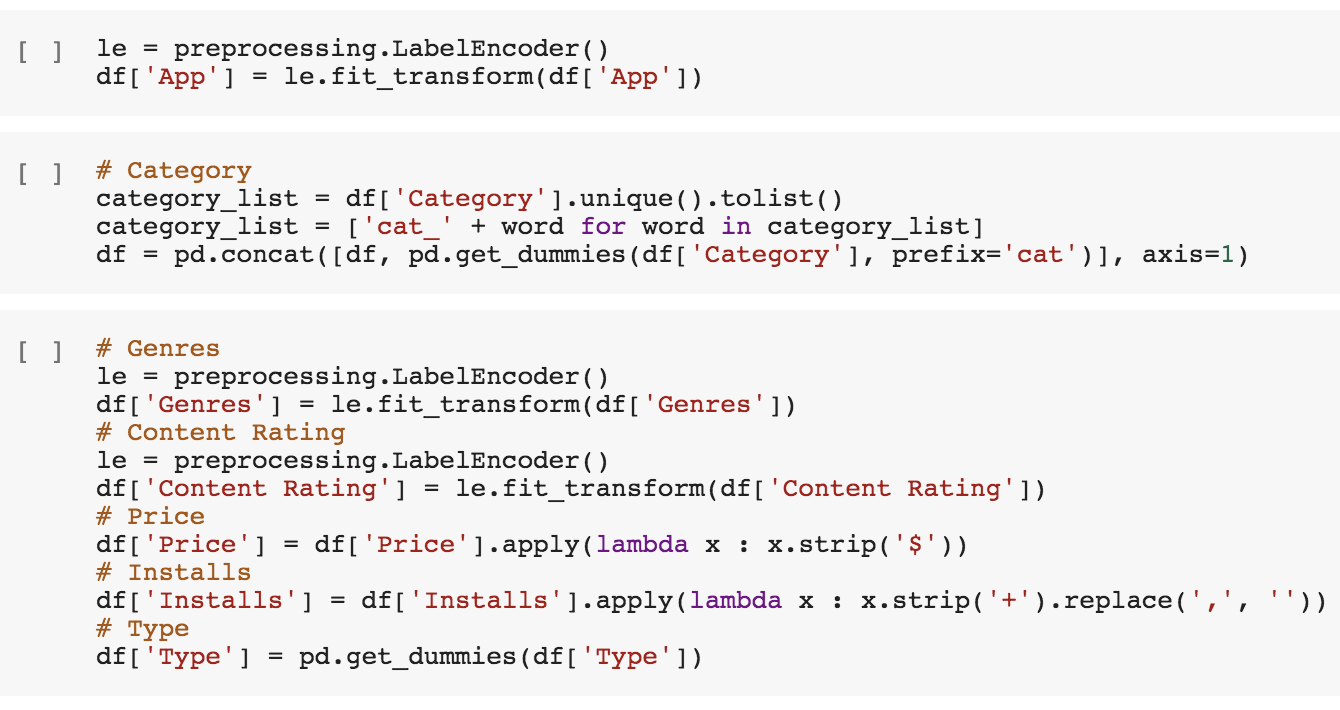


*(figure 8)*

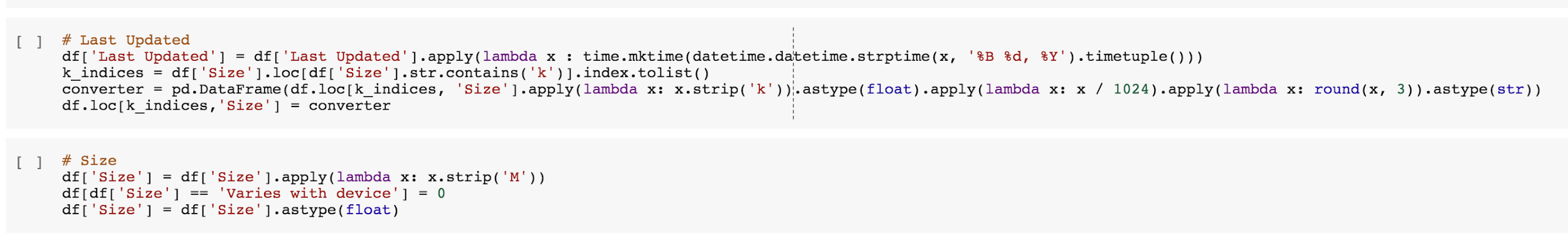
**

*(figure 9)*

After basic data cleaning, we need to transfer the non-numeric attributes into numeric ones. The reason is that some algorithms do not support those kinds of values. The figure below shows the codes of encoding attributes.

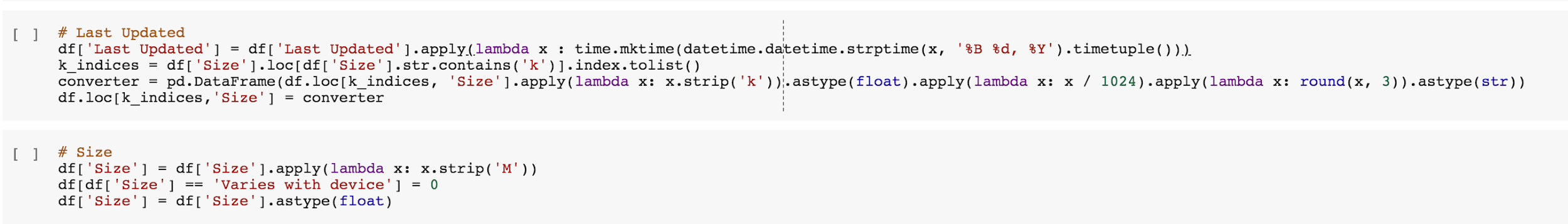


*(figure 10)*



*(figure 11)*

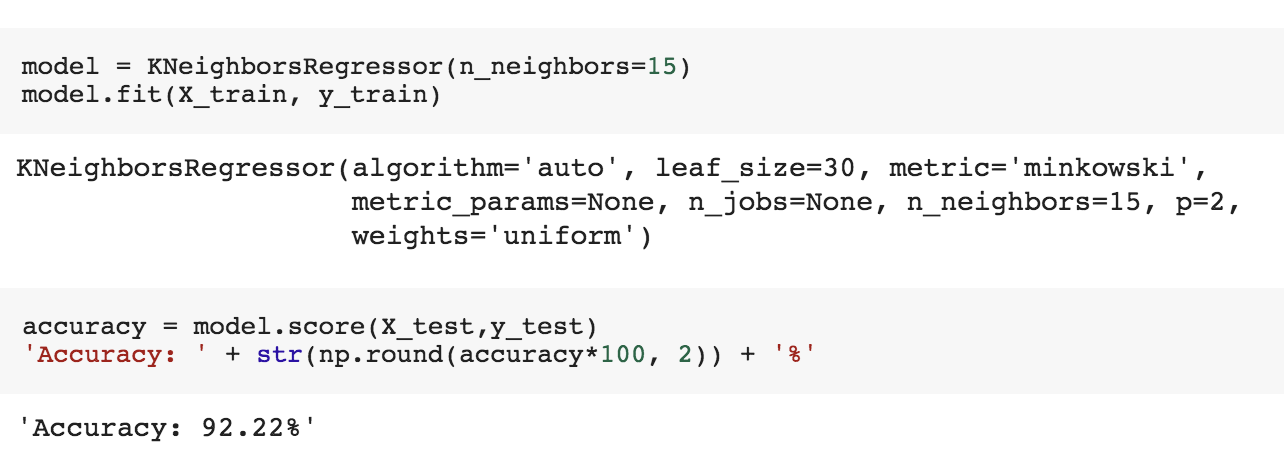
Current dataset needs to be splitted into testing set and training set, and then use the two algorithms to do the prediction. The python codes in figure 12 present how to divide the dataset into two. And we split the dataset into 75% training set and 25% testing set.



*(figure 12)*

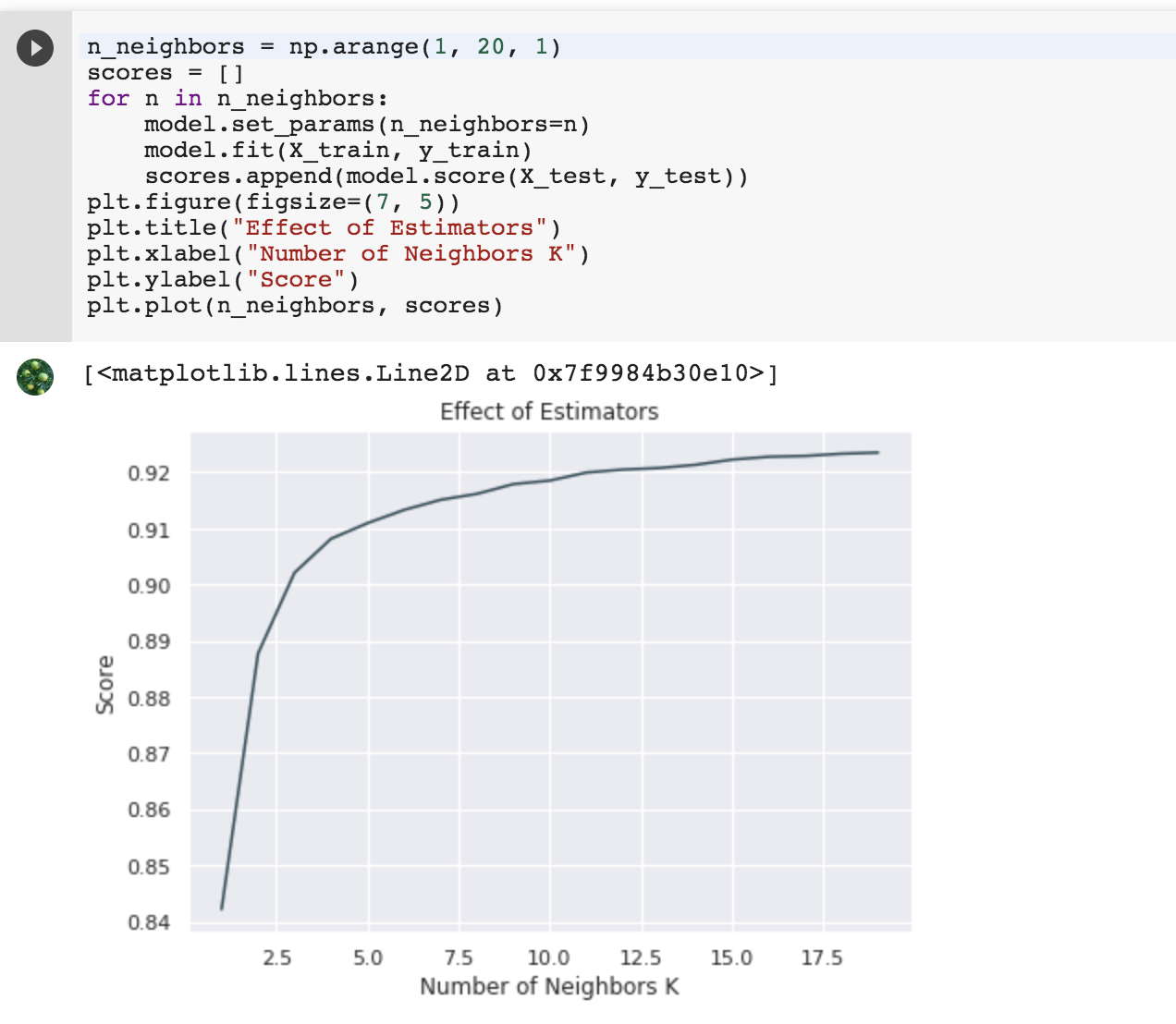
## 2.2 KNN

K-nearest neighbor is one of the most widely used clustering algorithm. This method could compute the distance between each value, take the first train value as the nearest point and then predict the class by the frequent class label. The first line of figure 13 refers to choose the 15 closest neighbors, and then use the x\_trian and y\_trian to calculate the mean accuracy. The accuracy equals to 92.22% in this case which means that the prediction is quite reliable based on the testing set.



*（figure 13）*

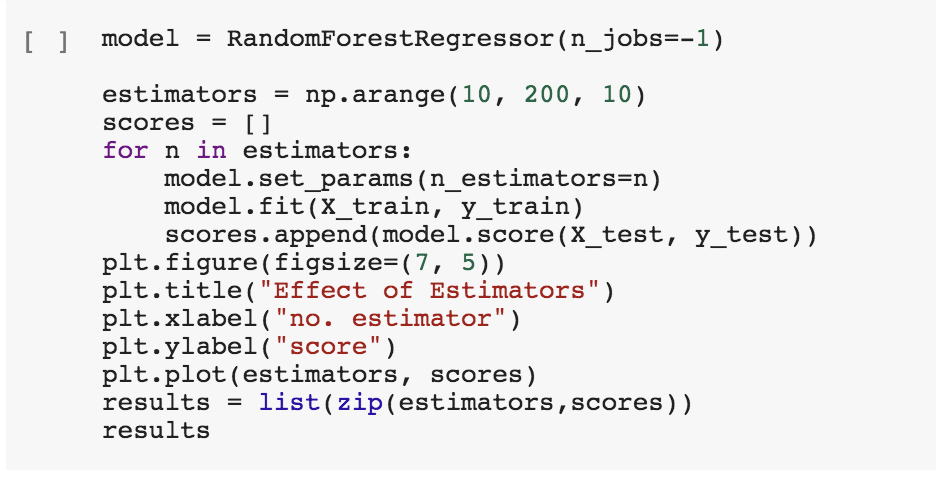
Moreover, various estimators are calculated in the next step. Figure 14 contains codes of testing different estimators and diagram of the relationship between number of neighbors and score. The diagram shows that when the number of neighbor increases, the score will increase as well.



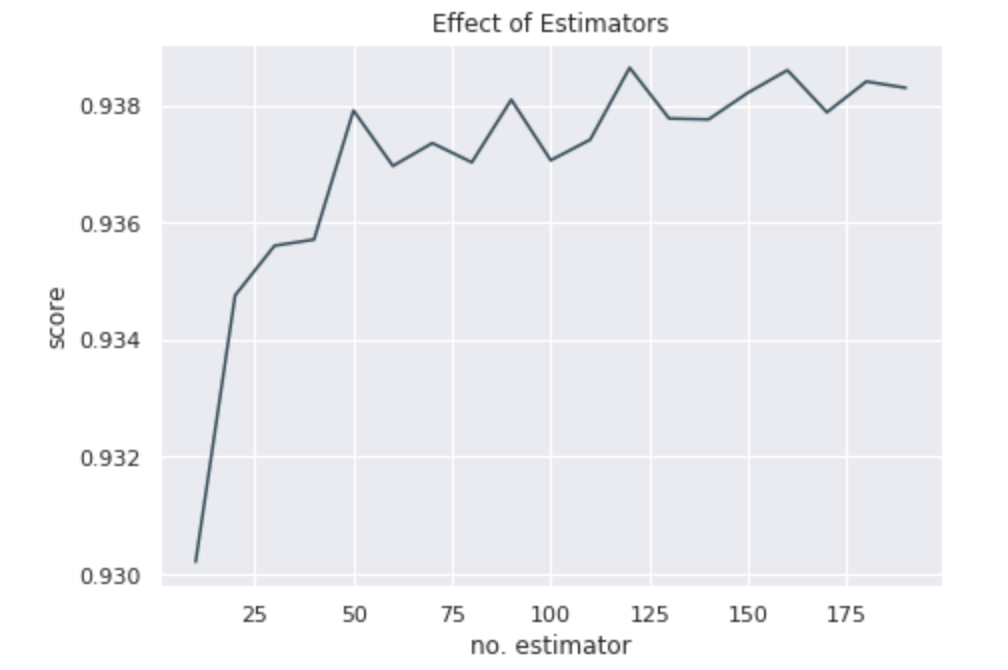
*(figure 14)*

## 2.3 Random Forest

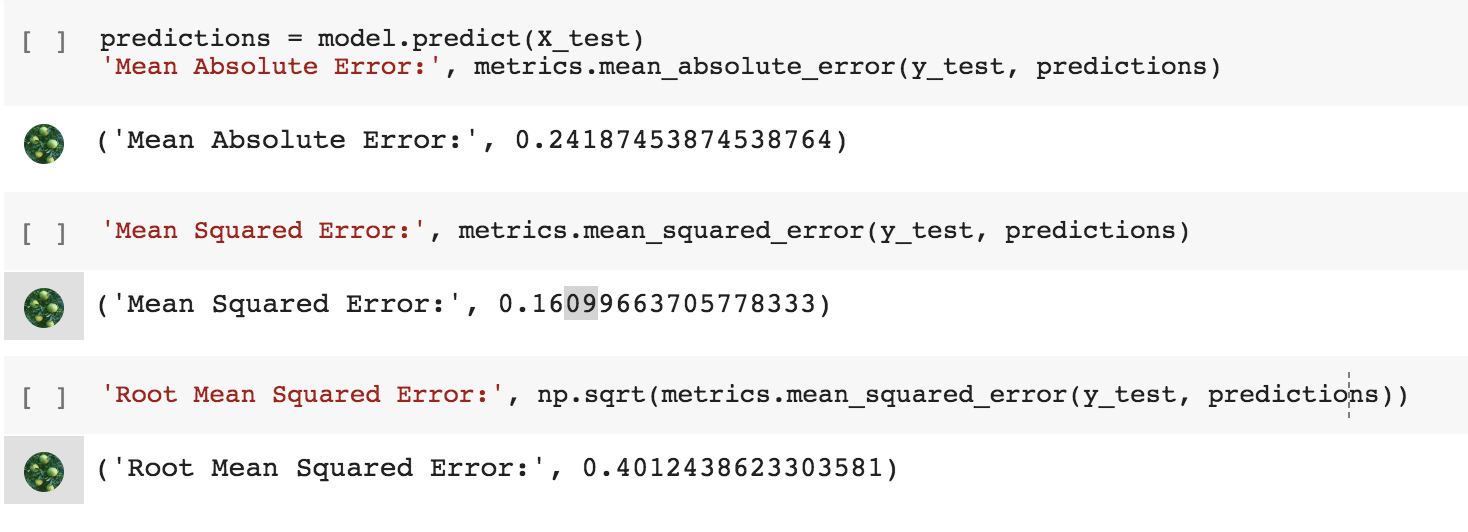
Random forest is one kind of algorithm that could collect classification and find out the most voted prediction. This method could produce highly accurate classifiers, handle large of input variables, and produce an unbiased estimate of the error after generalization. The number of trees is defined by the code in figure 15. The range number of estimator is between 10 to 200. And figure 16 is a diagram which shows the relationship between number of estimator and score. With the incremental of no. of estimator, the score various by the estimators. The relationship is not linear.



*(figure 15)*



*(figure 16)*

The accuracy of the prediction using random forest could be calculated in three ways, mean value, mean squared and root mean squared. Figure 17 display the python code of applying each method and their accuracy. We can see that mean squared random forest has the highest accuracy at 84%, while root mean squared method has the lowest one at 60%.

*（figure 17）*

# **Evaluation**

This report tries to use different algorithms to predict the rates of applications in google store. Random forest and K-Nearest Neighbour are the two algorithms that we choose to analysis the data and predict the rating of applications. The result shows that the random forest has a higher accuracy rate. K-Nearest Neighbour is not suitable for large dataset because this kind of algorithm only computes the "nearest" neighbor samples. If the number of samples in a class is large, or the samples are not close to the target sample, or such samples are very close to the target sample, then the sample will be an new input and large-capacity class among the K neighbors of the sample will also be the majority. Try different methods that could help us get better predicting results. Besides, this research could be better improved in the future. For example, the attributes correlation could be better examined. This paper use heatmap to visualize the correlation of each attribute. But the result of heatmap could be influenced by the dimensions that we chose. The correlation rates will be various from the dimension. Therefore, how to better perform the research of correlation between each attributes considered later.

# **Ethical**

This paper aims to predict the rating data of apps in store and assists app developers to decide the future development of app category and details. While, during the processes of addressing the rating data-related problems, it can be found that almost 95% apps’ target users are teenagers and above, and the install data shows that family, game, tools, and communication apps are the most popular categories. Teenagers, family (parent-child interaction) and game are the keywords of the rating contributions, so the ethical issues will raise to all the app developers that do not produce poison game apps for their selfish ‘happiness’ and also consider how to bring positive effects to be young.

Utilitarian approach refers to measure whether the behaviors could promote happiness. An act that is conducive to the promotion of happiness is correct; if it leads to something contrary to happiness, it is wrong. In this case, most applications target teenagers. But some news and reports indicate that teenagers are easy to addict to games and applications on the mobile phone. The actions of those companies that aim to attract more teenager users will lead to the declining of grade and addiction to online applications. Therefore, the application targeting teenagers not in line with promoting happiness.

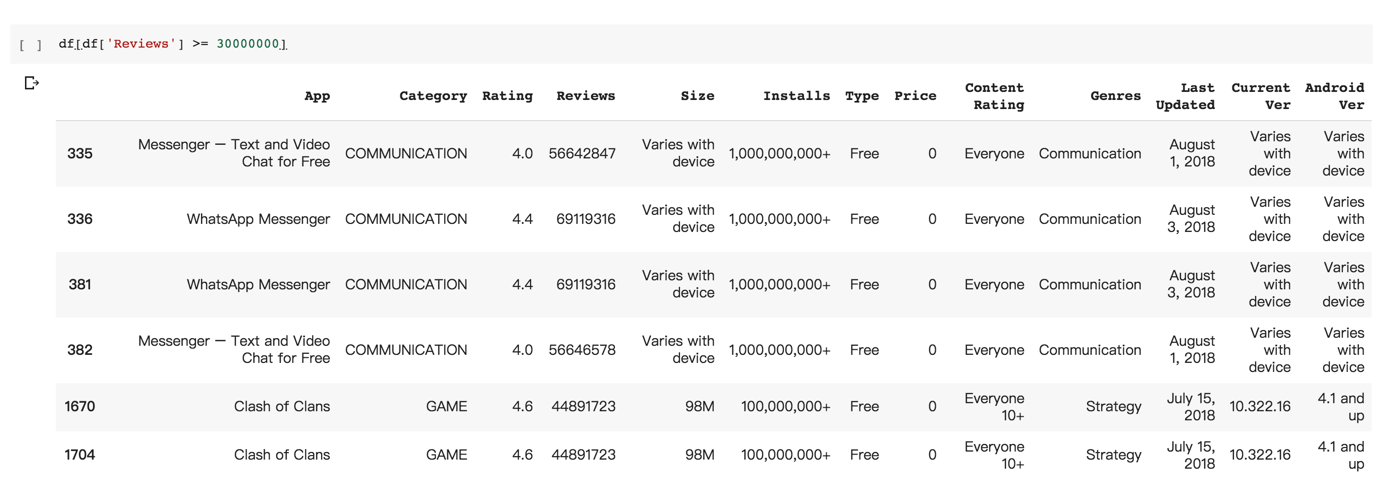
Different from the utilitarian approach, the Kantian duty-based approach focuses on principles and responsibilities that require each person’s behaviors have to follow the rules. Therefore, based on this approach, each developer has to keep an appropriate moral behavior principle framework in mind when they are developing apps. For developers, the purpose of developing an app should not relate to their intentions and desires. Thus, the normal game apps should not make teenagers and even children addicted with, and the communication apps should not be the place for the illegal erotic trading, and also some tool apps cannot be a method to track others. So, based on the duty-based approach, developers need to follow world principles and do not see people as the purpose of developing apps.

# **Conclusion**

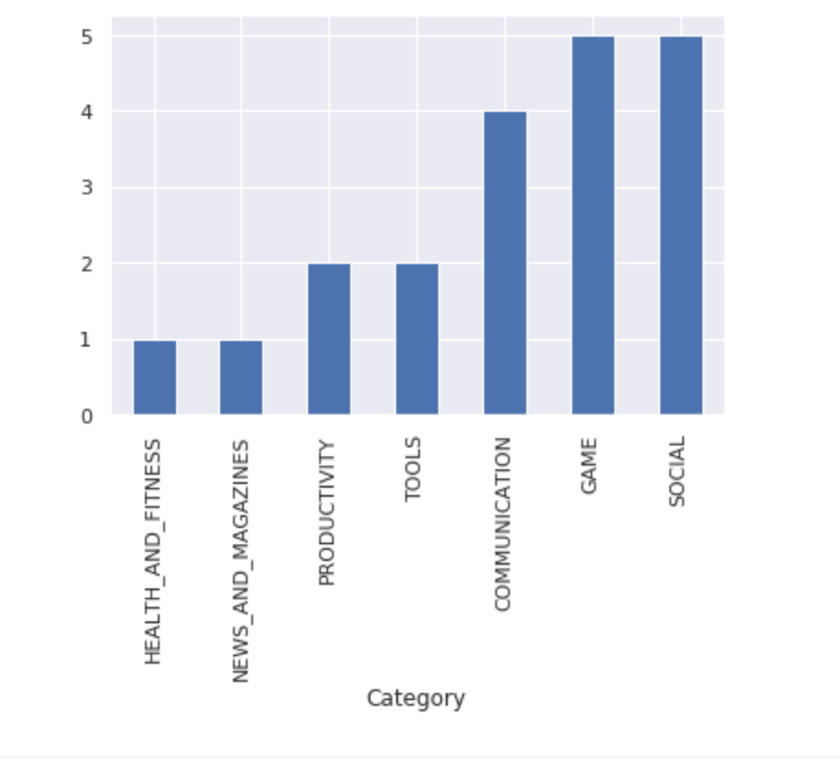
This report aims to find out how to get higher application rating by examining the dataset of google store applications. The dataset contains 10,841 data and is applied to two algorithms which are K-Nearest Neighbor and Random Forest to predict the rating. The research use python to do the data cleaning, delete the missing value and process the data. The result shows that K-Nearest Neighbor performs better than Random Forest in predicting the rating because the accuracy of K-nearest Neighbors is 92% which is higher the accuracy of Random Forest. Moreover, this report contains the discussion of ethical aspects of the usage of the application.

**Appendix**

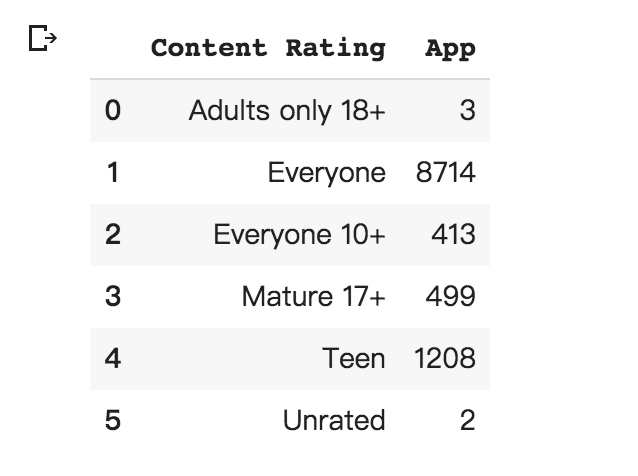
1. Find out applications whose review are greater than 30milion.



1. Histogram of Install group by category.



1. Apps group by content rating



1. Number of apps in different content rating.

