CS 210 – Introduction to Data Science Final Report

Project Title: Google Play Store Apps

Group Members: Uğurcan Ertuğrul, Anıl Kantoğlu, Rutkay Yıldırım, Canberk Keleş, Tuvan Gezer

Date: 11.12.2019

1. INTRODUCTION

The purpose of this project is to increase knowledge about data science and getting hands-on practical experience. We have assembled a group with 5 group members. 3 Computer Science Students and 2 Industrial Engineering Students. We have chosen the project that contains Google Play Store applications dataset from the given datasets. In this dataset we have 13 columns,and rows with applications name.These columns contains information about applications which are on Google Play Store. Columns are ordered by; application name, category, rating, reviews, size, installs, type, price, content rating, genres, last updated, current version and android version. Our main goal in this project is to develop a certain data analysis about an application we want to release, with an in depth analysis of Google Play Store dataset we want to estimate the price and number of install numbers in contrast with other information that contains in this dataset. We will conduct a hypothesis test and get insights from data and answer them using the tools and techniques that we learned in this class. Asking related questions about this dataset is the key factor to understand the problem we want to solve in this project.

2. PROBLEM & PURPOSE

2.1 Problem

Mobile application industry is a multi billion dollar industry. Most of the applications are developed for profit and development isn't free. Having a good forecast about the return of investment for an application before it's development is crucial. There are many parameters that has a role in an application's success or failure such as its price or target audience. The vast number of parameters make it hard to create a prediction system that can efficiently extract information from the features and come up with a realistic prediction.

2.2 Purpose

Our purpose is to use the Google Play Store dataset and use it with machine learning models to create an accurate model that will predict the success of an application based on its various features so that application developers don't waste their precious time on an idea that is less likely to succeed.

3. OUR DATASET & METHODS

Recently, the usage of Google Play Store has been increased, so does the usage of the applications which are downloaded from Google Play Store. So, what we examine here is a comprehensive dataset of the applications uploaded or found in Google Play Store, we have 2 datasets to examine. The first is General data of the applications, the second one includes the reviews of the related applications in Play Store. In the first dataset, which has 13 columns, lists the names of the applications and there are 9660 unique items, this means there are 9660 applications listed in this data set. At least it has to mean like that but when examined, dataset seems to have more than 10000 rows, so there are more than one application has the same names. Aside from that, Each column gives a specific attribute of the application.

1. App: the name of application. The name of the application is a unique data type so there must be only one found in the entire data set.
2. Category: genre of the application, facilitates to sort similar kinds of applications in dataset.
3. Rating: The mean of the ratings of the application made by users in the dataset, so that the user can have a pre opinion about the application whether it is useful and satisfies the expectations of the user.
4. Reviews: This shows how many reviews are made for the applications by the users.
5. Size: Shows how much storage that an application will occupy in memory. M is for Megabyte (MB), G is for Gigabyte(GB), K is for Kilobyte(KB) for the possible cases.
6. Installs: This column shows how many times the application has been downloaded by users. The data in this columns are not the exact values, there is 5,000,000+ also 10,000+. The download amount is categorized within numbers to make it more practical.
7. Type: This shows whether the application is free or paid. For our case we don't need this column explained in the next column.
8. Price: Shows the price of the applications. This column beats the purpose of the type column. Because if the price is 0, this means free if price is higher than 0, this means paid. So there is no need for Type column actually.
9. Content Rating: This is a column that shows aim of the application's age category. It can be for everyone, adult and mature only, and x+ age only.
10. Genres: This column functions similar as the Category column. This seems unnecessary at first but the difference between the category is, in category only the most general 'One' term can be entered. In genres, more than one tag can be included; for example, there is Art & Design; Creativity. So genre becomes useful for this occasions.
11. Last Updated: This column shows the date of when the up to date version released for the application.

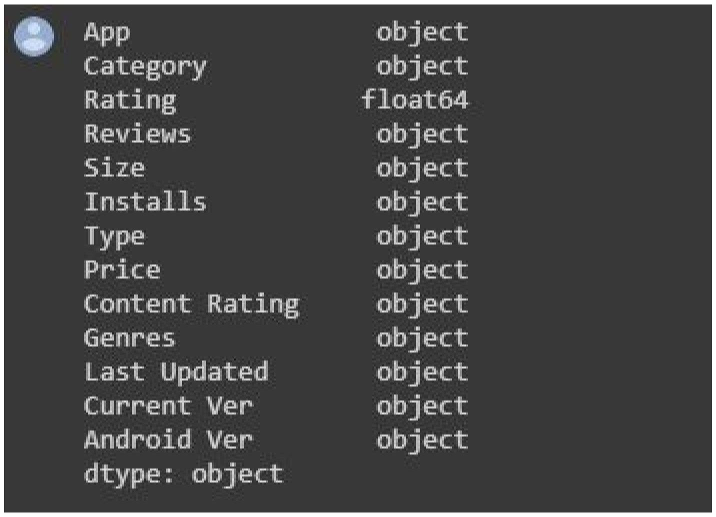
12. Current Ver: This explains / shows the up to date version of the application. However, there is no data about the release date of the application.
13. Android Ver: Shows the required software version of the applications. It says x and up. Means below x versions cannot run the application(s).

In this dataset there are flawed data, this dataset is not a perfect dataset. For example, there is an entry of a decimal in category section. ‘1.9’ in category section. Also, there is a case that there are NaN values in some sections. In an app, the rating sections says NaN but the app has 3 reviews. In addition, In Size section, the application can vary with devices. So there is no exact value of size of the application this is a bad data type, obstacles the estimation among other data. Overall, the dataset has dirty values and missing values (NaN). So this dataset needs data cleaning.

METHODS THAT ARE USED

- **READING THE DATASET**

First, we need to get an image about the dataset we are provided (Even though the explanation is provided from the source where the data set is obtained, we should double check the dataset.).



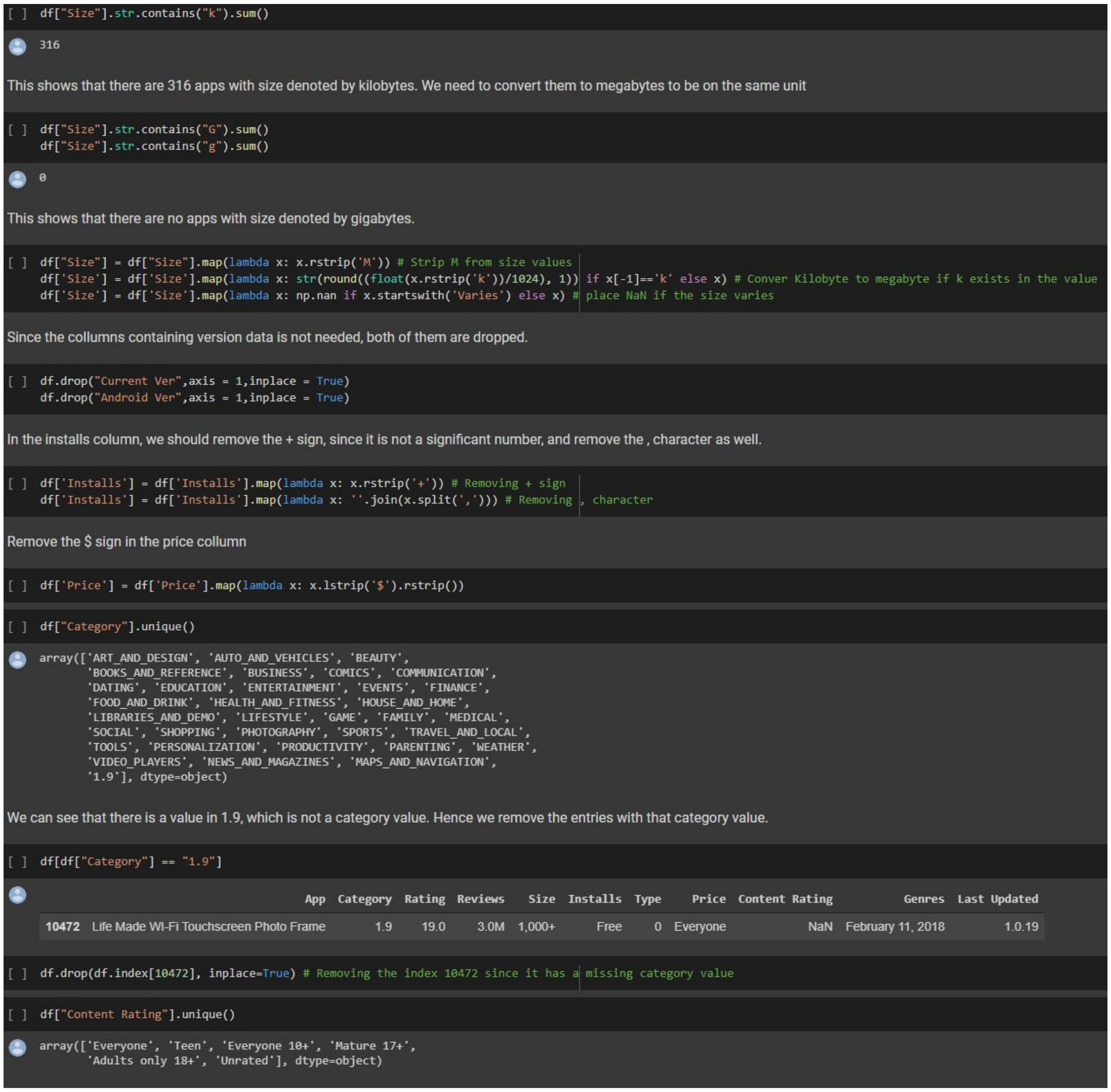
```
App                object
Category           object
Rating            float64
Reviews            object
Size               object
Installs           object
Type               object
Price              object
Content Rating     object
Genres             object
Last Updated       object
Current Ver        object
Android Ver        object
dtype: object
```

As can be seen, almost all of our data is in object dtype, it is difficult to make predictions with object types since we don’t know what object is. With that and the possibility of NaN values, we need to clean the dataset to our needs.

- **CLEANING THE DATASET**

Data comes with unexpected values, or values such as that with cleaning the value the performance of trained models will be better, and NaN values should be taken care of as well since we can not make any assumption on NaN values. First thing to note is that every data in

"Size" column have an additional "M" after the size to denote the size in megabytes. This is not necessary if all of them have M, and if not we can convert the values to megabyte values to make more effective predictions.



```
[ ] df["Size"].str.contains("k").sum()
```

```
316
```

This shows that there are 316 apps with size denoted by kilobytes. We need to convert them to megabytes to be on the same unit

```
[ ] df["Size"].str.contains("G").sum()
    df["Size"].str.contains("g").sum()
```

```
0
```

This shows that there are no apps with size denoted by gigabytes.

```
[ ] df["Size"] = df["Size"].map(lambda x: x.rstrip('M')) # Strip M from size values
    df['Size'] = df['Size'].map(lambda x: str(round((float(x.rstrip('k'))/1024), 1)) if x[-1]=='k' else x) # Conver Kilobyte to megabyte if k exists in the value
    df['Size'] = df['Size'].map(lambda x: np.nan if x.startswith('Varies') else x) # place NaN if the size varies
```

Since the collumns containing version data is not needed, both of them are dropped.

```
[ ] df.drop("Current Ver",axis = 1,inplace = True)
    df.drop("Android Ver",axis = 1,inplace = True)
```

In the installs column, we should remove the + sign, since it is not a significant number, and remove the , character as well.

```
[ ] df['Installs'] = df['Installs'].map(lambda x: x.rstrip('+')) # Removing + sign
    df['Installs'] = df['Installs'].map(lambda x: ''.join(x.split(','))) # Removing , character
```

Remove the $ sign in the price collumn

```
[ ] df['Price'] = df['Price'].map(lambda x: x.lstrip('$').rstrip())
```

```
[ ] df["Category"].unique()
```

```
array(['ART_AND_DESIGN', 'AUTO_AND_VEHICLES', 'BEAUTY',
       'BOOKS_AND_REFERENCE', 'BUSINESS', 'COMICS', 'COMMUNICATION',
       'DATING', 'EDUCATION', 'ENTERTAINMENT', 'EVENTS', 'FINANCE',
       'FOOD_AND_DRINK', 'HEALTH_AND_FITNESS', 'HOUSE_AND_HOME',
       'LIBRARIES_AND_DEMO', 'LIFESTYLE', 'GAME', 'FAMILY', 'MEDICAL',
       'SOCIAL', 'SHOPPING', 'PHOTOGRAPHY', 'SPORTS', 'TRAVEL_AND_LOCAL',
       'TOOLS', 'PERSONALIZATION', 'PRODUCTIVITY', 'PARENTING', 'WEATHER',
       'VIDEO_PLAYERS', 'NEWS_AND_MAGAZINES', 'MAPS_AND_NAVIGATION',
       '1.9'], dtype=object)
```

We can see that there is a value in 1.9, which is not a category value. Hence we remove the entries with that category value.

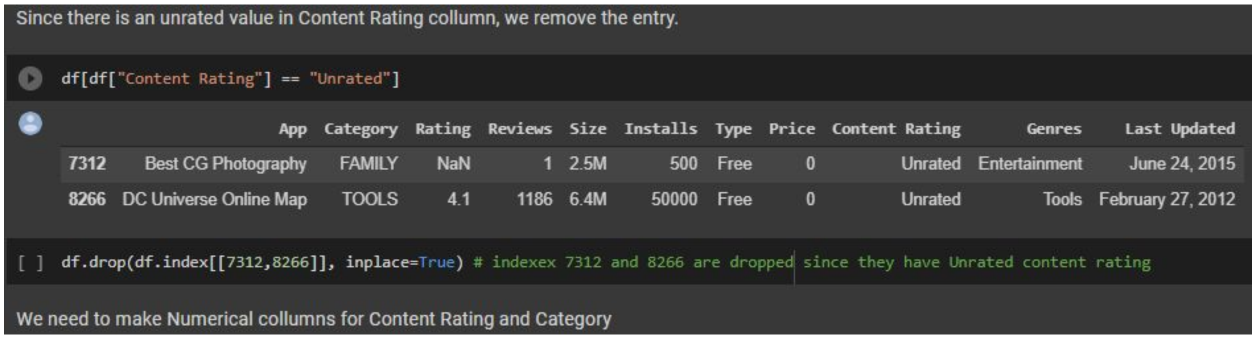
```
[ ] df[df["Category"] == "1.9"]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated
10472	Life Made WI-Fi Touchscreen Photo Frame	1.9	19.0	3.0M	1,000+	Free	0	Everyone	NaN	February 11, 2018	1.0.19

```
[ ] df.drop(df.index[10472], inplace=True) # Removing the index 10472 since it has a missing category value
```

```
[ ] df["Content Rating"].unique()
```

```
array(['Everyone', 'Teen', 'Everyone 10+', 'Mature 17+',
       'Adults only 18+', 'Unrated'], dtype=object)
```


Since there is an unrated value in Content Rating collumn, we remove the entry.

```
df[df["Content Rating"] == "Unrated"]
```

	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated
7312	Best CG Photography	FAMILY	NaN	1	2.5M	500	Free	0	Unrated	Entertainment	June 24, 2015
8266	DC Universe Online Map	TOOLS	4.1	1186	6.4M	50000	Free	0	Unrated	Tools	February 27, 2012

```
df.drop(df.index[[7312,8266]], inplace=True) # indexex 7312 and 8266 are dropped since they have Unrated content rating
```

We need to make Numerical collumns for Content Rating and Category

So we put an index to every category alphabetically, after that we managed with the NaN values of dataset. Our method in cleaning NaN values was, filling them with the mean values of it's section. However, this method cannot apply on all sections, At that point, we cleared our data as best as possible but there are still NaN values in Size, which we can not make any assumption on, and 1 NaN value in Type column. Therefore, we delete them from our data. Finally, we converted object variables into numeric variables.

- **DATA VISUALIZATION**

For this section, we decided to create 2 kinds of graphs, the first one shows mean values of installment count with respect to categories of the applications. Since this graph will be highly related to our purpose and question because of predicting installment numbers according to category and also the next purpose, we executed the graph with mean values of rating with respect to their categories.

- **TRAIN TEST SPLIT**

From this part, all methods are belong to our ML methods, we took the values of %75/%12.5/%12.5 for split test.

Feature generation

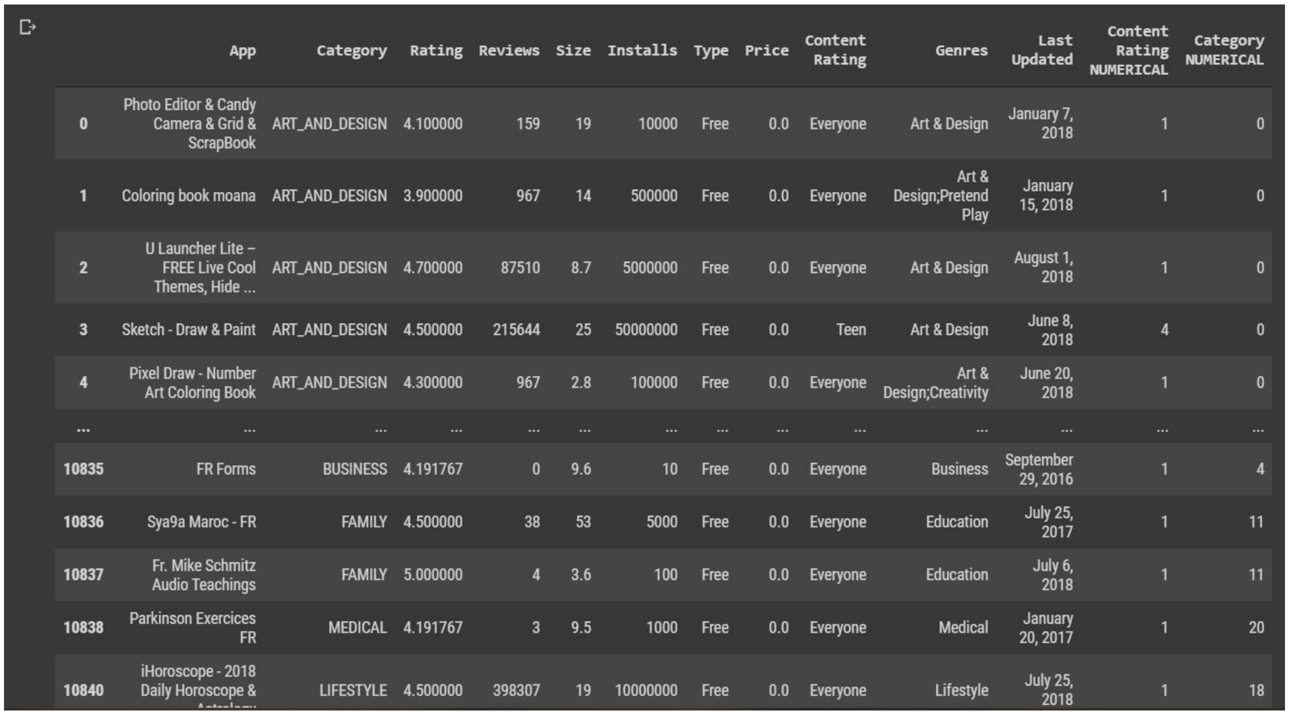
As the purpose of the project suggests, we are going to predict the number of installments before releasing an app to Playstore, and after putting an app and collecting some number of reviews, we will try to predict the optimal price for the app. We are going to use features:

1)Category
2)Rating
3)Reviews
4)Installs
5)Price
6)Size
7)Content Rating

- **MODEL SELECTION**

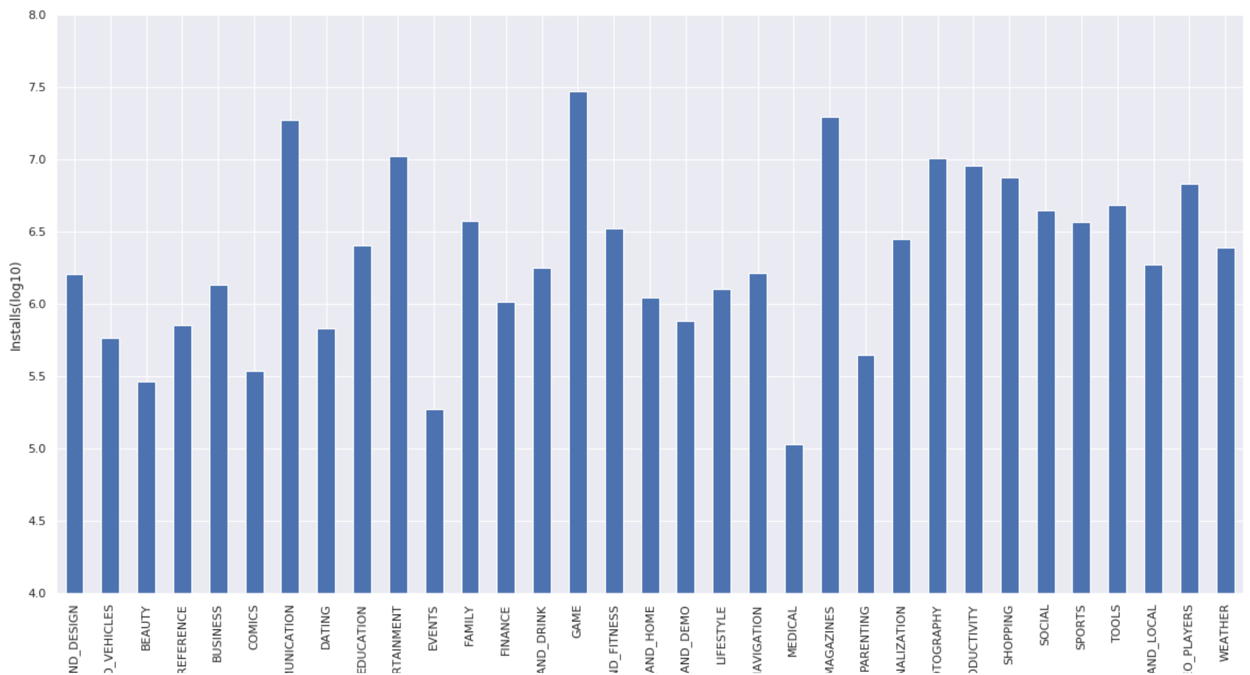
This problem is a regression problem, so we are going to use Random Forest model for the purpose of this problem. After having results of random forest model, firstly, matrix number graph was used with the random forest model value. Secondly, we visualized the accuracy score by graphs, including test error, validation error and also train error values.

4. RESULTS OF OUR TOOLS

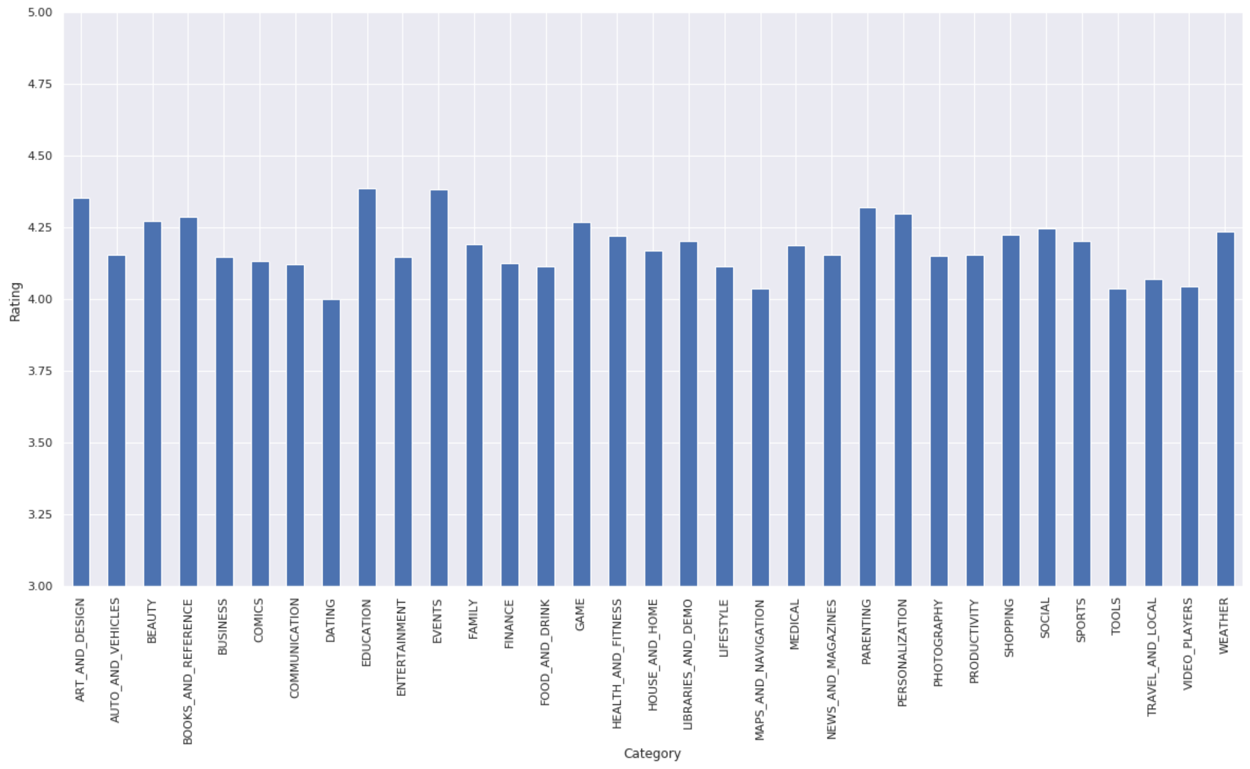


	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Genres	Last Updated	Content Rating NUMERICAL	Category NUMERICAL
0	Photo Editor & Candy Camera & Grid & ScrapBook	ART_AND_DESIGN	4.100000	159	19	10000	Free	0.0	Everyone	Art & Design	January 7, 2018	1	0
1	Coloring book moana	ART_AND_DESIGN	3.900000	967	14	500000	Free	0.0	Everyone	Art & Design;Pretend Play	January 15, 2018	1	0
2	U Launcher Lite – FREE Live Cool Themes, Hide ...	ART_AND_DESIGN	4.700000	87510	8.7	5000000	Free	0.0	Everyone	Art & Design	August 1, 2018	1	0
3	Sketch - Draw & Paint	ART_AND_DESIGN	4.500000	215644	25	50000000	Free	0.0	Teen	Art & Design	June 8, 2018	4	0
4	Pixel Draw - Number Art Coloring Book	ART_AND_DESIGN	4.300000	967	2.8	100000	Free	0.0	Everyone	Art & Design;Creativity	June 20, 2018	1	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
10835	FR Forms	BUSINESS	4.191767	0	9.6	10	Free	0.0	Everyone	Business	September 29, 2016	1	4
10836	Sya9a Maroc - FR	FAMILY	4.500000	38	53	5000	Free	0.0	Everyone	Education	July 25, 2017	1	11
10837	Fr. Mike Schmitz Audio Teachings	FAMILY	5.000000	4	3.6	100	Free	0.0	Everyone	Education	July 6, 2018	1	11
10838	Parkinson Exercices FR	MEDICAL	4.191767	3	9.5	1000	Free	0.0	Everyone	Medical	January 20, 2017	1	20
10840	iHoroscope - 2018 Daily Horoscope &	LIFESTYLE	4.500000	398307	19	10000000	Free	0.0	Everyone	Lifestyle	July 25, 2018	1	18

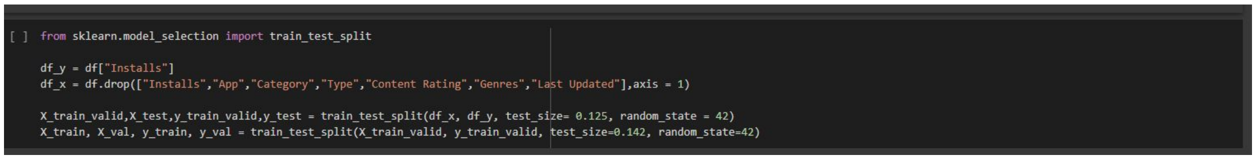
After cleaning the dataset for our purposes, we were able to get a dataset that looks like above figure.



The above figure shows the mean of number of installs relative to each category, with numbers being applied the log10 function. The data shows Game category has the highest mean in number of installs.

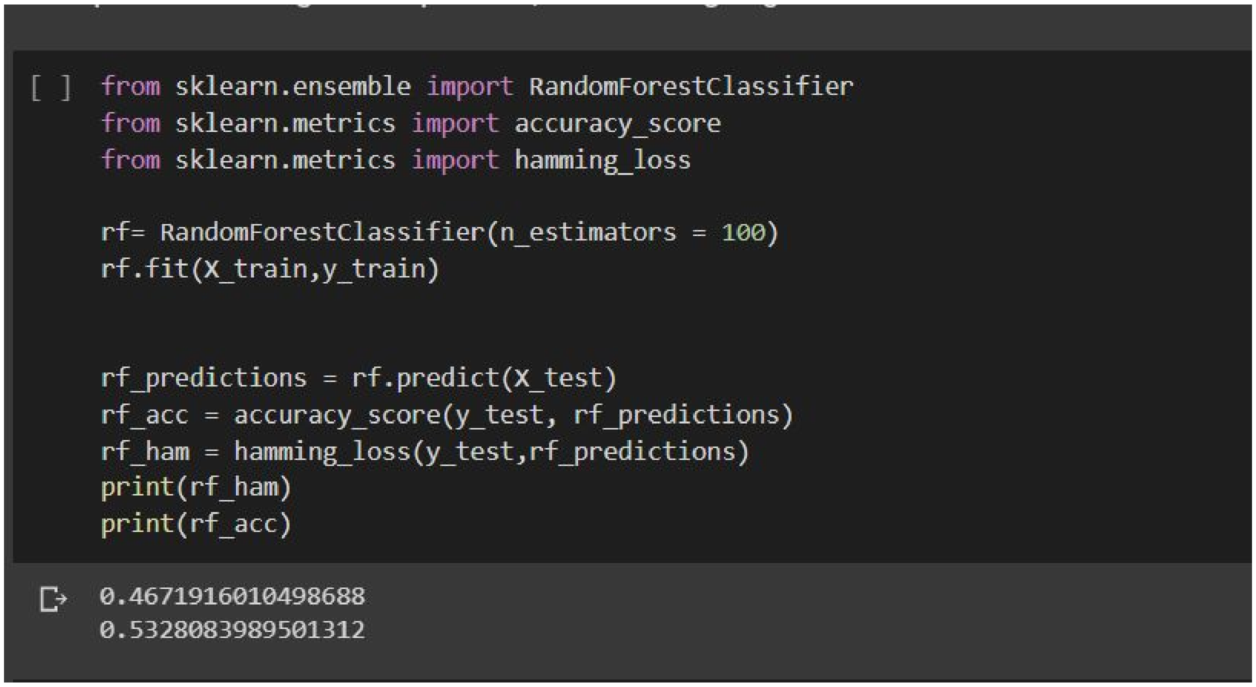


This graph shows the mean of ratings for each category. It can be said that Education and Events category has the highest mean rating. Even though most applications downloaded were from games category, Games category does not have the highest rating.



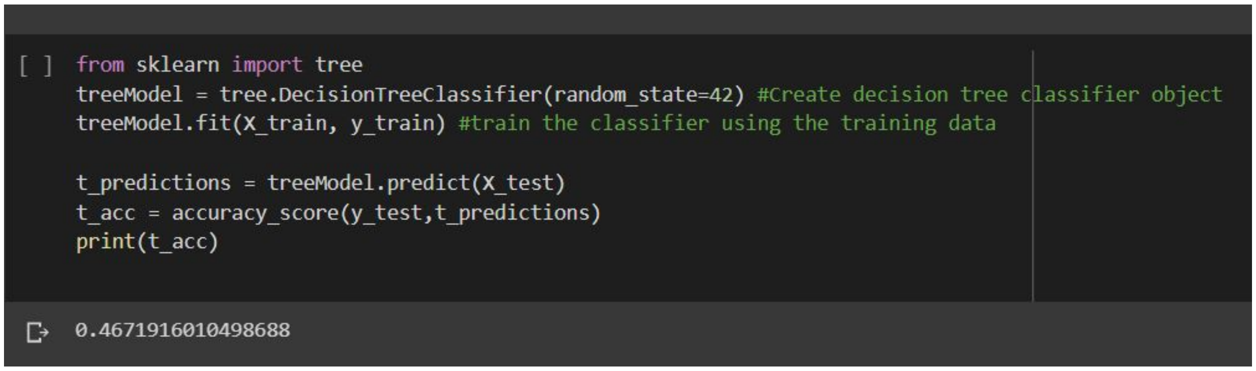
```
[ ] from sklearn.model_selection import train_test_split

    df_y = df["Installs"]
    df_x = df.drop(["Installs","App","Category","Type","Content Rating","Genres","Last Updated"],axis = 1)

    X_train_valid,X_test,y_train_valid,y_test = train_test_split(df_x, df_y, test_size= 0.125, random_state = 42)
    X_train, X_val, y_train, y_val = train_test_split(X_train_valid, y_train_valid, test_size=0.142, random_state=42)
```

Number of installs is a measure of how much an app was successful. Hence the prediction will be the number of installs for a hypothetical app given its Category(Numerical), Size, Content Rating(Numerical), Rating and Price.
We split the data into %75 test, %12.5 train and %12.5 validation test data.



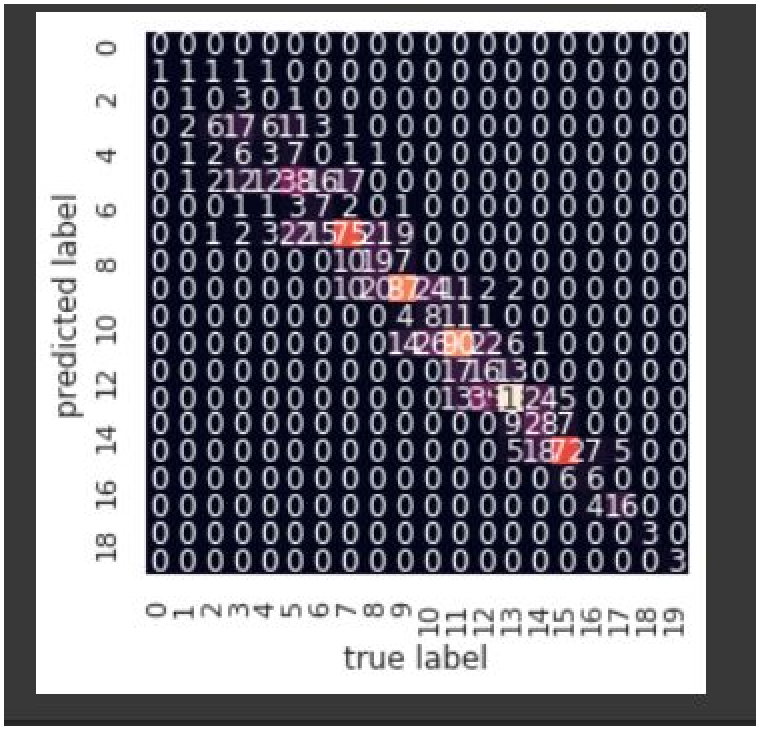
```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.metrics import accuracy_score
    from sklearn.metrics import hamming_loss

    rf= RandomForestClassifier(n_estimators = 100)
    rf.fit(X_train,y_train)


    rf_predictions = rf.predict(X_test)
    rf_acc = accuracy_score(y_test, rf_predictions)
    rf_ham = hamming_loss(y_test,rf_predictions)
    print(rf_ham)
    print(rf_acc)

    0.4671916010498688
    0.5328083989501312
```

We calculated the accuracy of random forest model, and its hamming loss which gives a clue about how much the prediction deviates.

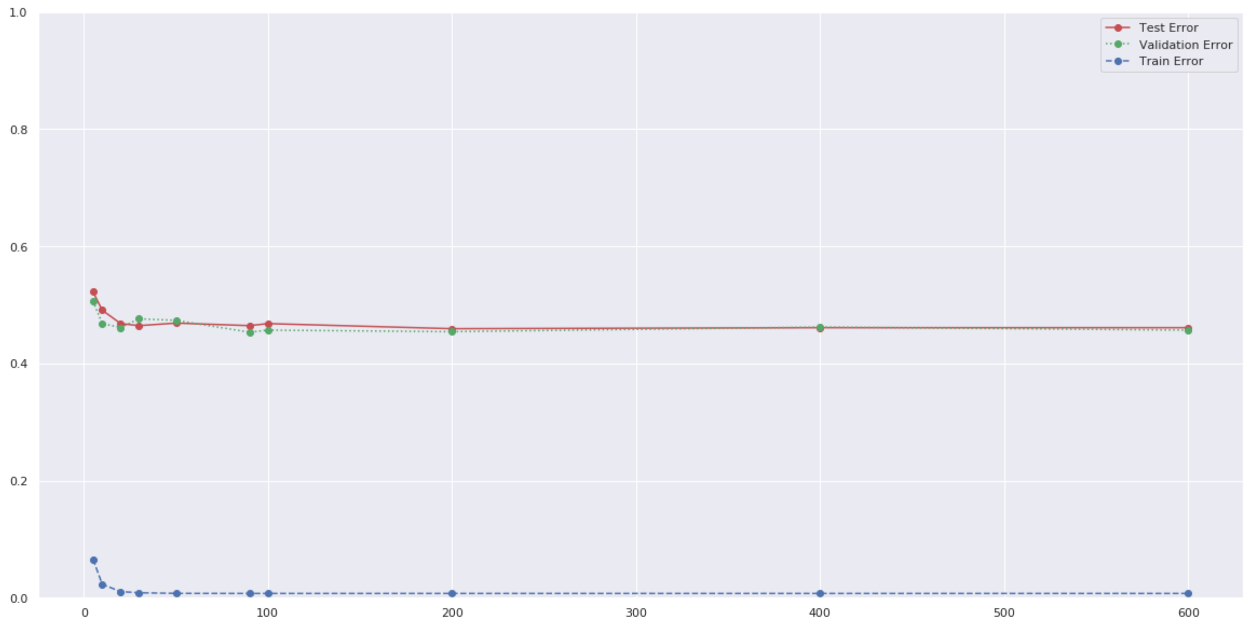


```
[ ] from sklearn import tree
    treeModel = tree.DecisionTreeClassifier(random_state=42) #Create decision tree classifier object
    treeModel.fit(X_train, y_train) #train the classifier using the training data

    t_predictions = treeModel.predict(X_test)
    t_acc = accuracy_score(y_test,t_predictions)
    print(t_acc)

    0.4671916010498688
```

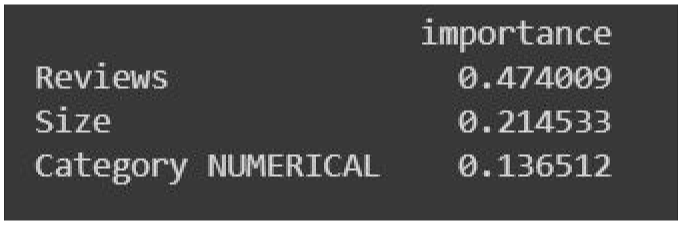
We tried a second model, decision tree, which has a lower accuracy than the random forest model hence the better model was chosen to be random forests.



The confusion matrix is shown above. Confusion matrix shows that the random forest model is working rather successfully. About 50% of the test results are correctly predicted and the rest are predicted closely. This can be seen from the confusion matrix as the upper and lower triangles are completely zero.



The figure above shows the test, train, and validation errors. It is clear that there is some overfitting after the estimator count goes above a certain threshold. The test and validation error settles at around 50% mark. This performance may make our model appear not very good but we believe that the fact that our model predicts close to the true value when it is wrong (shown by the confusion matrix above) and that there are many possible classifications for the random forest to choose from makes our model a successful model.



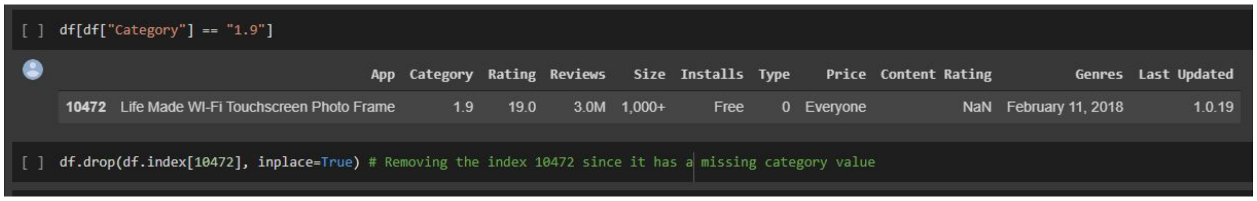
```
                      importance
Reviews                 0.474009
Size                    0.214533
Category NUMERICAL      0.136512
```

We then concluded that the 3 most important aspects for predicting the number of installs is Reviews, Application Size and Category.

5. DISCUSSION OF THE METHODS

In our purpose, we wanted to predict an installment number according to the category,rating,review, price, content rating and size of the application beforehand uploading the application on the google play store platform. One may be able to conduct a market research to obtain a number of reviews value, then predict the number of installs for the app. So we needed a comprehensive dataset. There were good points and drawbacks.

First of all, finding an existing suitable dataset that has a handful of sections was in our advantage, so that we could easily select our modelling method and machine learning algorithm method. However, the dataset had its drawbacks. Starting from its dirty data and missing data probably will cause problems for our machine learning algorithm and modelling. So we needed



```
[ ] df[df["Category"] == "1.9"]

                                          App  Category  Rating  Reviews  Size  Installs  Type  Price  Content Rating  Genres       Last Updated
10472  Life Made WI-Fi Touchscreen Photo Frame      1.9    19.0     3.0M  1,000+     Free     0  Everyone             NaN  February 11, 2018   1.0.19

[ ] df.drop(df.index[10472], inplace=True) # Removing the index 10472 since it has a missing category value
```

to take care of cleaning the dataset. In this method, for dirty data for example, 1.9 value in category section we dropped that row as shown in the figure below.Also, we decided to delete the type column since it’s a redundant column because if the price is 0, it’s free else it’s bigger than 0, it’s paid application. Furthermore, we numerized the other categories just to test them easily within the code, because they were string or object variable.

Secondly, after cleaning the dataset as best as possible, we visualized our data to see which category gets the best average of installments and best average ratings, the reason of using bar graph is the average installment values and average rating values are discrete, we first tried the linear graph but it became complicated thus bar graph showed our data in a more useful way.

Moreover, we divided our dataset into three groups to serve as our train, test, and validation samples. We chose our sample size as 75%,12.5%,12.5% respectively. This decision was made as it led to the best performing machine learning model.

In addition, we used more than one model in order to compare and contrast the performance of the different models. We observed that random forest gave the best results for our use case and it was followed by decision tree.

6. CONCLUSION

Main idea of this project is that if a software team wants to publish an application which categories shall they look to determine their probability of being successful. In terms of an accomplishment of team’s application we have concluded that the number of installments are the key for success. We have focused on predicting the number of installments with the most related categories of data.

Given dataset needed a data cleaning to decide the machine learning model that we will use. We have concluded that several columns on the dataset are redundant for achieving our goal. Removal of those redundant columns helped us to develop a simpler and a reliable test opportunity.

After cleaning our dataset with right processes and finding the redundant columns and also fixing the NaN values we have moved on our next step. Visualizing the data needed 2 kinds of graphs. Mean values of installment counts in accordance with the categories of applications and mean values of rating respect to their categories, these are 2 graphs that we have chosen.

Values of 75%, 12.5%, 12.5% used for our split test. As we have stated above this is a regression problem. Random Forest model is appropriate one for achieving our statement. For comparing and making sure of our selection of the model is the best choice we have also tried a second model, decision tree. Accuracy of the random forest model was higher than the decision tree, so we became sure that our selection is right.

In conclusion a software team should consider predicting the number of installments of their application in terms of succeed. Determining the right parameters for a problem is the essential way to make a good start. This project of ours successfully showed that gathering together the right data and selecting the most suitable models is important.

7. FUTURE DIRECTIONS

Future directions requires more observations and experiments to choose the ways of improving our science project. Developing a simulation model can be considered and this will eventually help us create more and more examples to determine our future directions. We might look upon the past applications that has been published since the creation of Play Store and look for a pattern. Looking at the trends of the past years can help us to choose. Finding the correct data are only the beginning for making a popular releases, we also need to find the psychology of the humans that has installed the applications. This will lead us to in depth understanding.

8. DIVISION OF WORK

As we have stated in the past reports our group consists of five group members. Three computer science students and two industrial engineering students. In the contrast of two sciences we divided our team work in to sub-parts. Tuvan is assigned to our statement of purpose that what we are doing and why we are doing. This part was essential for seeing our future steps in this project. Clear description of our datasets we are using has been managed by Rutkay, in the light of his descriptions Canberk and Anıl has determined the methods that we have used. Data preprocessing, future generation and ML models has been conducted by them. The future steps that we are planning to take has been decided by Uğurcan. In the end we have used our time efficiently and wisely to found an optimal solutions for our problems. All team members have collaborated and helped each other. Exchanging ideas and selecting the most beneficial one's made our task easier.