Google PlayStore Analysis

Philip Yoon

Abstract

When planning to list a mobile application, there is already a list of factors one must consider in order to create a successful app. What kind of app should I make? Does rating or install size even matter to increase the number of users I attract? In this project I analyze data from the Google PlayStore and address general characteristics of an app that contribute to high install counts. I use visualizations like boxplots and histograms to find that free, highly reviewed, large file size apps in categories with a large audience maximize install count. I fit a Ordered Logistic Regression model that can classify what quartile an app will fall under given general information about the app with 63.35% accuracy.

1 Introduction

For all app developers, the Google PlayStore is a must-enter market to list any app. In 2019, Google's PlayStore recorded an impressive 72% of global app downloads in the smartphone market. In order to creat a successful app, a high install count is essential; with more downloads comes more exposure and thus more opportunities for profit.

When creating the app, decisions such as the type of app, free vs paid, and size are all key to improving chances of a successful app. All these factors could possibly have a large effect on install count. It's common to see lists of top apps, like in the top charts of the PlayStore app, but in this project, I will investigate the characteristics that contribute to an application's number of installs based on data scraped from the Google PlayStore.

2 Questions of Interest

- Do free or paid apps get more installs?
- What categories have the highest install counts?
- What relationship exists between download size and install count?
- Do higher rated apps have better install rates?

3 Data and Methods

3.1 Data

The data comes from a Kaggle dataset[1] which scraped from the Google PlayStore API. All app data was last updated on February 3rd, 2019. This dataset's information is public information and was collected to analyze app characteristics on the Google PlayStore.

The key variable of interest is `Installs`. On the Google PlayStore there is no direct install count, only an indication of 10+, 100+, etc. The range of values is so wide that even just log

[1] https://www.kaggle.com/lava18/google-play-store-apps

transformation would not adequately symmetrize the values, so I created a new categorical variable named `Install_Percentile` which grouped the apps into 4 percentiles based on their install count.

Extensive pre-processing had to be done on the original data. Several of the variables were converted to numeric like `Size` and `Rating`. For `Size` and `Price` I had to remove all extraneous characters and values for `Size` were standardized to Mbs. `Last Updated` originally gave the date of last update; however, it would be more useful to see the number of days since an update, so a new variable `Days Since Update` was created to count the difference between date of last update and the date of the web scrape, making the values comparable between apps.

Of the original predictors I removed `Genre`, `Current Ver`, and `Android Ver`. `Genre` is redundant due to `Category` and the version numbers are all relative to each app and would be difficult to implement as a categorical variable with widely varying values.

Addressing missing values, nearly all were from `Reviews` and `Size`. Because there seemed to be no discernable bias towards which Apps had missing value and because the data set is large enough, I thought it fine to perform list-wise deletion of all these elements. The final adjusted predictors are:

- `Category`: the category the app is listed under
- `Size`: size of download (in Mb)
- `Rating`: overall user rating of the app
- `Reviews`: total number of user reviews
- `Type`: indicates whether the app is free or paid)
- `Price`: price of installing the app (numeric in USD)
- `Content Rating`: age group of target audience
- `Days Since Update`: number of days since last update (from scrape date)

Considering **principles of measurement**, all the final predictors can be considered relevant as the differing values between apps is likely a point of comparison in the user's perspective. For example, a user is likely to consider a higher reviewed, cheaper, or frequently updated app.

3.2 Methods

Q1: To analyze install counts of free vs paid apps I will create a column holding log transformed install count and view a layered histogram to see the distribution of install count on free vs paid apps.

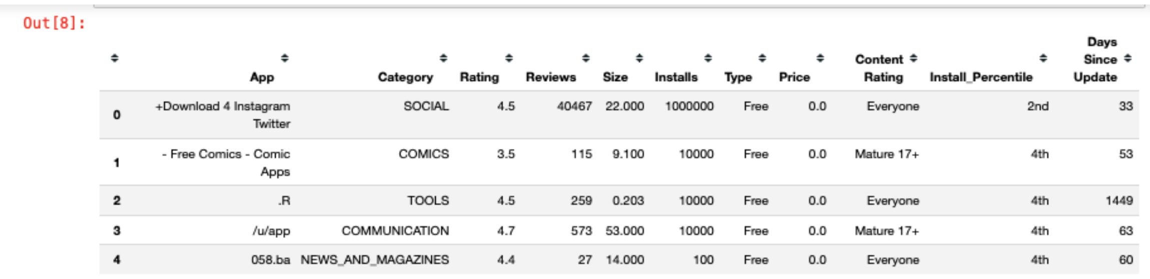
Q2: To determine what categories have the highest install counts, a bar chart of install count by category would allow me to observe any categories of interest and their distributions and trends. Since a raw count of installs by category would favor categories with more apps, it's useful to view proportion of high install counts (defined as above median) divided by total number of apps within that category.

Q3: To see the relationship between download size and install count, a box plot will show distributions of download sizes between different apps of the 4 install percentiles.

Q4: Analyzing a histogram of rating color segmented by install percentile will display the relationship rating and install count have with each other.

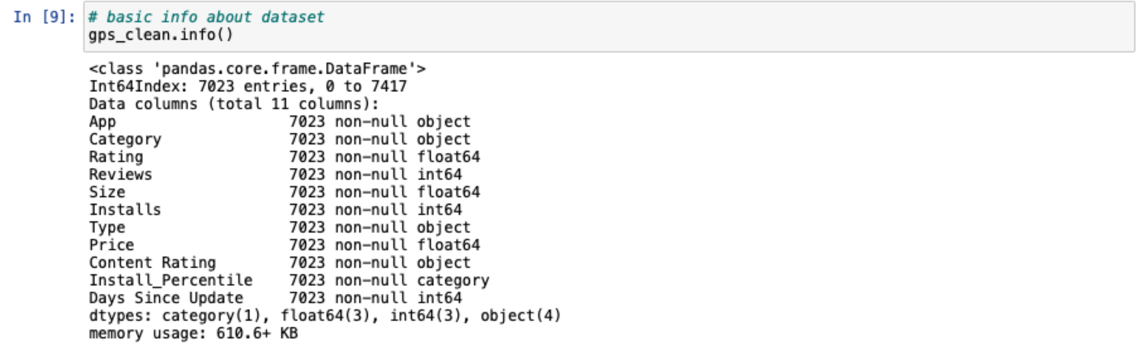
3.3 Exploratory Data Analysis

First we view basic info about the pre-processed data and a scatterplot matrix between the numeric predictors.



Out[8]:

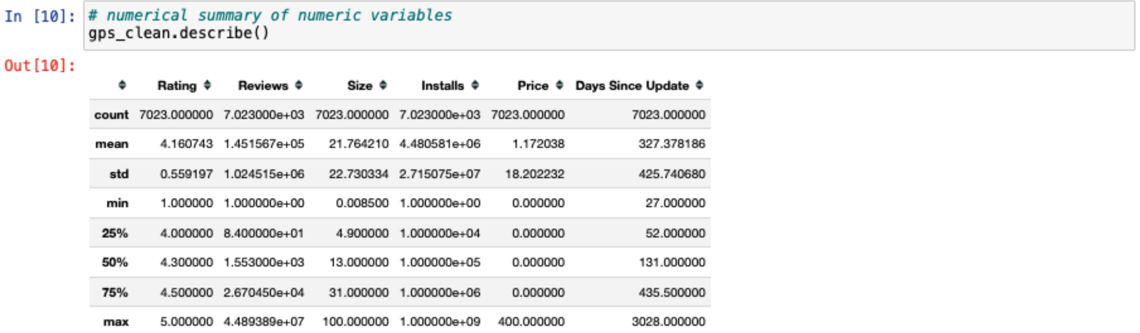
	App	Category	Rating	Reviews	Size	Installs	Type	Price	Content Rating	Install_Percentile	Days Since Update
0	+Download 4 Instagram Twitter	SOCIAL	4.5	40467	22.000	1000000	Free	0.0	Everyone	2nd	33
1	- Free Comics - Comic Apps	COMICS	3.5	115	9.100	10000	Free	0.0	Mature 17+	4th	53
2	.R	TOOLS	4.5	259	0.203	10000	Free	0.0	Everyone	4th	1449
3	/u/app	COMMUNICATION	4.7	573	53.000	10000	Free	0.0	Mature 17+	4th	63
4	058.ba	NEWS_AND_MAGAZINES	4.4	27	14.000	100	Free	0.0	Everyone	4th	60



In [9]:

```
# basic info about dataset
gps_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 7023 entries, 0 to 7417
Data columns (total 11 columns):
App                   7023 non-null object
Category              7023 non-null object
Rating                7023 non-null float64
Reviews               7023 non-null int64
Size                  7023 non-null float64
Installs              7023 non-null int64
Type                  7023 non-null object
Price                 7023 non-null float64
Content Rating        7023 non-null object
Install_Percentile    7023 non-null category
Days Since Update     7023 non-null int64
dtypes: category(1), float64(3), int64(3), object(4)
memory usage: 610.6+ KB
```

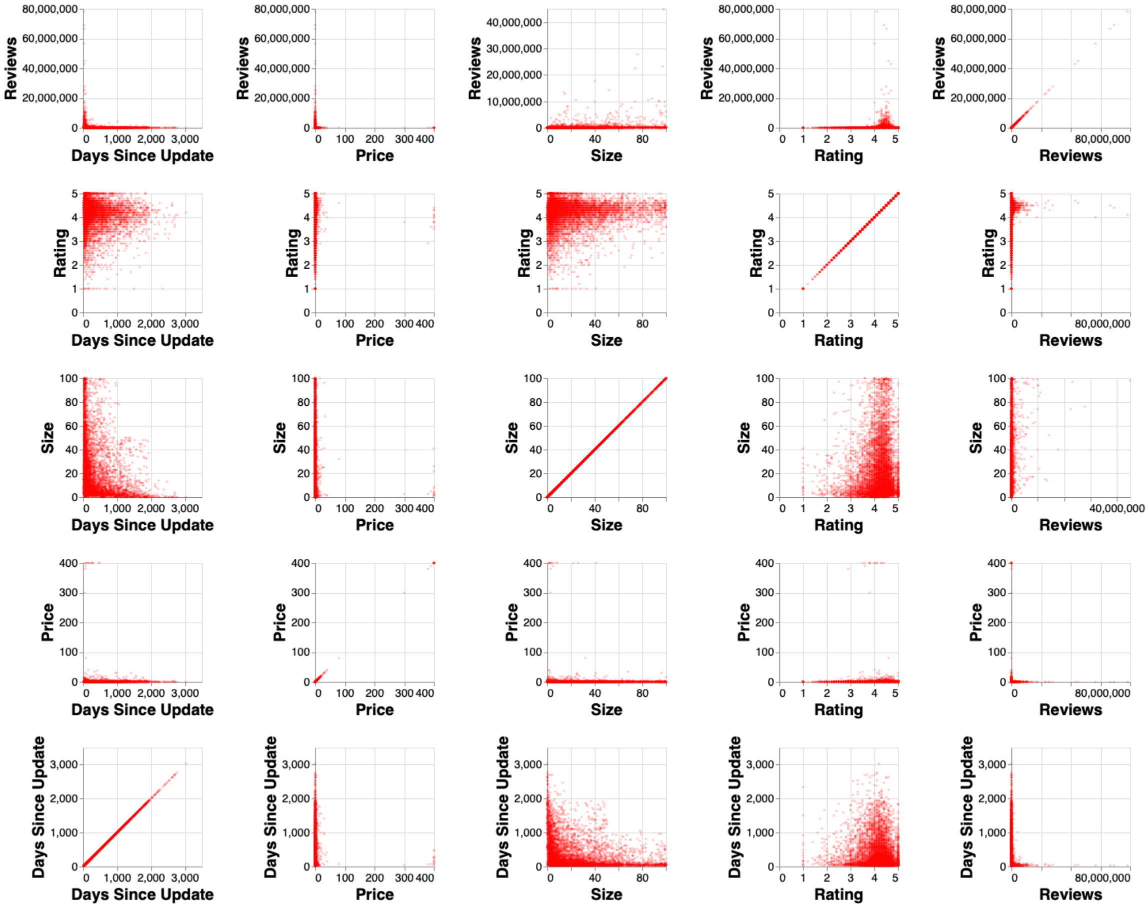


In [10]:

```
# numerical summary of numeric variables
gps_clean.describe()
```

Out[10]:

	Rating	Reviews	Size	Installs	Price	Days Since Update
count	7023.000000	7.023000e+03	7023.000000	7.023000e+03	7023.000000	7023.000000
mean	4.160743	1.451567e+05	21.764210	4.480581e+06	1.172038	327.378186
std	0.559197	1.024515e+06	22.730334	2.715075e+07	18.202232	425.740680
min	1.000000	1.000000e+00	0.008500	1.000000e+00	0.000000	27.000000
25%	4.000000	8.400000e+01	4.900000	1.000000e+04	0.000000	52.000000
50%	4.300000	1.553000e+03	13.000000	1.000000e+05	0.000000	131.000000
75%	4.500000	2.670450e+04	31.000000	1.000000e+06	0.000000	435.500000
max	5.000000	4.489389e+07	100.000000	1.000000e+09	400.000000	3028.000000



We observe larger file sizes appear more frequently in higher rated apps and are more recently updated, which is logical as larger apps likely have more content and require more frequent updates. Higher rated apps are more recently updated and have a much a larger review count than lower rated apps, although apps with a perfect rating of 5 have very few reviews. Highly priced apps seem to only be those with high ratings.

Many of the predictors have a significant number of **outliers** which will significantly distort the logistic regression model. To address this I will Winsorize the numerical predictors which transforms tail values of the variable. This does change the true distribution of data, but will allow the model to more clearly interpret the underlying patterns without interference from a few extreme points.

3.3.1 Modeling

Because the response is a categorical variable, a linear regression model would not be appropriate. Instead, a logistic regression model is more appropriate. Unlike linear regression which uses least squares to find a line of best fit, logistic regression regresses for the probability (using maximum likelihood) of a categorical outcome. It does this by using a logistic function instead of a linear equation which restricts the output between 0 and 1. I will use ordered logistic regression[2] in which the response can be non-binary and ordered.

The scoring is done using classification accuracy, or number of correctly classified responses divided by total number of observations. Additionally, I used a 5-fold cross validation[3] to find a more accurate evaluation of the model. In this procedure, the dataset was split into 5 equally sized chunks, where one is used as the test set to obtain an accuracy score while the others are used as training data. This occurs again until every chunk has been used as the test set. Then the scores are averaged to obtain a final accuracy score. I chose a 5-fold because with such a large dataset, larger values would take too much computer resources and time.

4 Results and Analysis

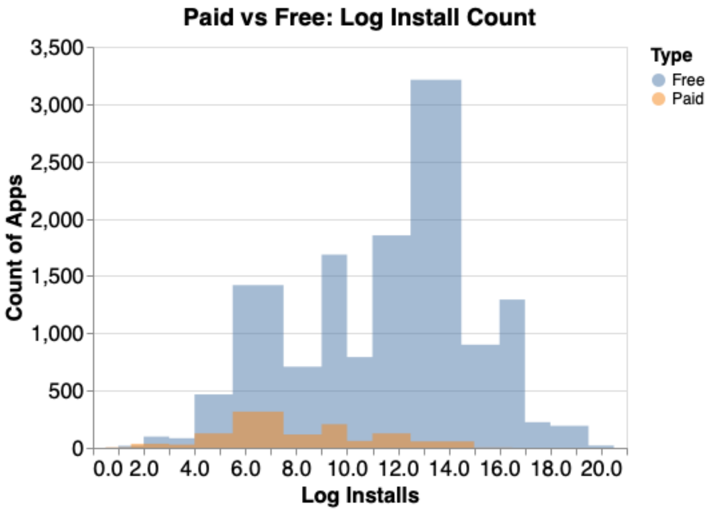


Figure 1: Shows free apps far outnumber paid apps.

In deciding whether a free or paid app is preferable for install counts, `Figure 1` makes it clear that the number of free apps far outnumber paid apps and nearly all apps with high install counts are free.

Addressing what category would be best for install count, I predicted categories like `Game`, `Productivity`, and `Family` would have the highest install counts as when I think of apps those are the categories that come to mind. From `Figure 2`, we can see the the most plentiful categories are `Family`, `Game`, and `Tools`. It looks as though by pure count the largest number of apps within the 1st percentile of install count reside in the Family category; however, if there are more apps in a category there would always be more installs. Instead, it would be more useful to see the number of high installs on a per app basis.

After dividing the number of high install counts(defined as above median) by total number of apps within that category. on a per app basis, the top 5 categories with the best install counts are

[2] implementation from mord package

[3] using sci-kit learn model$_{s}electiontools$

Entertainment, Education, Game, Photography, and Shopping. In Entertainment, approximately 4/5(Figure 5 in Appendix) apps are in the 50th percentile in download count.

For Size, I initially believed that users would prefer apps with smaller file sizes; however, Figure 3 tells us that the average file size increases as the number of installs increases. I hypothesize that this is due to larger apps having more content and thus having more to offer the user, at least up to a certain point.

Addressing Rating, Figure 4 tells us that most apps lie above a 3.0 out of 5.0 rating. An overwhelming majority of apps within the 1st quartile of install count have a rating of above 4.0. This tells to get an app with high install counts, it is almost a necessity to have a high user rating. the number of apps with a 5.0 rating that are all in the last quartile of install counts is likely due to the fact that a perfect score is nearly impossible to keep as one gains more users, so the only apps with this perfect score are those with few users and thus few reviews.

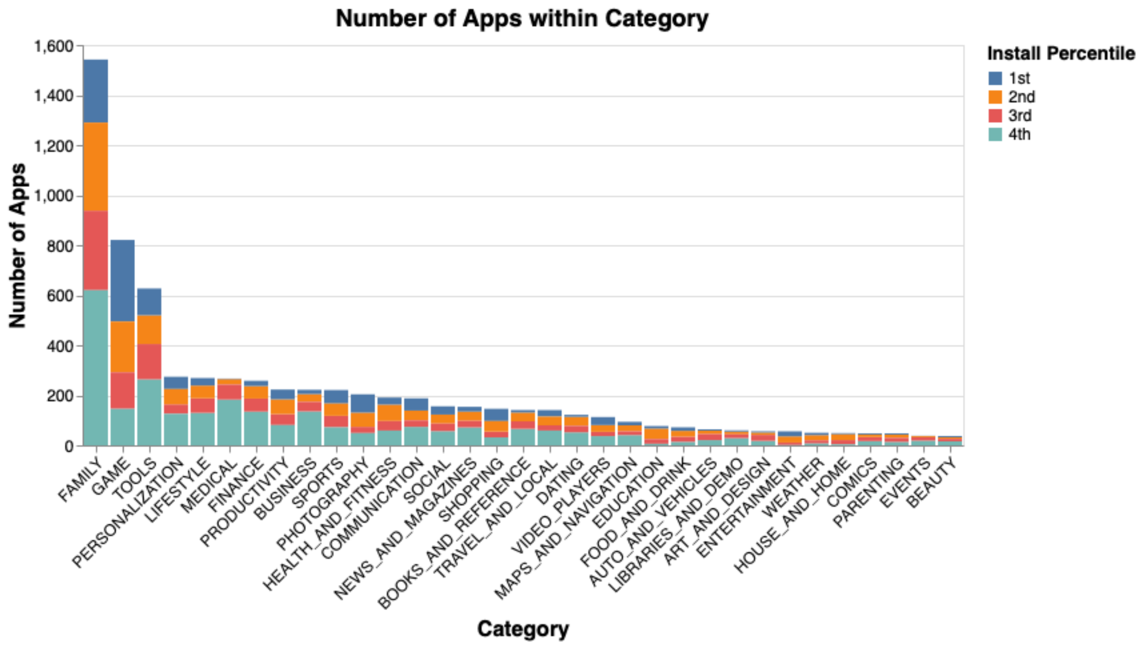


Figure 2: The number of apps within each category with color showing how many apps belong to what percentile of install count.

After numerically encoding all the variables and winsorizing the data for outliers, I fit an ordered logistic regression model using 5-fold cross validation to obtain a final accuracy of 63.35%. Considering I had not chosen an optimal probability threshhold to minimize false positive and false negative rates, it did better than expected. Given basic app from any app on the Google PlayStore, this model would accurately classify what install count quartile an app would fall under with around 65.35% accuracy.

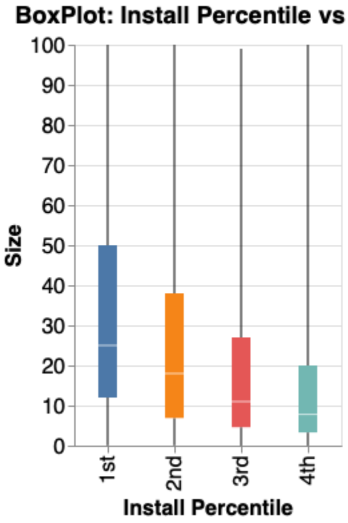


Figure 3: Shows the average file size of apps in the first quartile of install counts is larger than any other quartile.

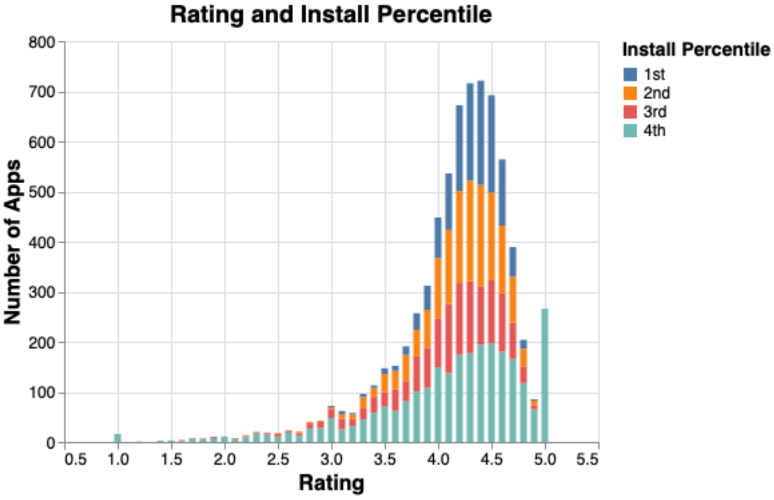


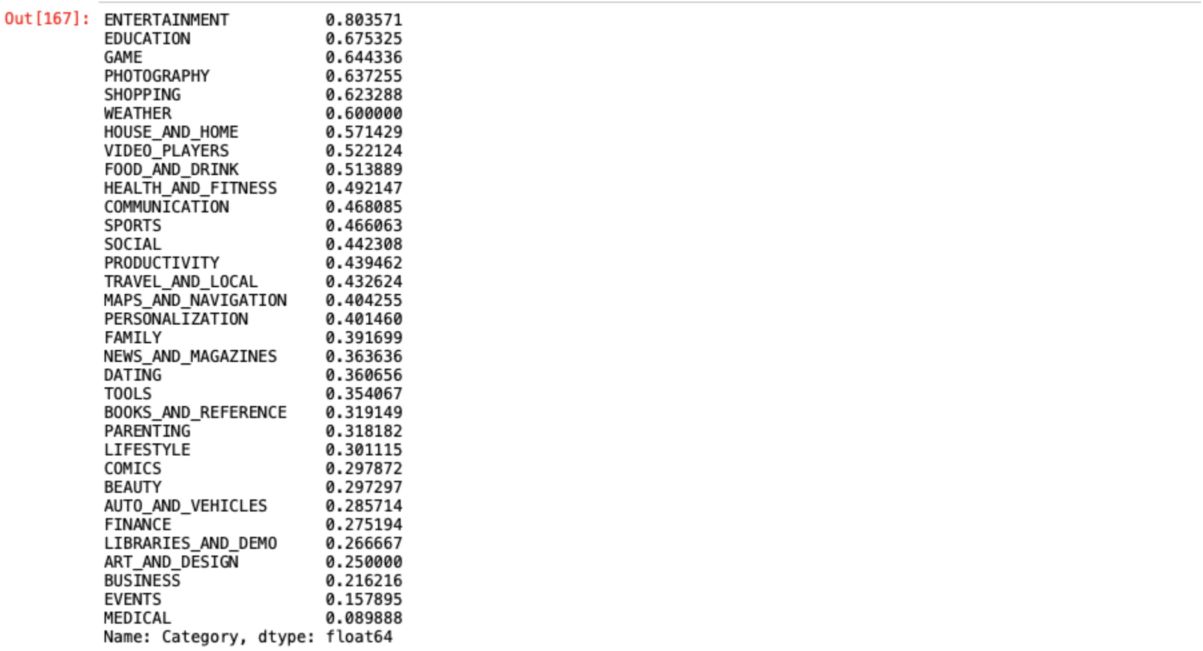
Figure 4: Shows majority of apps have above a 3.0 rating and a high rating is needed to gain high installs.

5 Conclusion

In this report, we found considerable evidence that many of the factors like category, rating, install size, and whether it is free or paid influence the number of installs. Free apps were found to be more frequently downloaded nearly all the most installed apps were free. The most common categories were general that applied to many user demographics like `Family`, `Game` and `Tools`. But on a per app basis, the categories `Entertainment`, `Game`, and `Education` had the highest install counts, so new developers can expect those to have more downloads than categories like `Medical` or `Business`, which are more niche. Additionally, highly rated apps with larger install sizes seemed to be successful in general.

I was only able to obtain a classifier with 63.35% accuracy. We can deduce that there are many factors unobservable on the PlayStore like aesthetic design of the app or the companies that produce them that affect install count. For any new developer, general advice that may lead to many downloads is to make the app free in categories that pertain to a wide audience with a focus on quality and features.

6 Appendix



```
Out[167]: ENTERTAINMENT          0.803571
          EDUCATION              0.675325
          GAME                   0.644336
          PHOTOGRAPHY            0.637255
          SHOPPING               0.623288
          WEATHER                0.600000
          HOUSE_AND_HOME         0.571429
          VIDEO_PLAYERS          0.522124
          FOOD_AND_DRINK         0.513889
          HEALTH_AND_FITNESS     0.492147
          COMMUNICATION          0.468085
          SPORTS                 0.466063
          SOCIAL                 0.442308
          PRODUCTIVITY           0.439462
          TRAVEL_AND_LOCAL       0.432624
          MAPS_AND_NAVIGATION    0.404255
          PERSONALIZATION        0.401460
          FAMILY                 0.391699
          NEWS_AND_MAGAZINES     0.363636
          DATING                 0.360656
          TOOLS                  0.354067
          BOOKS_AND_REFERENCE    0.319149
          PARENTING              0.318182
          LIFESTYLE              0.301115
          COMICS                 0.297872
          BEAUTY                 0.297297
          AUTO_AND_VEHICLES      0.285714
          FINANCE                0.275194
          LIBRARIES_AND_DEMO     0.266667
          ART_AND_DESIGN         0.250000
          BUSINESS               0.216216
          EVENTS                 0.157895
          MEDICAL                0.089888
          Name: Category, dtype: float64
```

Figure 5: Number of apps with install count above median divided by number of apps within the category.