Most Popular Google Play Store Apps

Dhruval Patel
dpatel175@student.gsu.edu

Samantha Newkirk
snewkirk1@student.gsu.edu

Janki Patel
jpatel134@student.gsu.edu

Abstract

We are living in the "information age” is a popular saying; however, we are actually living in the data age. Terabytes or petabytes of data pour into our computer networks, the World Wide Web (WWW), and various data storage devices every day from business society, science and engineering, medicine, and almost every other aspect of daily life. This explosive growth of available data volume is a result of the computerization of our society and the fast development of powerful data collection and storage tools. Businesses worldwide generate gigantic data sets, including sales transactions, stock trading records, product descriptions, sales promotions, company profiles and performance, and customer feedback. For example, large stores, such as Wal-Mart, handle hundreds of millions of transactions per week at thousands of branches around the world. Scientific and engineering practices generate high orders of petabytes of data in a continuous manner, from remote sensing, process measuring, scientific experiments, system performance, engineering observations, and environment surveillance. Data Mining also gives the way into utilizing the machine learning models and can be extended to deep learning methods such as CNN, AE [1] in real live scenarios such as twitter analysis [2].

Introduction

A dataset was selected from kaggle.com which is about Google Play Store Apps [3]. This dataset consists of over 10k records which are basically different apps of the same and different types. The dataset consists of 12 features for each app where it provides information regarding the app such as category, ratings, size, price, versions, etc.

Mobile apps are constantly being updated and new apps in constant development. For any need or problem there may exist hundreds of different apps, all of which declare their superiority over others.

For the motivation of this report, we intend to better understand which attributes successful apps have in common. With so many apps that exist today, we are interested in determining which genre of apps are the most popular, if apps that are paid for have higher ratings, what target audience has the most successful apps, are a few questions we intend to explore. Furthermore, a prediction model will be developed with the intention of predicting whether an app would be highly successful (above a 4.0 rating) based on certain attributes.

Data Preprocessing

Preprocessing is an important step to begin with for any data science project. Through the process a better understanding of the features of a dataset may be gained. Additionally, when problems are encountered, deciding how to deal with these problems can affect the outcome of the results later when models are used.

To begin, it is helpful to know how many missing values are present in the dataset. Beginning with missing values exploration is helpful because later as we transform the data further, missing values may inhibit the ability to work with the data. The Rating column had 1474 missing values and was the only feature with significant missing values. This column consisted of Ratings with floating point values ranging from 0.0-5.0, therefore the mean value of the column was determined and used to fill in where values were missing. There were twelve missing values left from other columns which were dropped from the dataset.

Further preprocessing was needed in order to transform the data points from object data types to numerical.

Install and Price columns had similar issues that were handled in similar ways. The data points within these columns were numerical though the values had extraneous symbols. In order to remove the unnecessary symbols a for loop was used to iterate through each value and through python methods the symbols were stripped. Additionally, the Size column had values in megabytes and kilobytes with extraneous symbols used to differentiate the two. These values were transformed to be one size, megabytes, and the symbols stripped as well. For the Reviews and Type columns a Python dictionary data structure was used to manually transform the values to numerical data. At this point most features had been transformed and then cast as numeric types. The remaining features were later transformed through the Sklearn library LabelEncoder when needed.

With preprocessing done, further data exploration can be conducted. Additionally, machine learning algorithms can also be implemented on the dataset. Further processing can be done in order to scale the data or deal with outliers if needed.

Visualization

The best way to understand any dataset is through visualization techniques by plotting different plots such as Bar graph, Pie chart, Scatter plot. etc. It gives us an idea of what our data really consists of. Upon evaluating some features of our dataset, we discovered the following information.

First step in data visualization was to find which app category is most popular. Therefore, we created a Pie chart as you can see in [Figure 1]. Based on this figure, we can tell that the Family category (18.2%) is the most popular category in the entire Google Play Store. Following that, we have the Game category (10.6%) as the second most popular kind of app. All categories' popularity is displayed in the figure in the form of percentage.

Rating visualization was performed through a Bar chart [Figure 2] and Distribution plot [Figure 3]. After analyzing the bar chart, we discovered that 4.2 rating is the most popular rating. Using the distribution plot, we learned that most apps are rated between 4.0 and 4.5 since the peak is highest for those values.

Next visualization that we performed is on the Types feature. This feature tells us how many apps are free versus paid [Figure 4]. From the figure we can say that 92.64% of the total apps available in the play store are free whereas 7.36% of the total apps are paid apps.

Content rating was performed through a Bar chart [Figure 5] to find the number of apps available for a specific age group. From the figure, we can tell that roughly 9,000 available apps are rated for all everyone, roughly 1,000 apps are rated for teens, and so on.

Now since we have a good idea of the popularity of category, rating, types and content rating feature, the next step is to learn what kind of games are most popular. To do this, we plotted a pie chart by collecting all the genres that belong in the game category [Figure 6]. Based on the chart, we can say that sports games (17.78%) are the most popular. Following that, action games (16.30%) is the second most popular game and third we have arcade games (9.83%), and so on.

Next we plotted a horizontal bar graph [Figure 7] to visualize what genre of family app is most popular. Based on the output, the Music & Video genre is the most popular among the family category. After Music & Video, it was Creativity, Education, and so on.

Scatter was plotted next to learn how Category and Ratings [Figure 8] relate with each other. As per the Scatter plot, we can say that app rating does not really affect the app category. We can confirm this statement since low rating and high ratings are spread evenly.

Lastly, we plotted another scatter plot of Rating versus Content Rating [Figure 9]. Based on the graph, we can say that apps that are rated for everyone have all kinds of rating from as low as 1.0 and as high as 5.0. Teen, Everyone 10+ and Mature 17+ were rated mainly between 3.5-5.0.

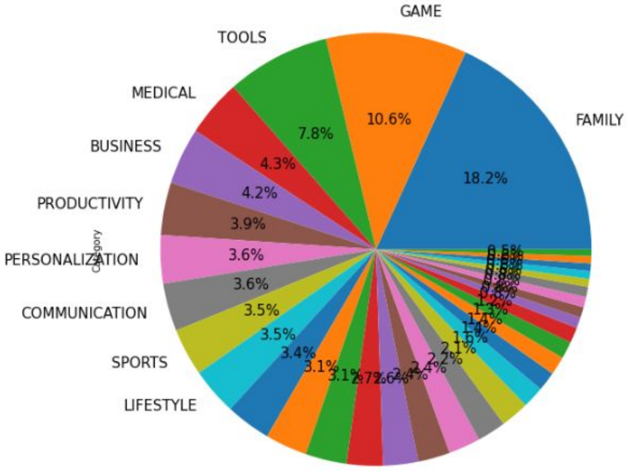

Figure 1: Category

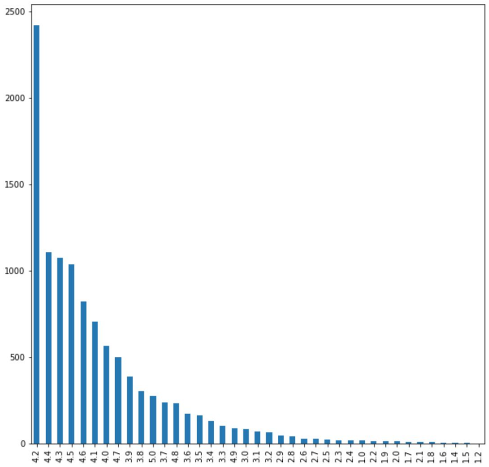

Figure 2: Ratings

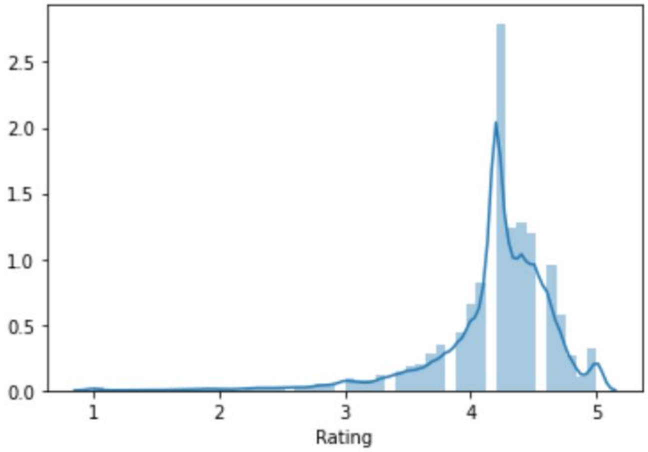

Figure 3: Ratings

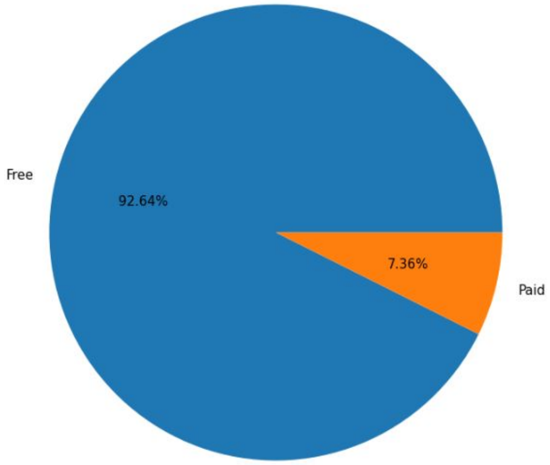

Figure 4: Type

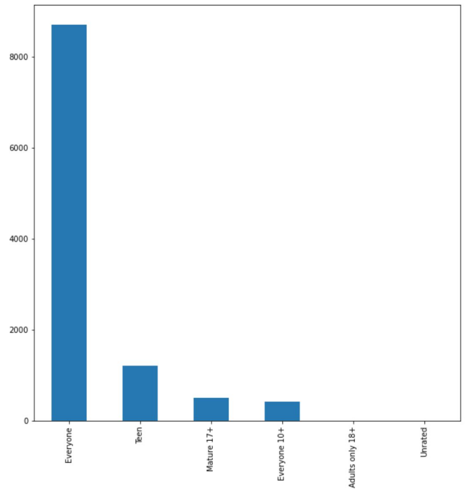

Figure 5: Content Rating

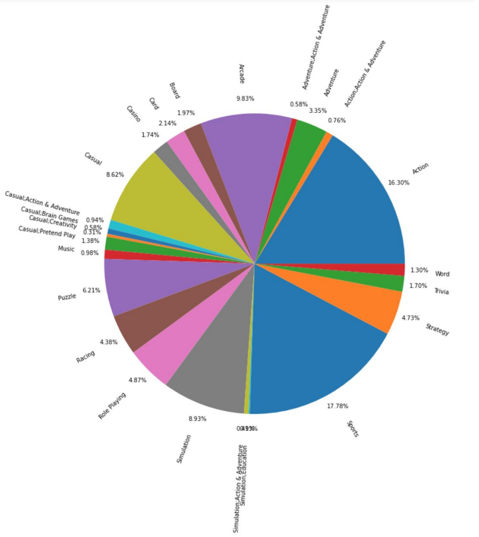

Figure 6: Category vs Genre (Game)

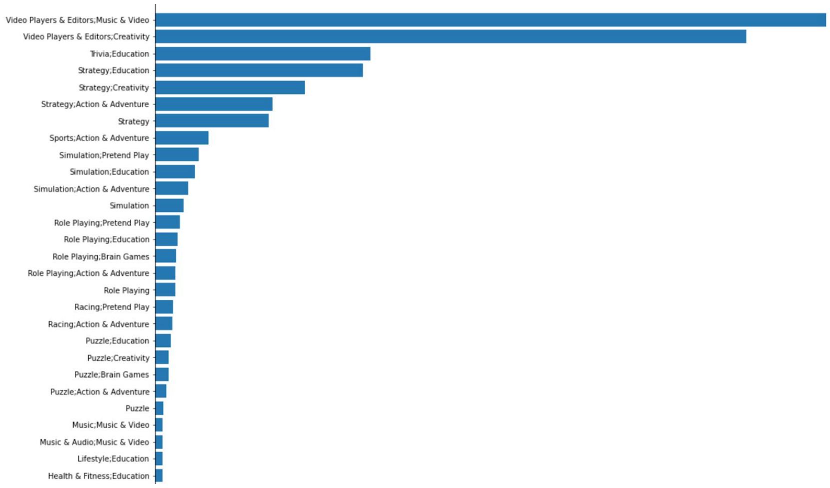

Figure 7: Category vs Genre (Family)

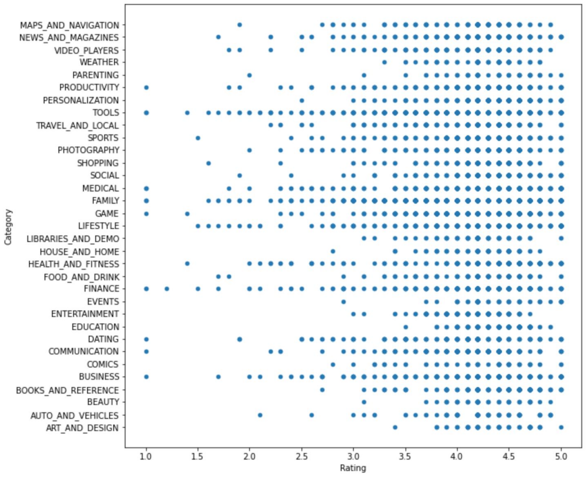

Figure 8: Rating vs Category

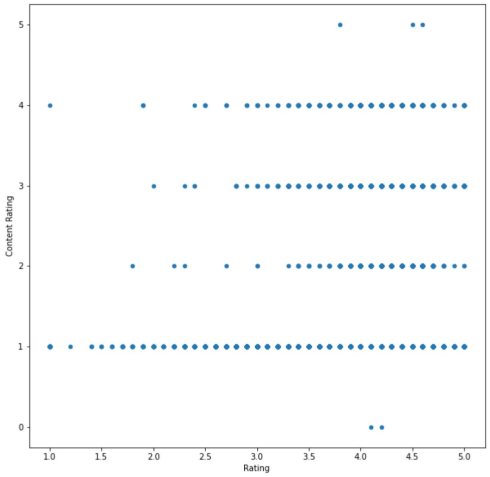

Figure 9: Rating vs Content Rating

Algorithms

In order to gain a clearer insight into the natural relationship among the columns of the dataset, a heatmap correlation matrix was generated [Figure 10]. Additionally, the z-score was determined in order to find outlier data points among the columns and those outliers were then dropped from the dataset

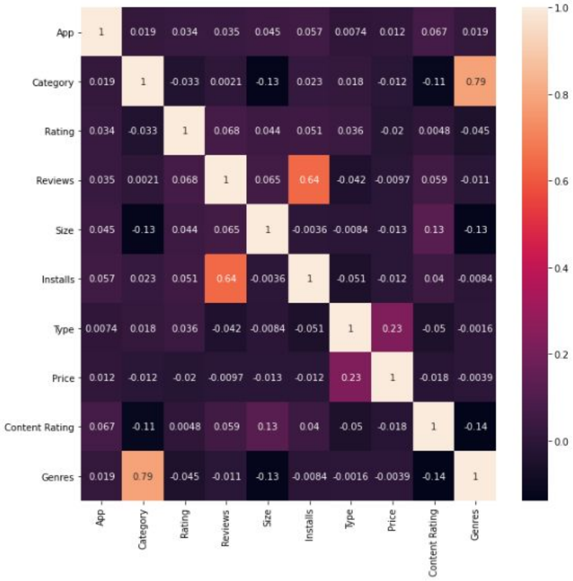

Figure 10: Correlation Matrix Heatmap

With the intention of further visualizing our dataset, K-means clustering was used. K-means generates k number of optimal clusters that the observations will be placed into based on their relation to the mean point (centroid) of each cluster.

Additionally, Decision Tree Classifier was used to create a prediction model to determine if an app, based on some input, would be highly rated.

Random forest was used to know that we do not need all the data to make the accurate prediction. Using this model we got the accurate results by using only two variables which have the same performance as others. At this point due to the use of random forest we were pretty confident that the results of our model was 80% accurate from MSE and the RMSE.

We also used KNN to predict if an app can have higher or lower ratings. When we printed the accuracy for the testing and training models the score was shocking because we found accuracy was really low for both of them. As the KNN algorithm does not work well with the categorical data the results were not accurate or the one we expected.

Conclusion

From the heatmap correlation matrix, only two sets of columns had a strong correlation. Category vs. Genres and Reviews vs. Installs. The latter is interesting, upon investigation there is an observed relationship that the more Reviews that an application has, then that app will also have more Installs. The z-score of the dataset was taken to determine any outliers that existed within the dataset. However, because data points within some of the columns were binary data, the outliers eliminated large amounts of data from the Type and Price columns.

Through the Elbow Method, the optimal number of clusters to implement through K-means clusters algorithm was determined to be two clusters. However, the centroids generated appeared linearly and did not reveal any new insights regarding the relationship among the data points.

The first prediction model implemented was a Decision Tree Classifier. To begin, the Ratings column was selected as the target for our prediction with the goal of predicting whether an app would be top rated or not. The data points of the Ratings column were converted to binary data with ratings of 4.0 and above selected as the threshold of success, anything less than 4.0 would be deemed unsuccessful. Decision Tree algorithms require that the target variable be discrete values, the algorithm makes binary decisions (yes or no) based on qualifications at each iteration, hence the name Decision Tree. The optimal depth of the Decision Tree was determined to be 5 and produced predictions with an accuracy score of 81.04%.

Reference

[1] Jaya Krishna Mandivarapu, Blake Camp, and Rolando Jose Estrada. Self-net: Lifelong learning via continual self-modeling. *Frontiers in Artificial Intelligence*, 3:19, 2020

[2] S. T. Sadasivuni and Y. Zhang. Using gradient methods to predict twitter users' mental health with both covid-19 growth patterns and tweets. *second IEEE International Conference on Humanized Computing and Communication with Artificial Intelligence (HCCAI 2020)*

[3] L. Gupta, “Google Play Store Apps,” Kaggle, 03-Feb-2019. [Online]. Available: https://www.kaggle.com/lava18/google-play-store-apps?select=goog