Analysis of the Google Play Store Apps and Reviews

Sabrina F

July 2020

1 Introduction and Background

Apps govern our daily lives, with everything from tracking our diets to games. So we set off with a Kaggle data set from 2018 analysing the meta data of the app store apps at that time, in hopes to gain a better understanding of the types of popular apps on the app store. In essence, what principle properties govern a successful app, what types of apps are most popular? What are some properties in apps reviews that can help us understand more successful apps.

From developers to business experts to artists, we hope this analysis will help guide decisions in developing future apps and technologies. By determining various properties of successful apps that users like and respond to well. This information will be useful to understanding how to engineer future technologies that require app like interfaces.

2 Methods

We conducted our analysis into two parts, using unsupervised learning techniques of k-means clustering, PCA, and decision trees to better understand the structure of the data and various word count methods to better understand the properties of app store reviews, in hopes to understand how people view the apps they use.

In the extensive data cleaning process the app info data set was had 10841 row and 13 columns (10841×13) after dropping the rows with unfilled data, the data set set became 9360×13.

For the app review data set the processed data was reduced from 64295×5 to (37427×5) after dropping rows with empty column data.

The sample size of our app data information is statistically significant within a 95% Confidence Interval and 2.5% margin of error. Using the off hand rule of $N = \frac{1}{Margin-of-Error^2}$

Because most of the data is categorical, we transformed the data using one hot encoding. In some cases such as for counts of reviews, numbers were rounded to the nearest thousand, as to preserve the orders of magnitude and to make the one hot encoding have less parameters.
For the review data, I stripped all punctuation and set the case of all words to lowercase in order to allow easier understanding of word frequencies. While I wanted to dive deeper into the various apps and corresponding reviews from the two data tables, I found that there was not a representative sample from both data sets to perform a robust analysis.

3 Analysis

3.1 Unsupervised learning analysis of App Ratings

We utilized various unsupervised learning techniques in our analysis of the data using K-Means Clusters, Principle Component Analysis and decision trees in an attempt to find identifying features of the data

K-Means Clustering Using the skilearn k-means clustering package, we attempted to see if the data would naturally form clusters based on input of score ratings, content categories, audience types, number of downloads, and number of reviews. After accounting for the categorical data and standardizing it, the normalized score of my clusters was -41301028.676314965 The number of clusters that minimized the score was when $k = 2$. Due to the normalized score being very far from zeros, the k-means method did not yield any conclusive results.

Principle Component Analysis

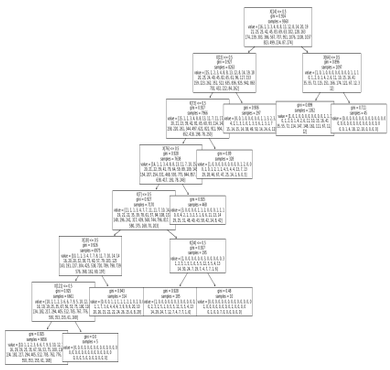
We then attempted to conduct a principle component analysis of the app store data in hopes to find some identifying features of the data. We used various parameters to visually discover if there was any single feature that is a principle component of the app store data. However after comparisons with App Ratings, Installs, App ratings over various thresholds, and genres; there was no statistically compelling evidence that there was any principal components.

Decision Trees

Finally, we constructed a decision tree model to understand how the data natural divides. While this model did yield the most promising results, the Gini score was still not optimal. However the decision tree

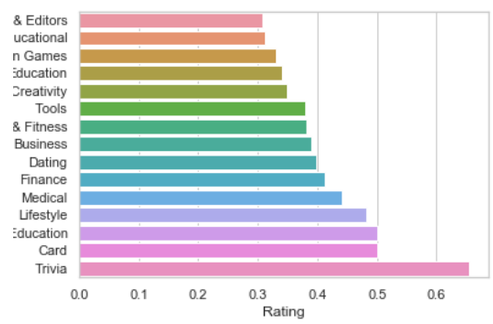
did shed light on the different categories and types that data could be split. By creating a decision tree with a maximum of 10 leaf nodes, the tree splits in regards in the following order by the Game genre, Health and fitness category, Casual category, Communication, education, dating, lifestyle, everyone rating, and puzzles.

The very poor gini scores was mostly likely responsible for the non uniform distribution of various types of apps, for example there are lots of social apps while there are proportionally less weather apps. While we did standardize and account for the categorical data, the disproportionate data and unequal variance between categories would obscure these machine learning techniques. The figure below shows the decision tree, with each node from left to right in order game genre, Health and fitness category, Casual category, Communication, education, dating, lifestyle, everyone rating, and puzzles.



3.2 Analysis of App Type Variations

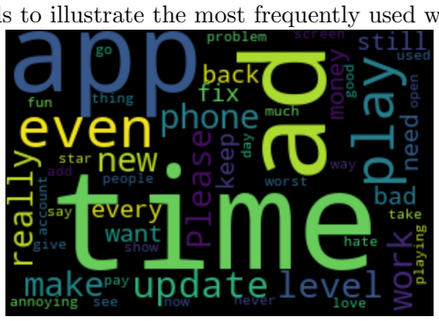
To better understand the variation across app categories we plotted the mean variation of each categories and made a plot of the top 10 categories with the most variation and a plot of the top 10 categories with the least variation.



As we can see from the above plots, the apps with the msot variation are games and educational apps.

3.3 Analysis of App Reviews

To begin my analysis we sorted the reviews by sentiment, positive and negative, and then we created word clouds to illustrate the most frequently used words.



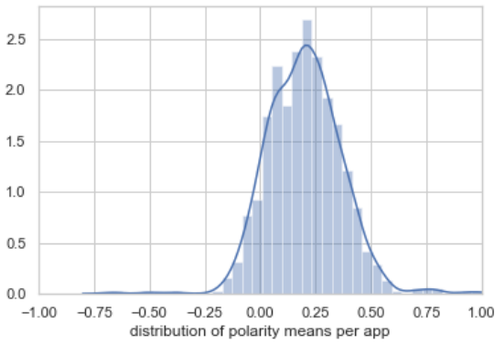
After taking frequency counts of positive reviews and negative reviews the following statistics were found.

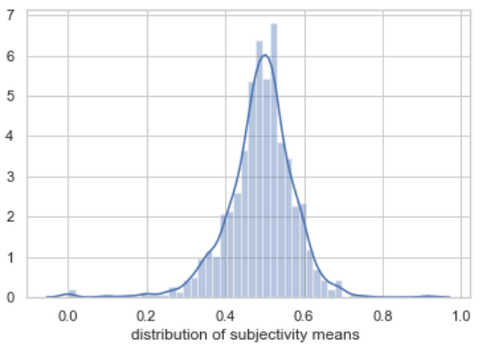
positive reviews mean 7.227360594795539
negative reviews mean 6.972514887769125
positive reviews variance 8.764069241718609
negative reviews variance 10.976706776576336

By conducting the following hypothesis test, to determine if the difference between a positive review and negative reviews is zero. With a p-value of less than 0.00004, we can reject the null hypothesis and confidently say that the average length of a positive review is greater than an average negative review.
$H_0 : \mu_{poslength} - \mu_{neglength} = 0$ $H_A : \mu_{poslength} - \mu_{neglength} < 0$
In fact we can say with 99.8% accuracy that the word length on average between a positive and negative review will differ by about 3.5 words.

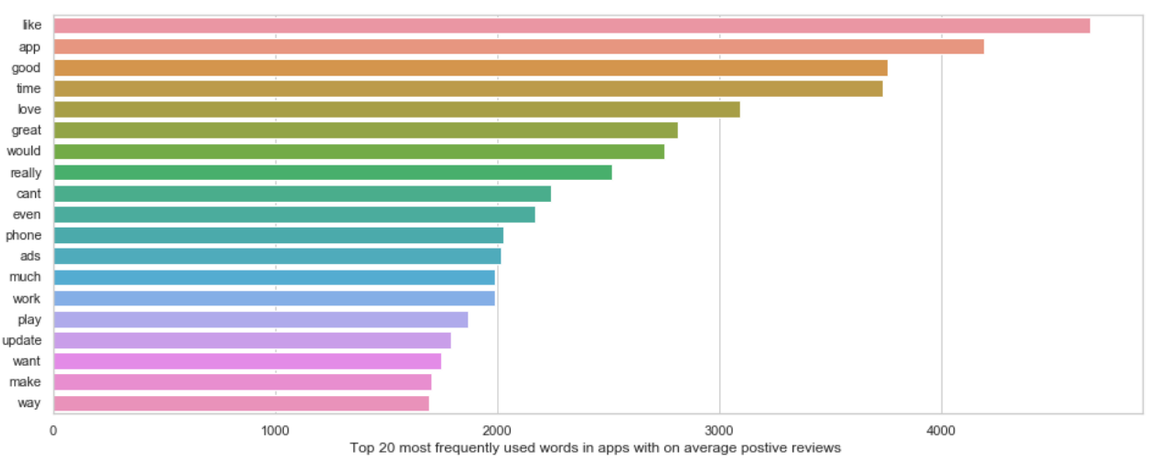
3.4 Sentiment & Polarity Analysis

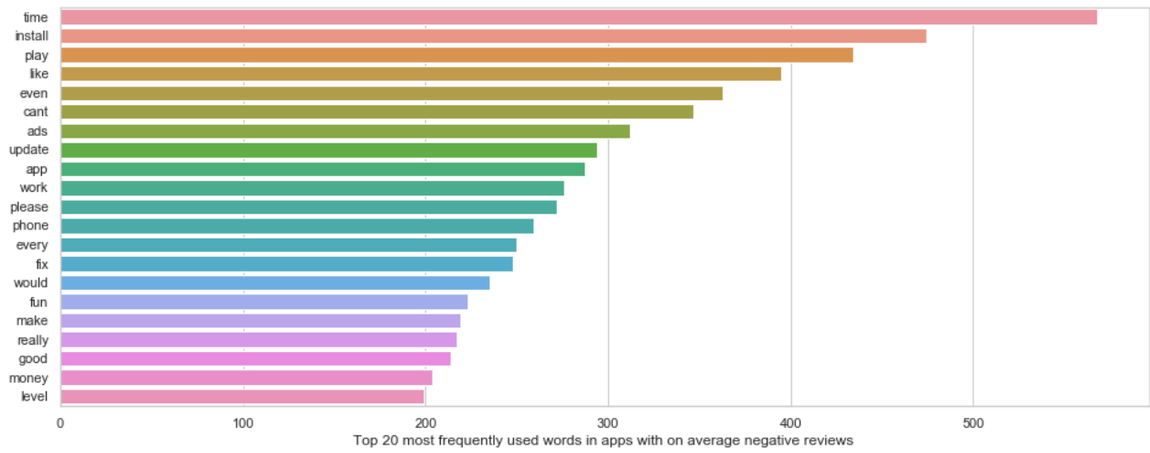
We then analyzed the distribution of polarity and subjectivity means on each categories





From the two above images we can see that while the subjectivity follows a normal distribution, the polarity of our reviews tends to skew positive. That is app reviews tend to be more positive, which supports our earlier findings of average review rating.
We then sorted apps based in their mean sentiment score, if an app on average had mean sentiment greater than 0, it was labeled a positive app. We then graphed the most frequently used words of those two classes.





As we can see from the bar charts, there are similar words in both negative and positive reviews. However in the negative reviews, the mentions of logistical words such as "update", "install" or "fix" shows that the logistics of an app working is very important. While in the positive review we see more positive words and words that do not mention logistics–in fact most of the words in positive rated apps tend to be more normal words such as "would", "like", "want".

4 Results and Discussion

So what can be learned of the Google Play Store app analysis?
From the above graphs we can see that most ratings and reviews of apps tend to be mostly positive. While we assumed in all our data that the reviews and ratings of these apps were all from people reviewing an app they actually used and downloaded. There may be phony reviews and ratings, however since there are not enough reviews and ratings to properly link together, there is not much we can do.
However from looking at the frequently used words of apps with mostly positive or negative reviews, we can see that the negative review users tend to dislike "ads", "time", "install", "fix"– suggesting that many apps had issues that users tended to avoid.
From the distribution plots of polarity means, we can see that the app polarity is again skewed right, with most apps having on average favourable reviews. In fact, even with the apps with negative polarities, the polarities rarely go below -0.25 with most negative polarities being just below 0. So what is going on here? Why do most apps tend to have overall positive reviews? This could be one of a few factors, the apps in the data set contained some bias or the apps had a disproportionate amount of positive reviews. Which would make sense if a user was frustrated with an app, uninstalled the app and didn't bother to take the time to leave a review.
While we did not find a robust model to predict an apps success based on its parameters from the app store, the decision tree model did articulate various splits that showed division of data. Splits such as education, dating, lifestyle, communication, and puzzles; show that that is where lots of variation of reviews lead. Which is backed up by the bar plots of variation in section 3.2.

5 Suggestions for Future Apps

While there are no definite ways to accurately predict the ratings and reviews of apps based on this data set. In the future this perhaps could be remedied by including the source code of the apps and comparing abstract syntax trees of the code of successful and unsuccessful apps.
However in suggestions for future apps, developers should take care in creating interfaces and models that do not waste users time, and are not glitchy. Great care and testing should be taken with apps that are created that would be classified in the high variation categories such as education and games, to ensure that users will actually use and interact with apps. Finally developers should constantly monitor the app reviews for potential bugs as well as the length of such reviews, as our analysis showed negative reviews tend to be longer. So developers should take care to ensure that long reviews are quickly resolved.