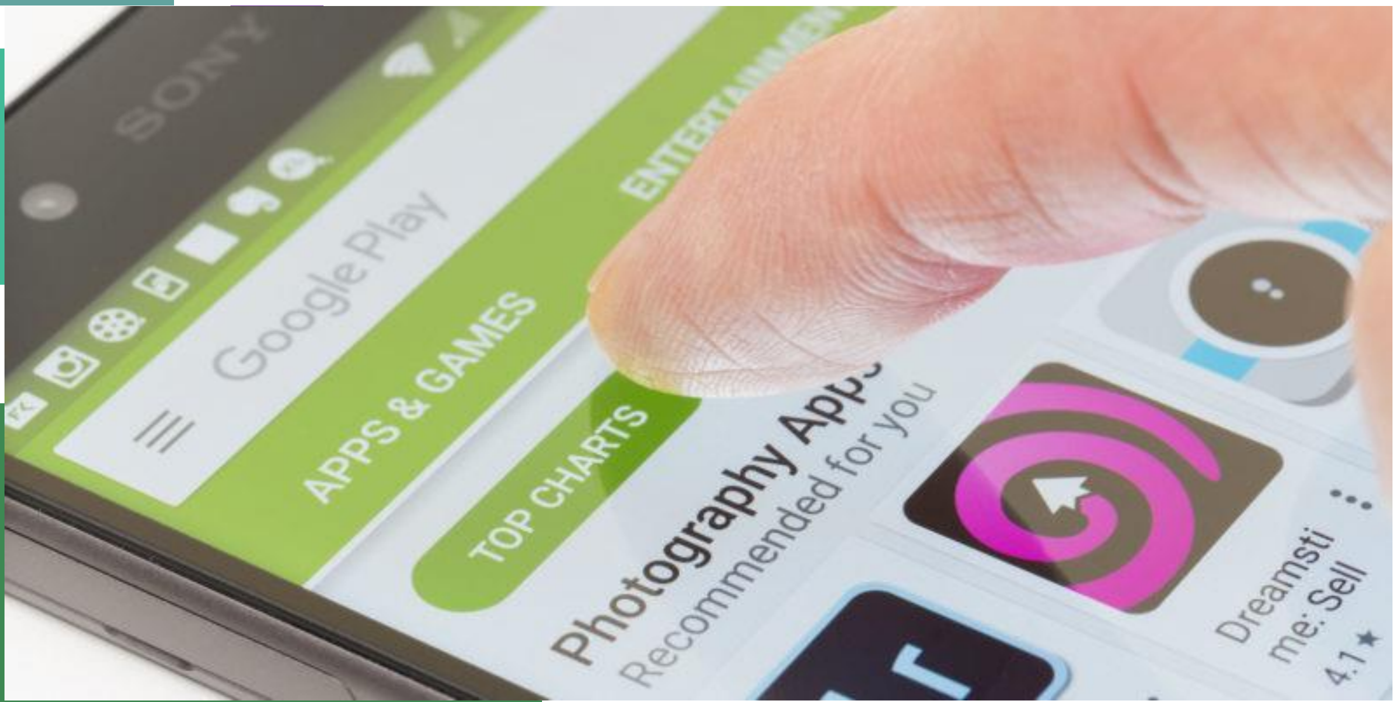

Assessment task 3: Research Paper

Google Play App Rating Prediction Report

42048 Studio 3: Innovation - Autumn 2022

Luyang Zheng (14102228)
Thanh Thuy An Tran (14218768)
Rabboni Arjonillo (14107938)
Junyan Lai (14111823)
Xiaotian Wu (12648301)

Table of Contents

1. INTRODUCTION

1.1 Business context

The development of technology and wide adoption of mobile phones have accelerated the demand for mobile applications. Google Play Store is one of the most popular app stores with 3.48 million Android apps which provides users with both free and paid access (Statista, 2018). Since the Google Pay Store has millions of apps with different functionality, smartphone users can benefit from the diversity of choices. However, in the case of designing and selling software applications, it has been a challenge for app developers. This is because apps exist in digital space and can be easily transferable across regions, the app developing industry is highly saturated and competitive. Everyday there are more than 3500 apps added on Google Play Store so to create a smartphone app that stands out is extremely difficult. Apart from allowing users to download apps, this online platform also allows users to share their experiences in the form of numeric app rating and text review. App star rating is the average rating of all the rating given by users (Mahmood, 2019). According to Xiao & Li (2019), online review has a significant impact on consumers' decision-making process. Therefore, consumers are attracted to the apps with high ratings as they are likely to have better quality.

1.2 Main problem

Therefore, in this paper, we will examine number of mobile application features to identify the influence of these factors on app star ratings. A model of app rating prediction will also be built to estimate numeric user review once the app is created. With thousands of apps currently being in Google Play Store with little to no exposure to users from the sheer number of similar applications by more marketed developers, this study will help pull focus to attributes in the form of features of an app to appeal more to users.

1. 3 Motivation

Considering that high app ratings is the main factors in obtaining new users, this paper will discover software application factors influencing consumers' review to improve quality of product design. We also create an app rating prediction to help designer estimate app rating prior to launching the product. The app rating prediction we are building will help improve not only ratings of an app, but also the overall standards of designing and developing apps.

1. 4 Contributions

In this paper, our main objective is to evaluate the impact of different variables on app star rating, therefore coming up with a recommendation system for app design. For this purpose, we collected the dataset of Google Play Store app rating from Kaggle which include web-sourced data of 10k play store applications. We then perform data cleaning to get the final dataset used in our machine learning model. We also perform exploratory data analysis in terms of app categories, sizes, number of instalments, prices, app types and content ratings with the app star rating to discover the correlation between each variable. Random Forest (RF) machine learning model will be applied to find variable importance and Root Mean Square Error (RMSE) will be used for model evaluation work. The thought process of the group for using the Random Forest instead of just utilizing Decision Trees is although it is a bit more complex and requires a decent more work, it provides more accurate results and removes the possibility of overfitting, which will be tested by RMSE.

Random Forest results indicate that feature importance heavily relies on the version of android devices more so than the price as initial testing indicates. Through initial categorical analysis, price, number of reviews, and size proved that there was causal relationship between them affecting app ratings. However, using RF demonstrates that these correlations are biased as the model as whole in **Figure 27 Variables Importance** showed a bigger picture of the analysis. RF already in its nature reduces overfitting as previously mentioned and an RMSE of 0.47 proves that the model can predict data accurately.

1. 5 How the paper is organised

The rest of this paper is organised as follows. Section 2 discusses project aims, objectives and deliverables, stakeholder analysis, and success criteria. Section 3 describes the detail of our research methodology while section 4 introduces the experiment plan and result before moving to the conclusion in the next section.

2. SCOPING THE PROJECT

There are various studies that investigate customer ratings on mobile applications. For example, Sadiq et al. (2021) investigated the discrepancy between user star rating and text review by proposing a quantitative and qualitative rating contradiction prediction framework using a deep learning model. Abdur Rahman et al., (2020) use a classification algorithm model on Android App review to discover factors for effective version release. The result showed that app users and their reviews pose a crucial part in the software development process. Another study by Umer et al. (2020) built a machine learning model to predict numeric app ratings based on text reviews to increase the accuracy of user star ratings. The finding of this study suggests that user numeric rating is inconsistent and can be biased by text reviews.

2.1 The aim of the project

While most studies focus on elaborating on the difference between numeric and text reviews, the aim of this paper is to assess the feasibility of predicting user ratings for android apps in the Google Play Store based on different app features to improve the quality and standards of app development. We also aim at finding the relationship between various factors influencing user star rating by analysing Google Play Store data.

2.2 Objectives and Deliverables

For this, our objectives include providing categorical analysis of current apps on the Google Play Store to distinguish certain unique app features from one category to another, developing predictors for user app rating, as well as predicting the feature prioritization in this case using the Random Forest algorithms. To ensure the predictors are reliable and satisfy our expectations, we would also perform further model testing by utilising it against the test data and evaluating the accuracy of the predictors.

During the development phase, the group made a presentation, and briefly demonstrated our plans for approaching the project to the audience as well as the methodologies we plan to implement. At the end of the project, we intend to deliver a research report that would introduce the aims of the project, descriptions of the objectives that we set to achieve, this report would also provide detailed explanations of the methodologies that were used in the development of this project, after series of discussion and referencing many relevant articles and literature that were used for better developing our project, we will provide an in-depth report of the result generated in the process. Finally, deliver our conclusion based on the results of our predictions.

2.3 Stakeholder Analysis

Establishing the stakeholders is an important part of the project, before the commencement of setting the aims and objectives, we first identified the person or groups who can influence our project or would be directly affected by it, since they are involved in many aspects of the project. For each stakeholder, we identified some of their characteristics such as the stake of their involvement, the level of priority, the motivators for their interests, what contribution they can provide to the development of the project, and what information they needed as feedback and the communication strategy. For example, product developers are one of the project stakeholders, as they will extensively use the app rating predictor to evaluate and help better improve their own product to achieve a higher app rating, they have a very high interest in our project, however, not as much power to influence the development of the app rating predictor, their project contribution is assisting during the product understanding process, by providing suggestions and expectation on the outcome of the project, the information they need is the operating procedures. Another example of a stakeholder would be the director of the sales department, their stake is divisional KPI to increase mobile app rating and purchase, unlike product developers, the director of the sales department has low interest in the development of the product, but high power over the decision we made. Their motivation is to increase the sale and market share of the company. They contribute to the project by giving strategic advice on the project. To better promote and sell the product, the director needs to understand and be able to explain the outcomes of the project. Even though the director has high power over the project, the person does not need to be updated every step of the process, could be limited to only the kick-off and final presentations of the project.

2.4 Success Criteria

The completion date was planned for the 16th of May, before the finalization, we need to measure whether the project was successful base on the result of our findings. The success criteria we set for the project were to prove the efficacy of using machine learning methods by portraying key features to improve app ratings and establishing the relationships between key app features such as rating and size, category and size, number of installations and reviews.

3. METHODOLOGY

3.1 Data selection

The selection criteria for dataset for the team needs to have following features:

1. The number of data entries should be large enough to support with both training and testing purposes for machine learning algorithms, which means a dataset that only has a few hundred records will not fit in this case.

2. The number of attributes for the dataset should be in a reasonable amount (roughly 10 to 30). A dataset with hundreds of attributes will bring more work for PCA (principal component analysis) to handle dimensionality reduction, which requires more knowledge for a deeper understanding of machine learning and so far, the team does not have sound knowledge base for this type of work.

3. The dataset should contain some noisy data/duplicates/outliers to seek practice data cleaning methods. If the team chose a dataset that has been well prepared, there will not be enough chances for the team get hands-on experience of data preparation.

With the above discussed criteria at hand, the team targeted one dataset that perfectly fits in. The data was collected from real google play store that carries features a mobile app usually has, which contains 10,000 + records 13 attributes with both numeric/non-numeric data included. The team considered this collection of data as good records to analyse and will implement machine learning algorithms accordingly.

3.2 Data cleaning and transformation

The data cleaning and transformation is essential for training the machine learning model as the data used as input should be standardised and contain no corrupted data entries. In our dataset, the team first filtered on the duplicated values on feature ‘app’, which represents the name of an app in google play store. This removes roughly 1181 duplications.

```
[14] df[df.duplicated(subset='app')]
```

	app	category	rating	reviews	size	installs	type	price	content rating	genres	last updated	current ver	android ver
10730	FP Notebook	MEDICAL	4.5	410	60M	50,000+	Free	0	Everyone	Medical	March 24, 2018	2.1.0.372	4.4 and up
10753	Slickdeals: Coupons & Shopping	SHOPPING	4.5	33599	12M	1,000,000+	Free	0	Everyone	Shopping	July 30, 2018	3.9	4.4 and up
10768	AAFP	MEDICAL	3.8	63	24M	10,000+	Free	0	Everyone	Medical	June 22, 2018	2.3.1	5.0 and up

1181 rows × 13 columns

Figure 1

The team contributed more on the data transformation steps as the dataset originally contains combinations of numbers and texts in various attributes. As a requirement of training machine learning model, it is much needed to transform non-numerical data and align the values of existing numerical data.

Some examples will be listed as below to show the steps that the team has conducted to perform data transformation:

1. The dataset contains an attribute “size”, which reflect the actual disk space that is needed for the installation. After the investigation, there seems to be inconsistency across the values in this column as some of them are measured in megabytes while others are in kilobytes. This will cause inaccurate prediction result and to prevent this to happen, the team has converted all the values to megabytes. Also, some data entries have 'Varies with device' under attribute “size”, which indicates the size can vary platform to platform. As this textual value doesn’t provide useful information, the team has decided to transform this value to NULL (in pandas dataframe this would be “NaN”).

```
[ ] df_clean[~df_clean['size'].str.contains('[k,M,Varies with device]$', regex= True, na=False)].head()
```

app	category	rating	reviews	size	installs	type	price	content rating	genres	last updated	current ver	android ver

Figure 2

```
[ ] df_clean['size'] = df_clean['size'].replace('Varies with device', 'NaN', regex=True)
```

```
[ ] df_clean['size']
```

```
0        19M
1        14M
2       8.7M
3        25M
4       2.8M
        ...
9654     53M
9655    3.6M
9656    9.5M
9657     NaN
9658     19M
Name: size, Length: 9659, dtype: object
```

Figure 3

```
[ ] size =[]

    for i in df_clean['size']:
        if i == 'NaN':
            size.append('NaN')
        elif i[-1] == 'k':
            size.append(float(i[:-1])/1000)
        else:
            size.append(float(i[:-1]))
```

```
[ ] df_clean['size'] = size
    df_clean['size'] = df_clean['size'].astype(float)
    df_clean.rename(columns={df_clean.columns[4]:'size(MB)'}, inplace=True)
```

```
[ ] df_clean.head()
```

	app	category	rating	reviews	size(MB)	installs	type	price	content rating	genres	last updated	current ver	android ver
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.1	159	19.0	10,000+	Free	0	Everyone	Art & Design	January 7, 2018	1.0.0	4.0.3 and up
1	Coloring book moana	ART_AND_DESIGN	3.9	967	14.0	500,000+	Free	0	Everyone	Art & Design;Pretend Play	January 15, 2018	2.0.0	4.0.3 and up

Figure 4

2. The dataset has an attribute called “installs”, which stands for the numbers of installations tracked by google play store for an app. This attribute can be used to reflect the popularity of an app so will be essential to be used in the prediction. The team investigated further and found most of the values have “+” tailing the actual number (e.g., 500+, 1000+). This plus sign makes the data type to be string instead of number. So as part of the data transformation, the team removed all the occurrences of this sign and renamed the column as “Installs(+)”.

3.3 Feature Selection

The team went through each feature in the dataset and admit that each one of them stands for a specific dimension that somewhat determines the popularity and the rating of an app. Below is the screenshot of all the features that will be used for machine learning modelling.

```
1 df.columns

Index(['app', 'category', 'rating', 'reviews', 'size', 'installs', 'type',
       'price', 'content rating', 'genres', 'last updated', 'current ver',
       'android ver'],
      dtype='object')
```

Figure 5

3.4 Model Selection

Before fitting the training data with the selected algorithm, the team first raised several questions regarding the correlations between the app rating and other major features. This will give the reader a better idea of how the app rating can be affected in a way that is easier to understand. Some of the questions are listed as below:

1. Do expensive apps usually get higher rating?
2. Do apps with high rating have more reviews?
3. Which category can be found with more reviews?
4. Which category usually gets higher rating?
5. What’s the relationship between the category and app size?
6. What’s the relationship between app rating and size?
7. What’s the relationship between the number of Installs and Reviews?

As for the machine learning algorithms that were selected, the team decided to choose KNN (K nearest neighbours), and random forest. The machine learning model were all imported from sklearn and fed with the dataset.

3.5 KNN (K nearest neighbours)

KNN is a supervised algorithm which is easy to implement and often favoured by beginners. "It can be used for both regression and classification purposes." (“Introduction to Support Vector Machine(SVM) | Dimensionless Academy”) One of the assumptions that KNN has is the similar things will be closely distributed, which means the data points will be near to each other if most values of the features are within the same range.

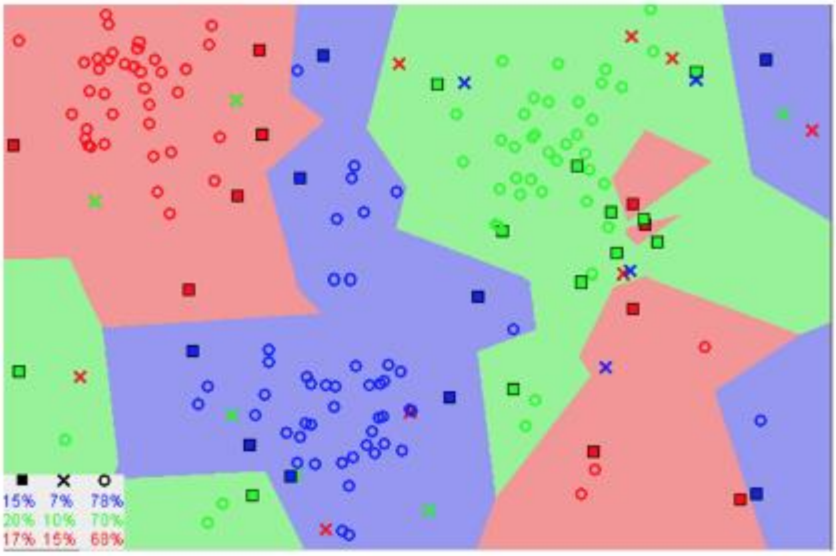

Figure 6 Image showing how similar data points typically exist close to each other

The workflow of how KNN works can be simplified as several steps:

1. Calculate the distance between the current data points and all the other data points.
2. Sort the based on the distance in an ascending order.
3. Select the first **K** nearest data points (this K value needs to be set properly by the developer).
4. Calculate the frequency of occurrence of the category of those K points.
5. The category with the highest frequency among the K points will be used as the predicted category.

3.6 Random Forest

Random forest can be simplified as a superset of a large crowd of decision trees, in which each of the decision trees will have several selected features, and the result of each will be collected and used to determine the answer to the classification problem. The meaning of randomness in this case can be described as 2 key points: First, the randomness during the selection of the dataset. Second, the randomness during the feature selection process that each decision tree performs.

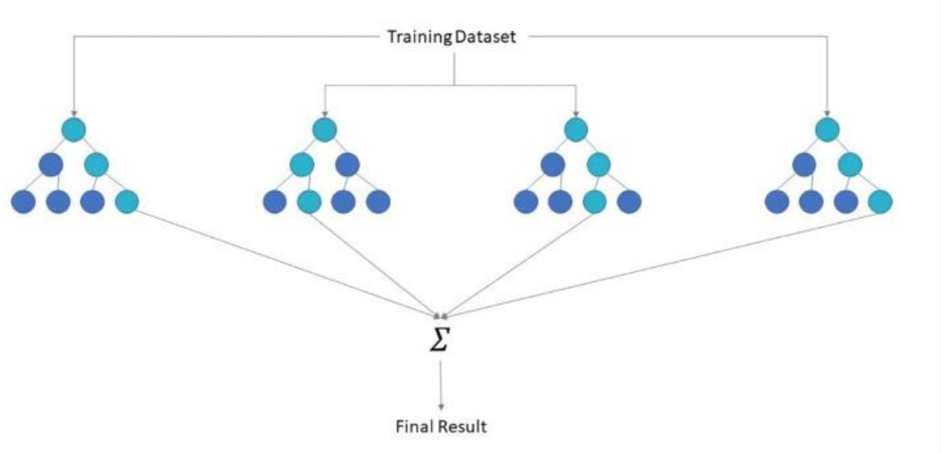

Figure 7:Diagram of Random Forest Classifier

Random forest has several advantages that should be considered:

1. It can provide high accuracy when it comes to prediction problem.
2. It can perform well when being applied on large dataset.
3. It can handle thousands of variables without dimension deduction.
4. It can generate estimates in terms of which features are more important to the classification.
5. It can still get good result when being used on dataset that either has missing data or has considerable proportion of missing data.

Some of the parameters that one should take care of when training with random forest:

1. Number of estimators: this value defines the numbers of trees the model would have. Higher number usually results in better accuracy.

2. Maximum number of features: this value sets the number of features a tree could have.

4. EXPERIMENTS

To predict the app rating, a series of experiments will be conducted. The implementation of the experiment can be described as follows. Firstly, the relationship between the app rating and other attributes will be analysed. The patterns of the analysis include heat maps, bar charts, boxplot, etc. Next, the random forest regressor will be introduced and applied to the machine learning model to predict the rating. Finally, the previous analysis and the machine learning model will be assessed by the root mean square error (RMSE).

4.1 The Analysis of Rating and Attributes

Through the analysis of attributes, we can get a preliminary understanding of their impact on the rating. This is helpful for later predictions.

4.2 Price vs. Rating

According to the graph (graph 4.1), free apps have a wide range of ratings, and apps that cost more than $10 usually have higher ratings.

```
sns.regplot(x='price', y='rating', data=df_clean, marker='+')
plt.title('Price - Rating')
plt.show()
```

Figure 8

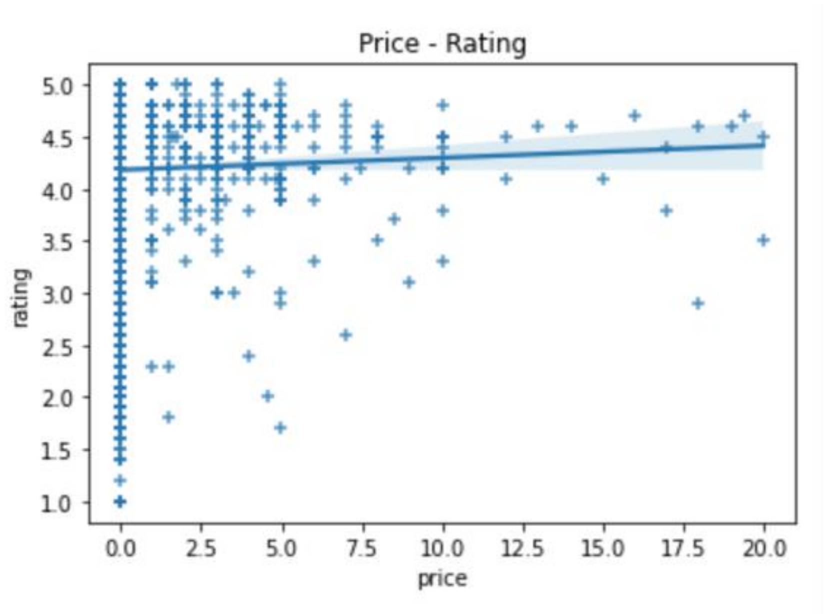

Figure 9: Price Vs Rating

4.3 Number of Reviews vs. Rating

It can be seen from the scatterplot (**Figure 11: Reviews (log) vs Rating**) that apps with more reviews tends to have higher rating. Apps with zero reviews have a wide range of ratings.

```
sns.regplot(y='rating', x='reviews', data=df_clean, marker='+')
plt.title('No. Reviews - Rating')
plt.show()
```

Figure 10

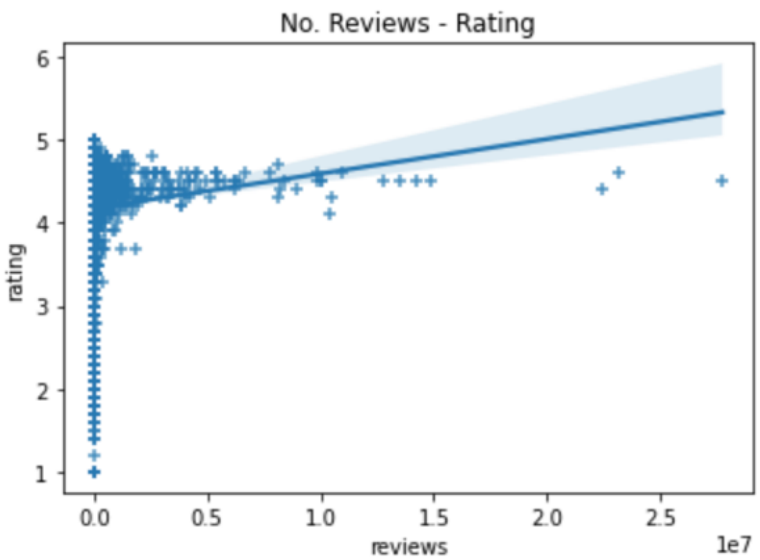

Figure 11: Reviews (log) vs Rating

4.4 Category, Number of Reviews, and Rating

The following boxplot (**Figure 13: Number of Reviews (log) vs Category**) shows all categories and their number of reviews (log). Taking the median number of reviews (log) as a reference, it can be seen that the category of game and entertainment are the categories with more reviews, followed by education and photograph. Besides, free apps are more likely to receive a review than paid apps.

```
plt.figure(figsize=(16,4))

sns.boxplot(x='category', y='reviews', data=df_clean, hue='type',palette="Set3")
plt.yscale('log')
plt.ylabel('')
plt.xticks(rotation=90);
plt.title('No. Reviews(log) - Category')
plt.show()
```

Figure 12

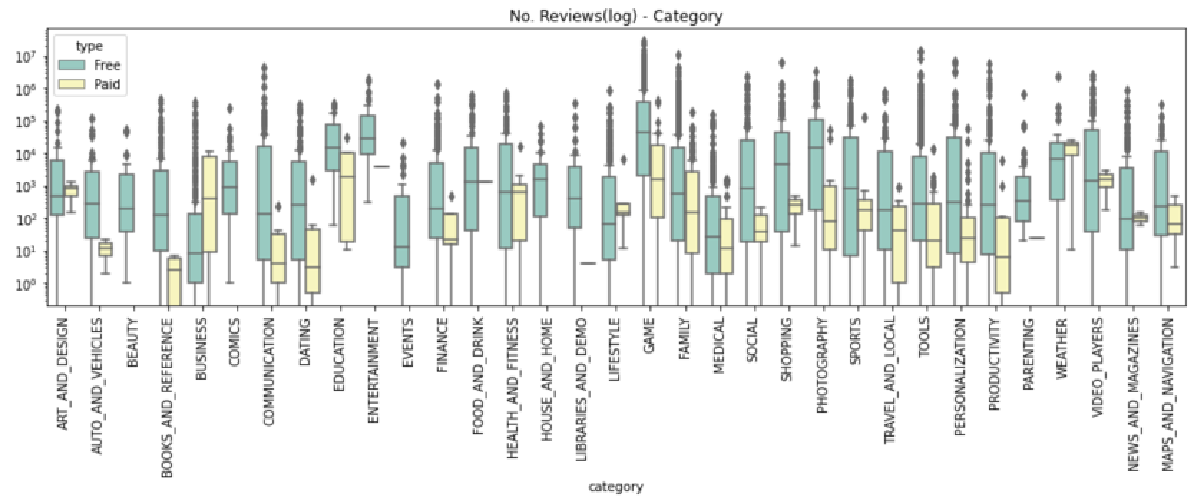

Figure 13: Number of Reviews (log) vs Category

The following boxplot (**Figure 15: Rating vs Category**) reflects the relationship between all categories and their ratings. For most categories, the rating of paid apps is higher than free apps. Moreover, free apps are likely to have more outliers.

```python
plt.figure(figsize=(16,4))

sns.boxplot(x='category', y='rating', data=df_clean, hue='type',palette="Set2")
plt.xticks(rotation=90);
plt.title('Rating - Category')
plt.ylabel('')
plt.show()
```

Figure 14

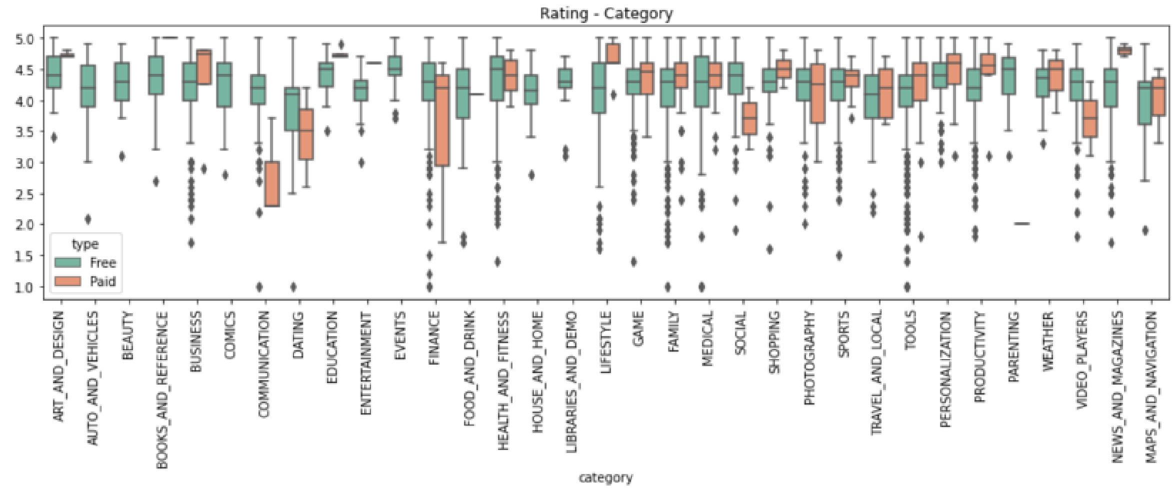

Figure 15: Rating vs Category

4.5 Rating vs. Size

This scatterplot (**Figure 17: Size vs Rating**) indicates that most of the lower-rated (rating < 3.0) apps are smaller than 40MB, and the high rating apps have a wide range of sizes.

```python
sns.regplot(x='rating', y='size(MB)', data=df_clean, marker='+')
plt.title('Size - Rating')
plt.show()
```

Figure 16: Rating vs Category

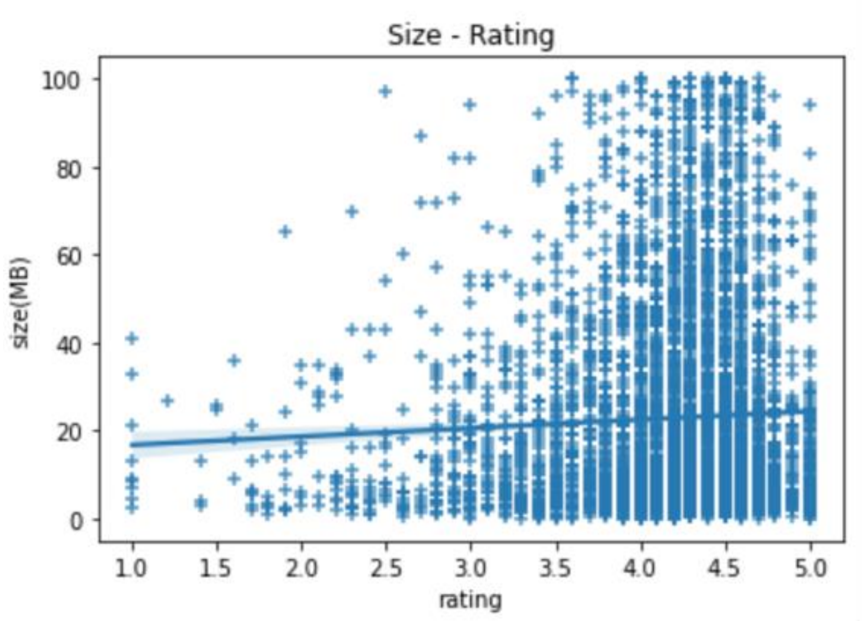

Figure 17: Size vs Rating

4.6 Applying a Machine Learning Model to Predict the Rating

After roughly understanding the relationship between scores and attributes, building a machine learning model for them can make the prediction of rating more accurate. In this part, a heat map for all attributes will be plotted and analysed, and the machine learning model will be introduced in detail.

4.7 Correlation Heat Map of Variables

A heatmap is a statistical chart that displays data by colouring blocks of colour. When drawing, you need to specify the rules for colour mapping (Tuzhidian, 2019). For instance, larger values are represented by darker colours, and small values are represented by lighter colours. In general, heatmaps are useful for viewing populations, spotting outliers, showing differences between multiple variables, and detecting any correlations between them.

Here, the heat map (**Figure 19: Correlation Heat Map**) is used to reflect the correlations between the 9 attributes that would affect the rating. According to the heat map, the number of installs and reviews have a strong connection. It is reasonable because usually, users only give leave a review after they install and use the app. Besides, it can be observed that the correlation index for ‘android vers_main’ and ‘android vers’ is 0.96 which means they are highly similar to each other. Therefore, ‘android vers_main’ will be removed from the later analysis.

```python
plt.figure(figsize=(8,5))
sns.heatmap(df_clean.corr(), cmap='Reds', annot=True)
plt.title('Correlation Heat Map')
```

Figure 18

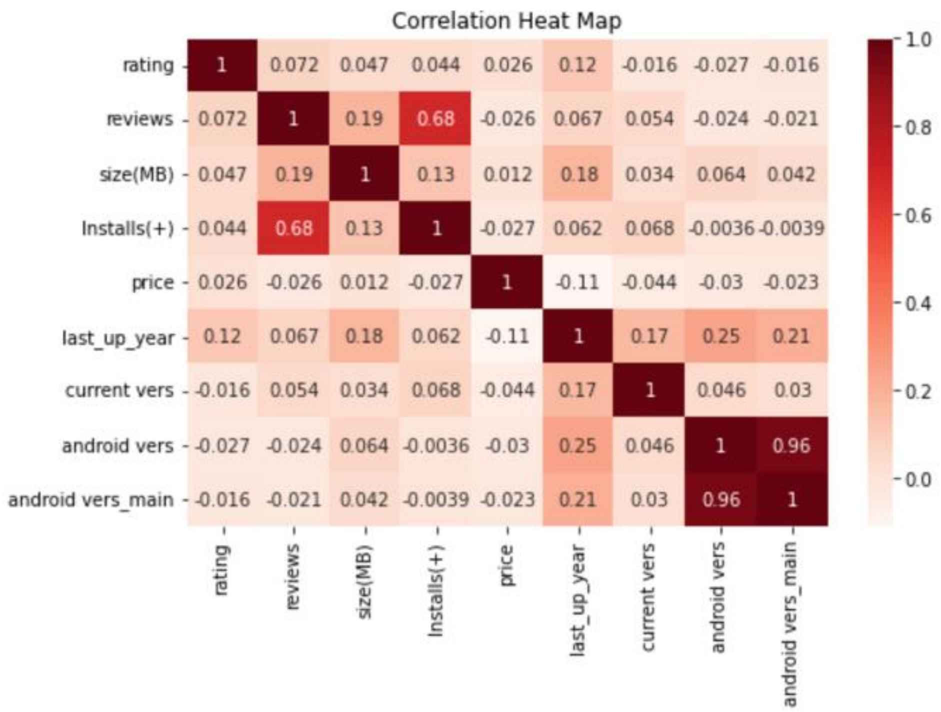

Figure 19: Correlation Heat Map

4.8 Machine Learning Model

In machine learning, a random forest is a classifier that consists of multiple decision trees, and its output category is determined by the mode of the categories output by the individual trees (Yiu, 2019). (“Read Random Forest-Random Forest (4 implementation steps + 10 ...”) Random forests are capable to classify categorical data. It will be applied to the following machine learning model.

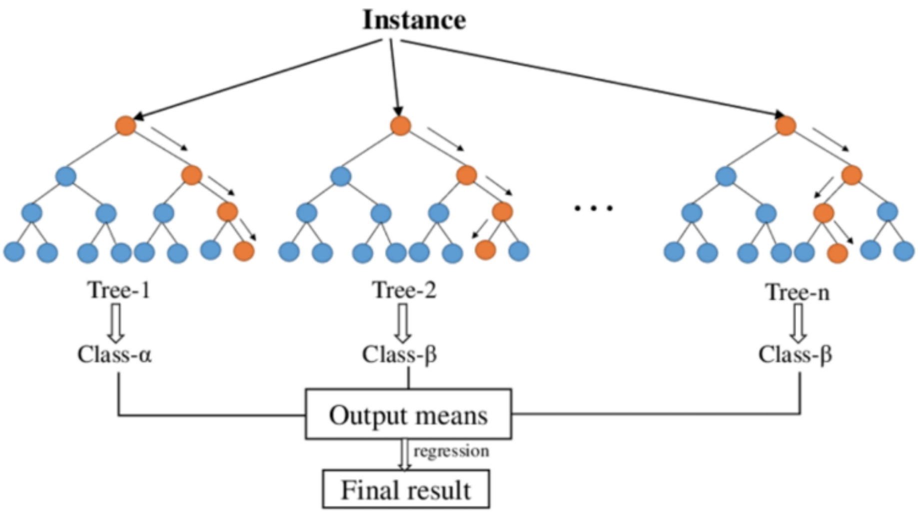

Figure 20: *The principle of random forest*

In the dataset, after variables are transformed into the ideal formats, there are 11 variables (X) and the output (Y), and there are shown in the following table (**Figure 22: Variables and Output**).

```
X=df1.drop('rating', axis = 1).values
y=df1['rating'].values
```

Figure 21

	category	reviews	size(MB)	Installs(+)	type	price	content rating	genres	last_up_year	current vers	android vers	rating
0	0.0	159.0	19.0	9.210440	0.0	0.0	1.0	9.0	2018.0	1.0	4.0	4.1
1	0.0	967.0	14.0	13.122365	0.0	0.0	1.0	12.0	2018.0	2.0	4.0	3.9
2	0.0	87510.0	8.7	15.424949	0.0	0.0	1.0	9.0	2018.0	1.0	4.0	4.7
3	0.0	967.0	2.8	11.512935	0.0	0.0	1.0	11.0	2018.0	1.0	4.4	4.3
4	0.0	178.0	19.0	10.819798	0.0	0.0	1.0	9.0	2018.0	1.0	4.0	3.8

Figure 22: Variables and Output

To conduct the machine learning model, it is necessary to randomly pre-split the data into a training dataset and a testing dataset. The training dataset occupies 80% of the total data, and the testing dataset occupies the rest. Here, the train_test_split function is used to complete this task.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size =
0.2, random_state = 42)
```

Figure 23: Splitting the dataset

Next, it is important to improve the accuracy of the machine learning classifier. It can be achieved by scaling different variables to a given range to make them more numerically comparable (Smilecoc, 2020). The MinMaxScaler( ) function is able to convert the numeric data to the interval [0,1].

```
scaler = MinMaxScaler()
scaler.fit(X_train)
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

Figure 24:Normalizing data

Everything is ready, now the classifier can be trained. Moreover, importing the variables in the testing dataset into the classifier can get the predicted output.

```
rf = RandomForestRegressor()
rf.fit(X_train,y_train)
y_pred_rf = rf.predict(X_test)
```

Figure 25:Training the classifier and predicting output

The following graph (**Figure 27 Variables Importance**) indicates the degree of impact of variables on the rating. ‘android vers’ and ‘current vers’ have a greater impact on the rating. The price, the number of reviews, and the category of the app did not have a huge impact on the rating.

```
feature_name_list=df1.drop('rating', axis = 1).columns
rf.feature_names = feature_name_list

sorted_idx = rf.feature_importances_.argsort()
plt.barh(rf.feature_names,rf.feature_importances_[sorted_idx],alpha=0.5, color ='grey')
plt.xticks(rotation=90);
plt.title('Variables Importance')
plt.xlabel('Feature Importance (%)')
```

Figure 26

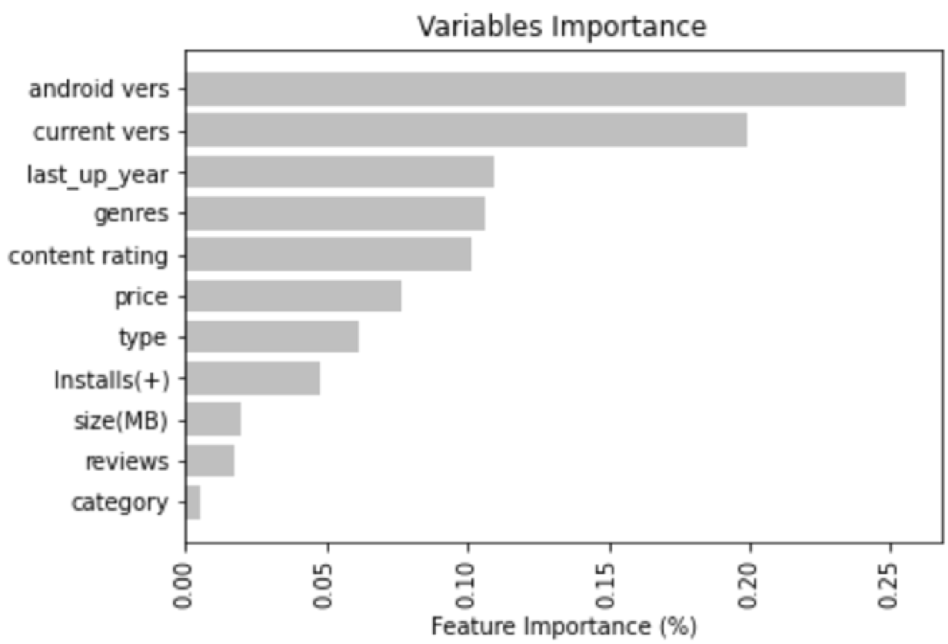

Figure 27 Variables Importance

4.9 Assessment of the Machine Learning Model

In this experiment, the root mean square error (RMSE), as an index, is going to assess the machine learning classifier. It is of the same order of magnitude as the real value. So, the predicted value can thus be evaluated more straightforward. The predicted rating (y_pred_rd) that is generated from the machine learning classifier and the true rating would be involved in the calculation. Here, the root mean square error value is 0.47, then it can be considered that the regression prediction is 0.47 different from the true value on average.

```
print(r"RMSE with RF: {:.3f}".format(np.sqrt(mse_rf)))
```

```
RMSE with RF: 0.477
```

Figure 28

The smaller the RMSE, the better. According to the rule of thumb, it can be said that the RMSE value is between 0.2 and 0.5, indicating that the model can predict the data more accurately (Saeedi, 2020). And the above RMSE value is 0.47 represents that this machine learning model is acceptable. As RMSE is of the same order of magnitude as the real value, this result can be described as that the predicted score is 0.47 different to the real score on average.

5. DISCUSSION AND CONCLUSIONS

5.1 Summary of Research Study

Our group set out to determine the features that help improve an app's ratings. The study focuses on using categorical analysis to determine causal relationships between variables or features in this instance and Random Forest to ultimately classify the variables based on its importance to app ratings. Root Mean Square Error (RMSE) is then used as an index for the model to help determine if the model predicts the data accurately. The use of RMSE instead of R2 was preferred because the former tells the distance of predicted values on average. The lower the value of RMSE, the better fit the model. ("Implementing Multiple Linear Regression - EDUCBA") Whereas R2 is a metric used in determining the variance of a dependent variable/s (x variable/s) that is explained by the independent variable in a regression model, which as can be seen in this study, was not utilized.

In the categorical portion of this study, price proved to have an effect to ratings as apps priced $10 or more tended to have better ratings. This can be due to an app being paid, hence, users tended to have the push to review the app for other users. The same goes for number of reviews as apps with more reviews tend to be better rated. Categorial analyses on categories and the log number of reviews was taken as well to show which genre or category had more reviews. With games and entertainment spearheading the categories with the most reviews, education and photography apps followed suit. When it comes to ratings however, paid apps proved to have higher ratings as outliers existed more on free apps. Finally, there is a correlation between size and rating as smaller sized apps tended to have lower ratings compared to bigger apps.

In the case of this study, we wanted and got an RMSE below 0.5 as this would entail that the average deviation between the predicted app rating is 0.47 or in simple words, predicted outcomes fit within the model. This degree of fitness can then be translated in the use of Random Forest where results show that the android version shows the most feature importance when it comes to app ratings. This can be interpreted by apps that require the latest version of android to have the smoothest utility. Good app ratings can be easily deterred by bugs or even possible inability to install the app on their phones. This can be further interpreted as applications now whose versions don't sync synonymously with the latest android updates are more likely to encounter issues when it comes to installing and/or running the app. This is probably why our model shows this as the most significant feature in app ratings.

5.2 Limitations and Future Direction / Recommendations for Study

My previous notion that R2 was not used to assess model fitness is a current weakness and can be used in future studies. In determining fit in a dataset, it is standard to calculate both RMSE and R2 because they both describe different things as I previously stated in the summary.

When it comes to determining feature importance, Random Forest can be compared to another Machine Learning algorithm such as XGBoost for future study. XGboost is a simple tool in helping plot features according to their importance. This is a built-in tool in programs like Python and can be used to further transform our dataset into a subset with selected features. This is called the SelectFromModel (Python Package Introduction, n.d.). The thought process is to take an existing trained model on the dataset, which will be used as a threshold for when you choose features from the training and test datasets. From experience of this method, results can have a random distribution and may need to be run multiple times and averaging the outcome to have a decent analysis.

Furthermore, due to time constraints, a less holistic categorical analysis had to be done to compensate for quality of work. The focus of this study heavily relied on Random Forest but could have explained more on categorical analysis. We wrote it this way so that future researchers and basic programmers can practice their basic categorical code to further study each feature and how it correlates to one another.

A more in-depth study can be done on the genre of apps that can possibly affect app ratings. Initial impressions suggest that productivity apps tend to have the highest and lowest ratings in a spectrum just because these apps require the most intricate designs and fluidity when it comes to utility. This a study for further research in the future.

Finally, our results show that android versions deeply affect app ratings. Further study can involve the investigation as to how versions of a phone can affect the stability of an app and if that app's smoothness directly correlates with the version of the phone.

REFERENCE

Australian Government Department of Health. (2021, February 24). *About private health insurance*. https://www.health.gov.au/health-topics/private-health-insurance/about-private-health-insurance

Tuzhidian. (2019, March). *Heatmap.* http://tuzhidian.com/chart?id=5c56e4284a8c5e048189c6fe

Yiu, T. (2019, June 12). *Understanding Random Forest. Towards Data Science.* https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Smilecoc. (2020, September 3). *Comparison of three data standardization methods in sklearn: StandardScaler, MinMaxScaler, RobustScaler.* https://smilecoc.vip/2020/09/03/sklearn%E4%B8%89%E7%A7%8D%E6%95%B0%E6%8D%AE%E6%A0%87%E5%87%86%E5%8C%96%E6%96%B9%E6%B3%95%E7%9A%84%E5%AF%B9%E6%AF%94%EF%BC%9AStandardScaler%E3%80%81MinMaxScaler%E3%80%81RobustScaler/

Umer, M., Ashraf, I., Mehmood, A., Ullah, S. and Choi, G., 2020. Predicting numeric ratings for Google apps using text features and ensemble learning. *ETRI Journal*, 43(1), pp.95-108.

Sadiq, S., Umer, M., Ullah, S., Mirjalili, S., Rupapara, V. and Nappi, M., 2021. Discrepancy detection between actual user reviews and numeric ratings of Google App store using deep learning. *Expert Systems with Applications*, 181, p.115111.

Mahmood, A., 2019. Identifying the influence of various factor of apps on google play apps ratings. *Journal of Data, Information and Management*, 2(1), pp.15-23.

Xiao, L. and Li, Y., 2019. Examining the Effect of Positive Online Reviews on Consumers' Decision Making. *Journal of Global Information Management*, 27(3), pp.159-181.

Statista. (2018). *App stores: number of apps in leading app stores 2018* | *Statista*. Statista; Statista. https://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/

Python Package Introduction — xgboost 1.5.2 documentation. (n.d.). Xgboost.readthedocs.io. https://xgboost.readthedocs.io/en/stable/python/python_intro.html

Saeedi, M. (2020). What's the acceptable value of Root Mean Square Error (RMSE), Sum of Squares due to error (SSE) and Adjusted R-square? *ResearchGate.* https://www.researchgate.net/post/Whats-the-acceptable-value-of-Root-Mean-Square-Error-RMSE-Sum-of-Squares-due-to-error-SSE-and-Adjusted-R-square/5e9b97b5c6a89b649d5223cf/citation/download.

Ibm cloud education. (2020, December 7). Random Forest. https://www.ibm.com/cloud/learn/random-forest

APPENDIX

Google Play Store Dataset link:

https://www.kaggle.com/datasets/gazishaikat/googleplaystore

Google Colaboratory link:

https://colab.research.google.com/drive/18BiMngiX5xH8eltBdPtIQr6h3_Lm6Jh7?usp=sharing