Marketing Analytics Analytics Final Project

Corey Austen

April 26, 2019

# Problem Formulation

This project will seek to find the find the optimal app to develop for a new startup app company. I hope to find key factors to increase the amount of installations for an app and gain insight into what app would be most likely to be downloaded over 1 million times. The project will also enable app makers to focus on key factors in promoting their app, preventing the wasting of time and resources.

# Data description

The first data set, ‘data’, is summary data on each app in the Google Play Store. The dataset contains 10,841 rows and 13 variables. The variables are app, category, rating, reviews, size, installs, content.rating, genres, last.updated, current.ver, and android.ver.

* “App” (factor) is the name of the app as it appears in the Google Play Store. Interestingly, the apps are not unique, with 9660 factor levels in 10,841 rows. There are very likely duplicates that will need to be cleaned up.
* “Category” (factor) is the overall type of app, including social media, communication, etc. It is set as a factor with 34 levels.
* “Rating” is the current consumer rating given on the Google Play Store. This is a numerical value.
* “Reviews” is the number of reviews collected for the app. This is currently a factor with 6002 levels. I think this needs to be transformed into a numerical value.
* “Size” is the amount of space the app takes up in the physical harddrive of the device. This is set as a factor with 462 levels. This will need to be transformed into a numeric value.
* “Installs” is the number of times the app has been installed on consumer devices. This is currently a factor with 22 levels. This will need to be transformed into a numeric value. This is also the dependent variable.
* “Type” describes the app in terms of purchase, whether its free, paid, etc. This is a factor with 4 levels, 0, Free, NaN, and Paid. This seems redundent because the price of the app is also listed, and the only real types here would be Free or Paid. This needs to be corrected to have only 2 levels and will be removed in the analysis.
* “Price” is the amount paid for the app in the Google Play Store. This is currently set as a factor with 93 levels. This will need to be transformed into a numerical value.
* “Content.Rating” is a factor that describes the audience the app is appropriate for. There are 7 levels, including “Adults Only 18+”, “Everyone”, “Teen”, etc.
* “Genres” is a factor that is similar to “Category”, but categorizes the app more specifically. This has 120 levels, including “Action”, “Health & Fitness”, “Music”, etc.
* “Last.Updated” is a factor containing the date the app was last updated on the Google App Store. This contains 1378 levels. This will need to be transformed into a date/time rather than a factor.
* “Current.Ver” is the software version that the app is on. This is listed in 0.0.0 style, or listed as “Varies with device” if the app has different versions for each device. This is currently listed as a factor with 2834 levels. This needs to be converted to a numerical value and “Varies with device” needs to be converted to NA.
* “Android.Ver” is the Android version that the app is available on. This is listed in 0.0.0 style, or listed as “Varies with device” if the app has different versions for each device. This is currently listed as a factor with 35 levels. This needs to be converted to a numerical value and “Varies with device” needs to be converted to NA.

The second dataset is the raw reviews for apps. This dataset includes 64,295 rows with 5 variables: App, Translated\_Review, Sentiment, Sentiment\_Polarity, and Sentiment\_Subjectivity.

* “App” is the name of the app as it appears in the Google Play Store. Interestingly, the apps are not unique, with 1074 factor levels in 64,295 rows. There are obviously multiple entries for each app since each row is a seperate review, so this is not an issue.
* “Translated\_Review” - This is the raw text for each review. This text has been translated via the Google English Translation Dictionary. This is listed as a factor with 27996 levels. This is interesting, as this means that there are a lot of reviews with the same text. This may not be an issue, as they could be very short, like “Good” or “Bad”. This will need to be converted into character values.
* “Sentiment” is the label of the overall sentiment of the text review. This is determined via a sentiment analysis. The data is a factor with 4 levels, including “Positive”, “Negative”, “Neutral”, and “nan”.
* “Sentiment\_Polatity” is the numerical score of the sentiment analysis, ranging from 1 (very positive) to -1 (very negative), with 0 as neutral.
* “Sentiment\_Subjectivity” is the numerical score for the text review, where 0.0 is very objective and 1.0 is very subjective.

One advantage to using this dataset is that it contains thorough metrics on all the apps in the Playstore. The data appears to be accurate and can be easily broken down into subsets based on the type of app, category, genres within those categories, etc. This also contains data on the applicable operating system that is required by the app and the data it was last updated. Another advantage is that the data already contains a sentiment analysis on the translated reviews of the apps, so the interpretation of the reviews required very little work.

However, this data is a bit dirty. There are duplications in the dataset, the installation rates are not specific. The data includes apps that are automatically installed on devices, which would need to be removed to give an accurate portrayal of consumer installations. The data doesn’t include any information on the demographic that actually downloaded the app, so it has to be assumed based on category, genres, and rating. The review text data was also fairly dirty, so it required cleaning. Some additional data on who the reviewer is (age, location, gender, etc.) would have been useful as well.

In this data, I assume that we will see associations between price and installs, since a free app seems more likely to be downloaded than an app you must pay for first. I would also imagine that we would see associations between rating and installs because we can assume that consumers would be less likely to install an app that is rated poorly. Another association we can assume to see is reviews and installations. An app that is downloaded more is likely to have more reviews.

### Data Cleaning

To clean up the data, I will need to do the following: - Remove symbols (+ and “,”) from the Installs variable, then convert it to a numeric value - Remove characters (M and k) from the Size variable, replace any value that contained a k to 0 since it would be less than 1M, then convert it to a numeric value. - Convert Reviews variable to a numeric value. - Remove the $ character from the Price variable, then convert it to numeric. - Convert the Last.Updated to a MDY date format. - Create a new column called Min.Android.Ver. Insert Android.Ver into this column. Change “Varies with Device” string to NA. Then convert the data to numeric. Remove the old column. - Filter out the rows that do not include the Type “Free” or “Paid”. - Drop any unused levels from factors so we can get a clearer view of the data.

data.clean <- data %>%  
 mutate(  
 # Eliminate some characters to transform Installs to numeric  
 Installs = gsub("\\+", "", as.character(Installs)),  
 Installs = as.numeric(gsub(",", "", Installs)),  
 # Eliminate M to transform Size to numeric  
 Size = gsub("M", "", Size),  
 # Replace cells with k to 0 since it is < 1MB  
 Size = ifelse(grepl("k", Size), 0, as.numeric(Size)),  
 # Transform reviews to numeric  
 Reviews = as.numeric(Reviews),  
 # Remove currency symbol from Price, change it to numeric  
 Price = as.numeric(gsub("\\$", "", as.character(Price))),  
 # Last Updated to date format  
 Last.Updated = mdy(Last.Updated),  
 # Replace "Varies with device" to NA since it is unknown  
 Min.Android.Ver = gsub("Varies with device", NA, Android.Ver),  
 # Keep only version number to 1 decimal  
 Min.Android.Ver = as.numeric(substr(Min.Android.Ver, start = 1, stop = 3)),  
 # Drop old Android version column  
 Android.Ver = NULL  
 ) %>%  
 filter(  
 # Two apps had type as 0 or NA, they will be removed   
 Type %in% c("Free", "Paid")  
 )  
data.clean <- droplevels(data.clean)  
str(data.clean)

## 'data.frame': 10839 obs. of 13 variables:  
## $ App : Factor w/ 9658 levels "- Free Comics - Comic Apps",..: 7227 2563 8996 8111 7292 7123 8169 5588 4947 5825 ...  
## $ Category : Factor w/ 33 levels "ART\_AND\_DESIGN",..: 1 1 1 1 1 1 1 1 1 1 ...  
## $ Rating : num 4.1 3.9 4.7 4.5 4.3 4.4 3.8 4.1 4.4 4.7 ...  
## $ Reviews : num 1183 5924 5681 1947 5924 ...  
## $ Size : num 19 14 8.7 25 2.8 5.6 19 29 33 3.1 ...  
## $ Installs : num 1e+04 5e+05 5e+06 5e+07 1e+05 5e+04 5e+04 1e+06 1e+06 1e+04 ...  
## $ Type : Factor w/ 2 levels "Free","Paid": 1 1 1 1 1 1 1 1 1 1 ...  
## $ Price : num 0 0 0 0 0 0 0 0 0 0 ...  
## $ Content.Rating : Factor w/ 6 levels "Adults only 18+",..: 2 2 2 5 2 2 2 2 2 2 ...  
## $ Genres : Factor w/ 119 levels "Action","Action;Action & Adventure",..: 10 13 10 10 12 10 10 10 10 12 ...  
## $ Last.Updated : Date, format: "2018-01-07" "2018-01-15" ...  
## $ Current.Ver : Factor w/ 2833 levels "","0.0.0.2","0.0.1",..: 121 1020 466 2826 279 115 279 2392 1457 1431 ...  
## $ Min.Android.Ver: num 4 4 4 4.2 4.4 2.3 4 4.2 3 4 ...

### EDA

The data has two additional problems that need to be addressed. The first is that some of the apps included in the data set are pre-installed onto android devices. This includes Google apps (Android is owned by Google) and some popular social media apps, like Facebook and Instagram. Because the installation rates are not a reflection of how the vast majority of apps will be downloaded, these apps should be removed from the dataset.

The second problem is duplicate app entries in the dataset. There are several apps that are entered multiple time because they are different versions of the same app. This will be corrected by keeping only the most recent version of the app (based on Last.Updated).

drop.apps <- c('Google Play Books','Messenger â???" Text and Video Chat for Free','Google Chrome: Fast & Secure','Gmail','Hangouts','Google Play Games','Facebook','Instagram','Google Photos','Maps - Navigate & Explore','Google Street View','Google','Google Drive','YouTube','Google Play Movies & TV','Google News', 'Google+', 'Google Calendar')  
  
data.clean <- data.clean[ ! data.clean$App %in% drop.apps, ]  
reviews <- reviews[ ! reviews$App %in% drop.apps, ]

data.clean <- distinct(data.clean)  
data.clean <- data.clean[ !duplicated(data.clean[, c("App", "Last.Updated")], fromLast=T),]

The App names should also be removed and set as the row names since they will not be used in the analysis.

Managing the NA values will be important. Some NA values are okay, but others may need to be removed or replaced with an average.

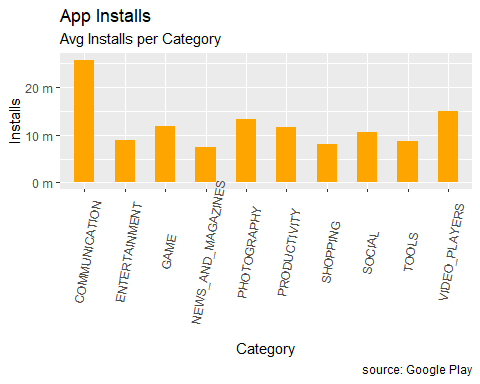
## Take a look at NA values

A picture containing text

Description automatically generated

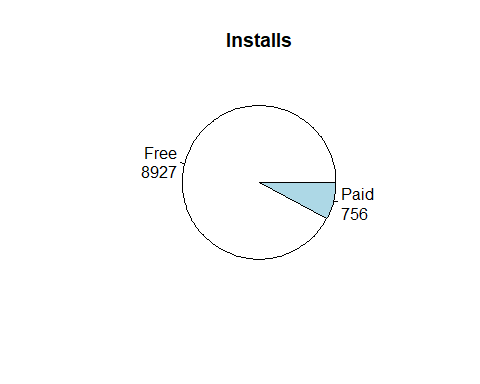
## Look at the installs

Since app installation is the dependent variable, I’ll take a peek at the top 10 most installed categories. This shows communication as the most installed, followed by video players, photography, and games. This is not suprising.

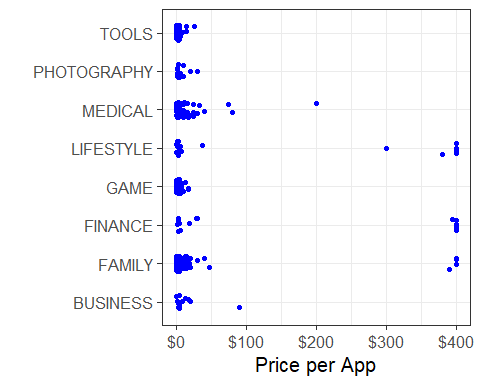


## Paid App Exploration

We can see that about 8% of apps overall require purchase at the time of download, but this varies by category.



Just out of curiosity, the price range for the paid apps is fairly small, with the exception of some extreme outlyers ($400!) in Medical, Lifestyle, Finance, and Family.



Looking at the list of the apps with the highest price tag, we can see these apps are mostly scams.

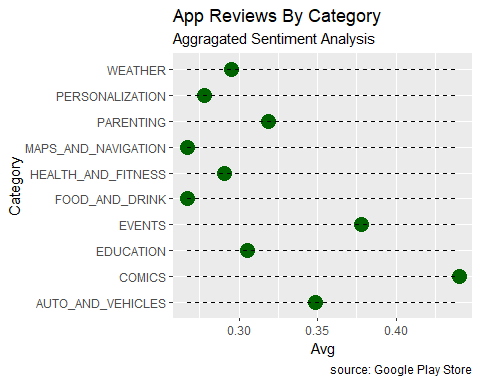
paid %>%  
 select(App,Category, Price) %>%  
 arrange(-Price) %>%  
 head(10)

## App Category Price  
## 1 I'm Rich - Trump Edition LIFESTYLE 400.00  
## 2 most expensive app (H) FAMILY 399.99  
## 3 ð I'm rich LIFESTYLE 399.99  
## 4 I am rich LIFESTYLE 399.99  
## 5 I am Rich Plus FAMILY 399.99  
## 6 I Am Rich Premium FINANCE 399.99  
## 7 I am Rich! FINANCE 399.99  
## 8 I am rich(premium) FINANCE 399.99  
## 9 I Am Rich Pro FAMILY 399.99  
## 10 I am rich (Most expensive app) FINANCE 399.99

## Review Sentiment Analysis

In order to use the sentiment analysis in the review text dataset, we need to pull “Sentiment\_Polarity” from the review data set into the data set with the corresponding app data, data.clean. But since there are apps in the data.clean dataset that do not have corresponding reviews in the review dataset, I will need create them. I will do this by assigning those apps a Sentiment\_Polarity of 0, since 0 is considered “neutral” in the sentiment polarity scale.

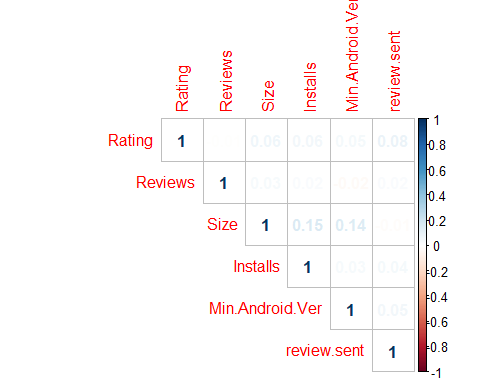
I went ahead and compiled the average sentiment polarity for each category. Interestingly, all categories have a positive sentiment polarity. I pulled the top 10 categories. I did not expect to see “Comics” and “Events” on this list. This makes sense though, considering there are relatively few reviews when compared to others, like “Health and Fitness.”



## Category Avg Volume  
## 1 COMICS 0.4400312 46  
## 2 EVENTS 0.3779327 158  
## 3 AUTO\_AND\_VEHICLES 0.3486827 289  
## 4 PARENTING 0.3188804 260  
## 5 EDUCATION 0.3053621 524  
## 6 WEATHER 0.2949783 201  
## 7 HEALTH\_AND\_FITNESS 0.2906940 2250  
## 8 PERSONALIZATION 0.2782125 1004  
## 9 MAPS\_AND\_NAVIGATION 0.2672184 192  
## 10 FOOD\_AND\_DRINK 0.2668824 638

## Correlation

The first part of the analysis is to look for any obvious correlation between several of the variables. I will look for correlation between “Rating”, “Reviews”, “Size”,“Installs”,“Min.Android.Ver”, and “review.sent.” I tried to include “Category” and “Genres” in this part of the analysis, but was unable to compile it.



Based on this, there is no significant correlation between the variables.

# Models

Before building the models, I will remove NA values for each app from “Rating” and “Size” with the mean values for the corresponding Category. I will also create a new variable, “ind”, which will be an indicator for whether the app has >= 1 million installs (1) or less (0). Then I will create test and train datasets at 25:75 split respectively.

df.2$ind <- ifelse(df.2$Installs>999999, 1, 0)

## Linear Regression

I’ve tested 2 different linear regression models. The first (glm.1), was built using the following variables: Category, Rating, Reviews, Size, Price, Content.Rating, Min.Android.Ver, Last.Updated, and review.sent with “ind” as the dependent variable. The second model (glm.2) Rating, Reviews, Size, Price, Category, Last.Updated, and review.sent with “ind” as the dependent variable. I chose to remove Genres because this is subcategory that is very closely related Category and both wouldn’t provide additional information and would greatly expand the amount of variables.

I am hoping the Linear Regression model can show me what variables are most significant with respect to the “ind” variable. I hope that this will also show what variables impact the dependent variable either positively or negatively. This will help determine what type of app a developer should create, and what factors could affect consumer popularity.

Based on the accuary of the models, the AIC, and testing for variance inflation (testing multicolinearity through the vif function), the second model (glm.2) is the chosen model.

glm.1 <- glm(ind ~ Category + Rating + Reviews + Size + Price + Content.Rating + Min.Android.Ver + Last.Updated + review.sent, data = train)

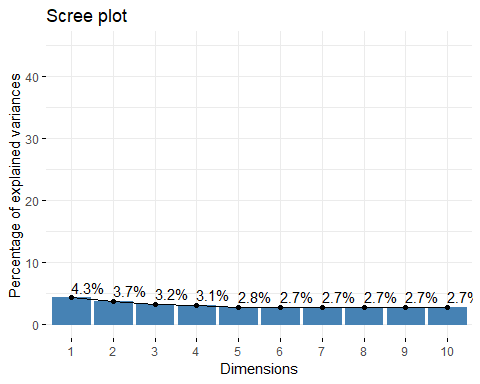
glm.2 <- glm(ind ~ Rating + Reviews + Size + Price + Category + Last.Updated + review.sent, data = train)

glm.1 Accuracy 0.7548298

glm.2 Accuracy 0.7571297

## Multiple correspondence analysis

The next part of the analysis is to run a multiple correspondence analysis on the factors in the dataset. This will be done to see if the number of variables can be reduced while keeping an acceptable level of varience. Based on the graph below, we can see that MCA is not a viable option and won’t be used.

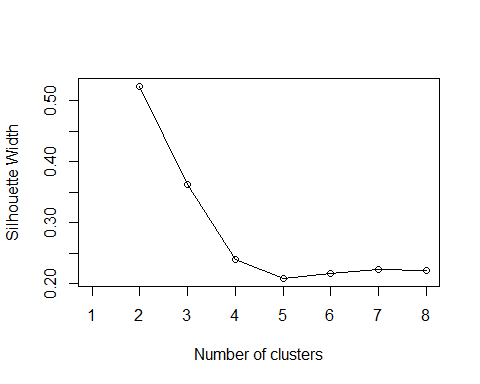


## Clustering

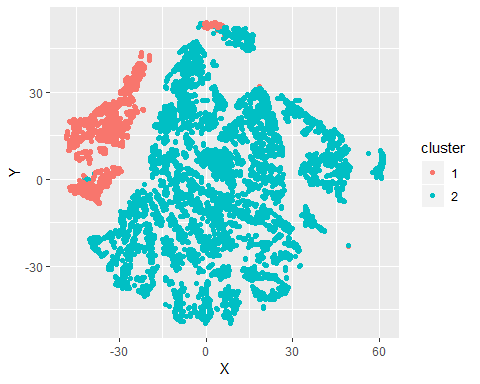
Clustering will likely show some unseen associations among the variables. I decided to only include 6 variables, using Rating, Reviews, Size, Price, review.sent, and Content.Rating. I dropped the category because when I ran the clustering analysis previously, the clusters formed around them. The reason I didn’t include genres because it is a factor with a lot of levels and is very closely related to category, so the clustering would likely just reflect this association.

The clustering model will hopefully show associations between some variables that would otherwise go unnoticed. I am also hoping that this will give some information on the type of app that would be associated with higher installation rates.

The optimal number of clusters was found using the Gower distance and calculating the silhouette width. The optimal number of clusters is 2.



set.seed(42)  
k <- 2  
pam\_fit <- pam(gower\_dist, diss = TRUE, k)



Freqs.inst <- table(w.clust$ind, w.clust$"pam\_fit$clustering")  
Freqs.inst

##   
## 1 2  
## 0 777 5491  
## 1 698 2717

# Results and discussion

The key findings from the analysis are that content ratings play a larger role in people downloading apps than previously thought. As stated before, the analysis found that there are two clusters, cluster 1 being significantly smaller than cluster 2. Looking at the cluster summaries, the clusters are divided up based on the variable “content.rating”, specifically the ratings “Teen” and “Mature 17+”. There also seems to be a higher concentration of “Games”, “Dating”, and “Social” in the categories of cluster 1 than in cluster 2. The most prominent “Genres” in cluster 1 are also “Action”, “Entertainment”, “Dating”, and “Social.” Cluster 1 also contained a higher percentage of apps downloaded over 1 million times than that of cluster 2.

The linear regression also revealed some important information about this data set. The Categories that were significant include: communication, education, entertainment, game, maps and navigation, personalization, photography, productivity, shopping, social, tools, travel and local, video players, and weather. Other variables that are significant are Rating, Reviews, Price, Size, Min.Android.Ver, Last.Updated, and review.sent. Looking at the coefficients also revealed that the review sentiment plays a very powerful role getting very large amounts of consumer installations.

The expected findings were the significance of ratings, reviews, and the review sentiment. It can be assumed that higher ratings and more positive reviews will increase the number of installations.

However, I did not expect to see the overall output produced by the cluster analysis. This is unexpected because I had not anticipated the content rating to be associated with higher installation rates. This could be because the overall demographic appears to be teens and young adults. This demographic may be more likely to own an android device and download an app on a whim than other android users. Teens and young adults would be the target audience for games, specifically “action” games. They are also the target audience for dating and social media apps.

# Recommendations

Based on the results from the 2 models and the exploratory analysis of the data, we can formulate a potential target audience for a new app, as well as the type of app it should be. As stated before, the cluster analysis showed that apps with content ratings of “Teen” and “Mature 17+”, categories of “Games”, “Dating”, and “Social”, and genres of “Action”, “Entertainment”, “Dating”, and “Social” are associated with installation rates of over 1 million. We also see in the linear regression model that the variables Rating, Reviews, Size, Min.Android.Ver, Last.Updated, Price, and review.sent are significant when analyzing the dependent variable. These insights point to types of apps marketed toward teens and young adults.

The recommended app to create would one of the following: - A mobile action game that is rated Teen or Mature 17+. Because this game has a higher rating, it is likely violent and includes content not intended for children. - A social or dating app targeting young adults. This could be an app for consumers seeking plutonic or romantic relationships.

Both apps should be a free to install, meaning it does not cost the consumer to download the app onto the device. However, the include could include in app purchases to boost revenue. The app should be updated often and run on as many android versions as possible.

The apps are more likely to be downloaded if they have a high number of reviews and a high rating. Some ways to achieve this would be: - The game could be reviewed by a Youtuber will a large audience. This would encourage additional consumers to install the game. - The dating/social app could be reviewed by social media influencers with a large following. This would encourage additional consumers to install the app. - If in-app purchasing is included in either app, a coupon or discount could be given to users that leave a review for the app. Apps that receive reviews tend to receive positive reviews more often the negative ones, so even if negative reviews are posted, more positive reviews will likely be posted. We can see that high numbers of reviews results in more installations, regardless of sentiment.

## Limitations

Some of the limitations with this dataset is that it doesn’t show any consumer demographic data. The target audience is built on an assumption based on the rating. However, if there was some sort of information regarding the consumer, such as age range, location, gender, device, etc., the models would much more robust. Another limitation is that it doesn’t include any information about whether the apps include in app purchases. This would also give some insight into whether including them deters consumer installations. If we had this included in the model, we could potentially determine a marketing strategy to generate additional revenue.