Getting a million downloads on Google Play Store!

After 6: Angela Yoon, Ki Hyun, Sara Mehta, Tharun Raj

2020-11-17

Contents

Section 1. Introduction and data

Motivation and Relevance of Insight

In this project, we are investigating a dataset called "Google Play Store Apps." The dataset, which we retrieved from Kaggle, contains information on 9,660 unique applications (10,841 apps, including repeats) on Google Play Store obtained using web scraping (Gupta, 2019). Google Play Store is a digital distribution service operated and developed by Google, providing access to content including apps, books, magazines, music, movies, and television programs. Play Store is Google's official pre-installed app store on Android-certified devices (Google Play, 2020).

The Google Play Store is growing at an extremely fast pace, and its popularity is not unknown over the Apple App Store. In fact, the number of Google Play Store apps surpassed 1 million in as early as July 2013, and is now placed at 2.87 million (Sharma, 2020). This makes this data not only extremely interesting and relevant to look at, but also highlights its enormous potential in gauging success in the app-making businesses.

With all four of us team members being extremely interested in application development and some of us being Android users, we are interested in exploring the different factors of a given application on Play Store that can predict the app's number of downloads. As per Forbes, an average app download brings in around 2 cents to its developer on Androids (Louis, 2013). Thus, the average number of downloads for an app is a good measurement we can use to get a sense as to what an average developer can expect to make on an app today, which is why it is a very important measurement for developers that is worth looking into.

There is never a dull moment when it comes to technology, and applications continue to thrive in our technologically advancing world. However, it is difficult to study what predicts an application getting more downloads, because various elements could affect an application being downloaded. Moreover, there is immense research on techniques that can be used to get the first 100,000 downloads for an application. As per Mobisoft, a top IT outsourcing company specializing in product development, custom software development, mobile app development, and more, getting ratings and reviews as well as strategically pricing the app can all be techniques to get the first 100,000 downloads for an application, which can in turn lead to even more downloads and consequently the success of an app (Mobisoft Team, 2015). Thus, this literature has inspired our investigation to delve deeper into the relation between such application features like ratings, size, and broad category, and the applications' number of downloads. These features are easier to grapple with.

All in all, actionable insights can be drawn for developers to work on and capture the Android market based on our results. Moreover, these results will be insightful for prediction of future applications' downloads as well.

Data

There are 10841 observations in the dataset. Each row represents an application available in the United States in February 3rd, 2019. Each case also contains 13 different variables. The 13 variables and their definitions are shown below (Gupta, 2019).

Variables	Description
App	Application Name (available in the US, Feb 2019)
Category	Category of the application
Rating	Overall user rating of the app on a scale of 1 to 5 (up to Feb 2019)
Reviews	Number of user reviews (up to Feb 2019)
Size	Size of the application in Mb or Kb (in Feb 2019)
Installs	Number of user downloads/installs (up to Feb 2019)
Type	Paid or Free (in Feb 2019)
Price	Price in dollars (in Feb 2019)
Content Rating	Target age group (e.g. Children / Mature/ Adult)
Genres	Genre (multiple could be assigned to one application)
Last Updated	Date last updated on Play Store (until Feb 2019)
Current Ver	Version of the application available on Play Store (in Feb 2019)
Android Ver	Minimum Android version requirement (in Feb 2019)

The number of downloads in this dataset is recorded as a categorical data, with values of "1,000+", "5,000+", "10,000+", "50,000+", and so on. Because these values are not exact and the increments between the rounded values are not constant, we decided that converting this to a continuous variable would lead to inaccurate analysis results. Thus, we decided to separate these values into two bigger groups: "1 million +" and "less than 1 million." We chose 1 million as a cut-off based on the above mentioned Mobisoft article, as well as guides such as the one by AppInventiv that illustrate one million downloads as a prominent, industry-standard goal for one's application (Srivastav, 2018).

Research Question and Hypotheses

With the above motivations and dataset in mind, the general research question our team will be exploring will therefore be as follows: "For Google Play Store Apps, to what extent are the aspects of the application such as ratings, genre, pricing, size, and other variables in the dataset predictive of the odds of the app having more than 1 million downloads?"

We think apps in the dataset will have a set of characteristics that will be similar among them because we believe an app must be of certain quality and practicality for people to actively download it. Accordingly, our formal null hypothesis and alternative hypothesis based on this research question are as follows:

Null Hypothesis (H_0): None of the aspects of an app will be predictive of the odds that an app has 1 million+ downloads.

Alternative Hypothesis (H_a): At least one of the aspects of the app will be predictive of the odds that an app has 1 million+ downloads.

Data Cleaning

For clearer EDA and modeling, we first examined each variable to look if any cleaning was needed. Data cleaning was done through the following steps.

(Note: Only the variables that required cleaning are mentioned in the following data cleaning steps, but we made sure that each variable is as ready as possible for analysis.)

1. Duplicates: We deleted all duplicate observations in the data, leaving 9660 of the 10841 observations.

2. Missing Information: We deleted the observations that missed information on ratings or number of reviews, leaving 8196 of the 9660 observations.

3. Size: We observed that some of the information on size was given rather unclearly, entered as "Varies with device." We deleted all such observations as well, leaving 7027 of the 8196 observations. Also, size of an application is given from the sample as a categorical variable, in the format of a string, instead of numerical. We checked that it includes a unit of either M or k at the end, standing for MB or KB. Then, for all values of size that end with k, we multiplied 1024, and for all values that end with M, we multiplied 1048576, to mutate size into a numerical value counted in bytes. (Note: 1KB = 1024 bytes and 1MB = 1024KB = 1048576 bytes)

4. Number of Installs: For reasons mentioned above, we will be converting our response variable, number of installs (`Installs`) into a categorical variable of two categories, "1m +" and "less than 1m". The number of observations pertaining to each category is as following.

Installs	n
1m +	2488
less than 1m	4539

5. Content Rating: We checked that the unique values of content ratings are "Everyone," "Teen," "Everyone 10+," "Mature 17+," "Adults only 18+," ad "Unrated." We deleted all "Unrated" observations, deleting one such observation.

6. Last Updated: The values for last updated are given in the dataset as a string object. We want each last updated date to represent a numerical value for interpretation. Therefore, we mutated the value into a Date object in R.

The first two observations from our final cleaned data look like the following:

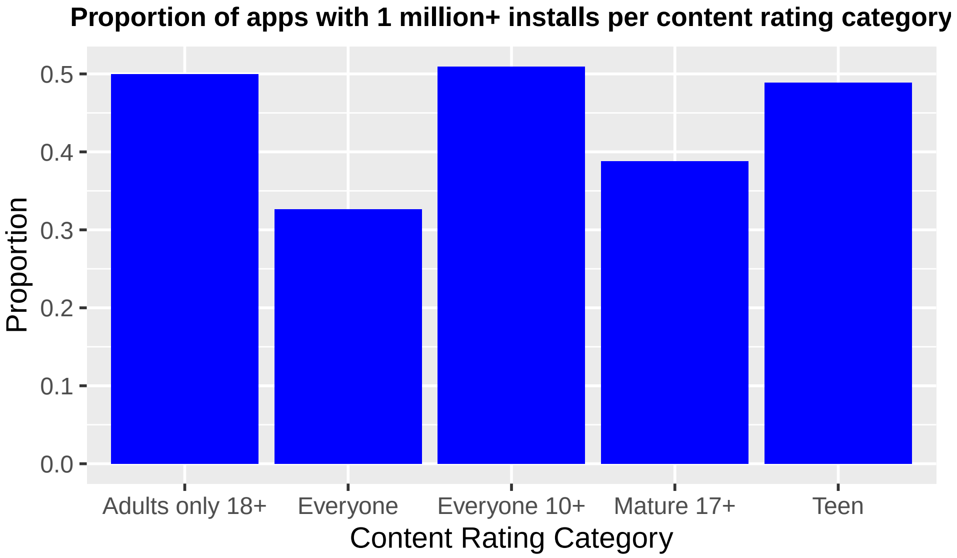
App	Photo Editor & Candy Camera & Grid & ScrapBook	Coloring book moana
Category	ART_AND_DESIGN	ART_AND_DESIGN
Rating	4.1	3.9
Reviews	159	967
Size	19922944	14680064
Installs	less than 1m	less than 1m
Type	Free	Free
Price	0	0
Content_Rating	Everyone	Everyone
Genres	Art & Design	Art & Design;Pretend Play
Last_Updated	2018-01-07	2018-01-15
Current Ver	1.0.0	2.0.0
Android Ver	4.0.3 and up	4.0.3 and up

Exploratory Data Analysis

Prior to answering our research question, we will be exploring certain summary statistics and relationships between the response variable, number of installs (`Installs`), and other variables in the data set such as ratings, content rating, payment requirement type and size to see if they will reveal any important information, patterns or problems with the data set. This will provide as a backbone for consequent exploration and inform our data analysis.

(Note: We have used our cleaned data for our EDA as our cleaning included changing variables' types and removing NA observations, both of which are necessary for EDA.)

Proportion of Applications with 1 million+ Installs per Content Rating Category

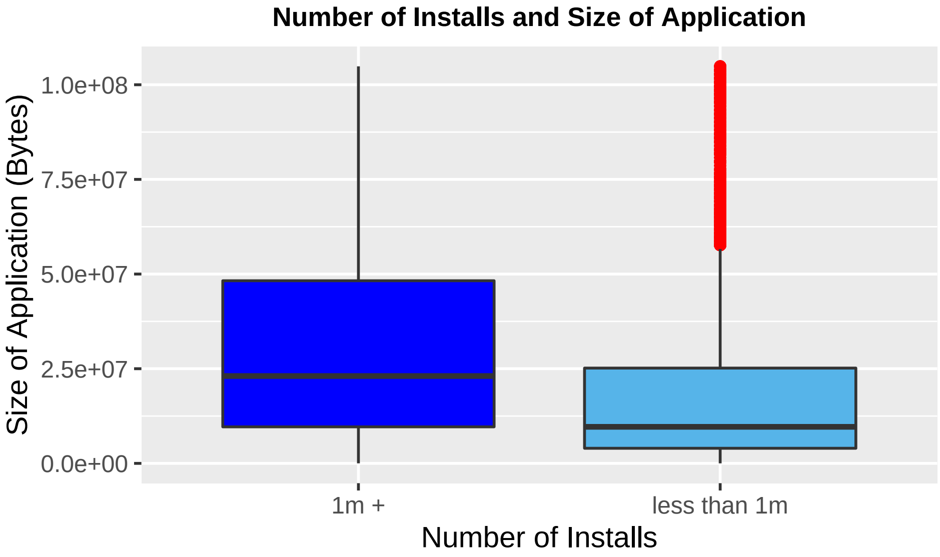


Interestingly, apps that are open to 'Everyone' have the lowest proportion of apps with 1 million+ downloads. It also seems like apps that are open to youngsters, such as in the 'Everyone 10+' category have a greater proportion of 1 million+ downloads. We also see a disparity in the number of observations in each of these categories. This suggests that there content rating may be predictive of the odds of an app getting 1 million+ downloads; however, we must interpret our results with caution owing to the skewed dataset.

Relationship between number of installs and size of the application

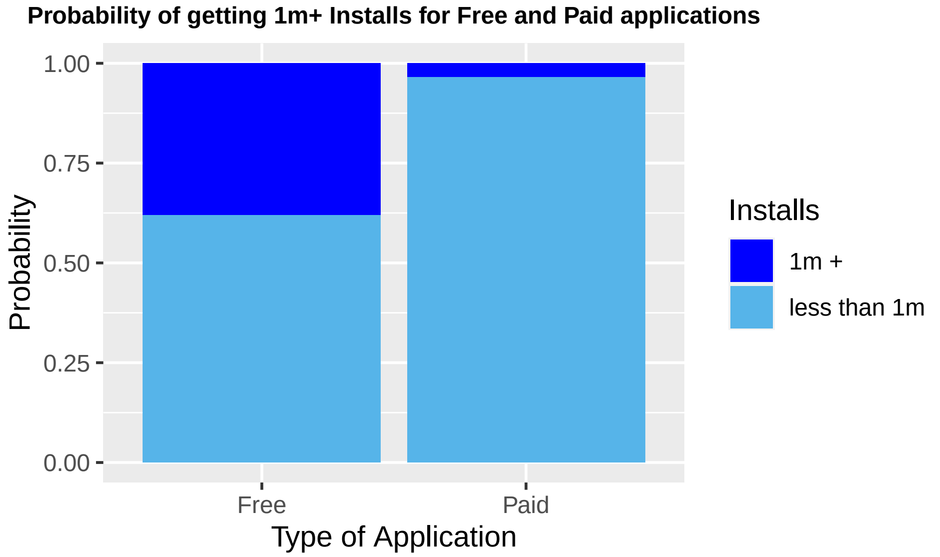
Installs	min_size	max_size	med_size	q1_size	q3_size
1m +	11264	104857600	23068672	9646899	48234496
less than 1m	8704	104857600	9646899	3984589	25165824

We will use these summary statistics to gain a more precise insight from our plot below.



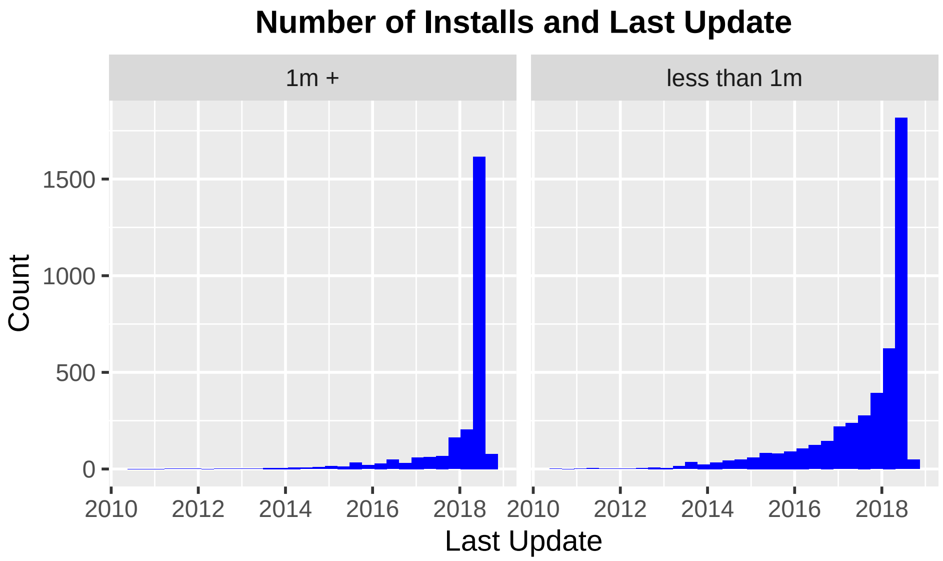
The median size for applications that have 1 million + installs is clearly greater than the median size of apps that have less than 1 million installs. We also see through the boxplots that the IQR for the size of applications with 1 million + applications is greater than the IQR of apps that have less than 1 million installs; however, there are numerous outliers, marked in red, for the size of applications with less than 1 million installs. These outliers may very well lend to size not being a significant predictor of the odds of an app getting 1 million+ downloads; this will be interesting to look out for further in our analysis.

Relationship between number of installs and type of application.



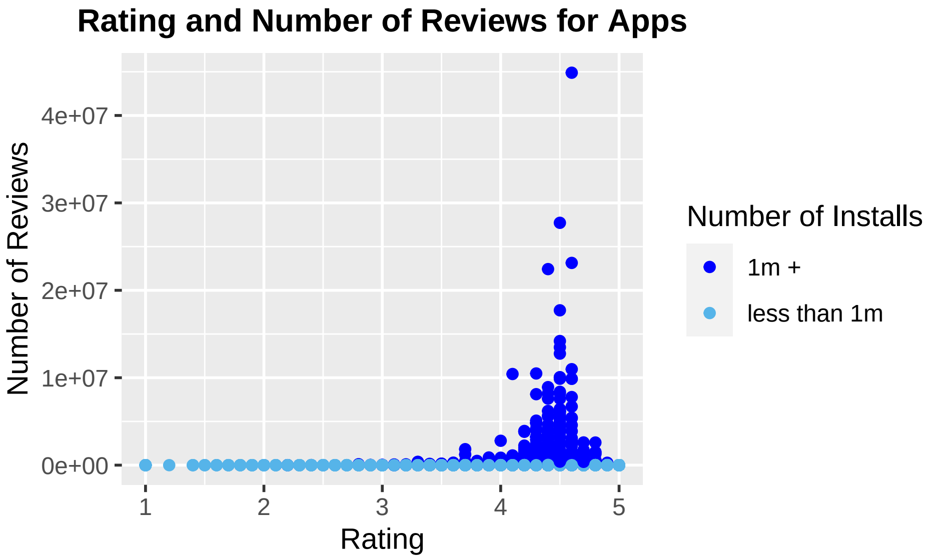
As we can see in the graph above, a much greater proportion of free applications have 1 million + installs, in comparison to the proportion of paid applications have 1 million + installs. This suggests there being an insightful relationship between the type of application and the odds of getting 1 million + installs.

Relationship between last updated data and number of installs



From the histograms above, we can gauge that majority of apps with 1 million + installs tend to be the ones updated around early 2019. We can also see that a great number of applications with less than 1 million installs had been updated in the past year. These relationships will be explored further in our model selection.

Relationship between Reviews, Ratings and Number of Installs



From the graph above, we can clearly see that apps that have 1 million + installs tend to have higher ratings, particularly above 3. At the same time, we can also see that such apps tend to have a greater number of reviews than apps that have less than 1 million installs. Moreover, in general, apps that tend to have a higher rating tend to have a greater number of reviews too. This suggests that there may be an interaction type of relationship between the rating and review variables.

Section 2. Methodology

Modeling Process

We will use the logistic regression model to fit the data as we are looking at the factors that relate to the *odds* of an app getting more than a million installs; our response variable is a binary, categorical variable.

Moreover, since we are trying predict the odds that an app has more than a million installs given the features associated with an app, our modeling objective would be prediction. The goal of a predictive modeling objective is identifying variables that are significant in predicting the odds that an app has more than a million installs. Therefore, the logistic regression model is most aligned with the purpose of our study.

To determine the best fit model, we will use the BIC backward selection. Since our goal is prediction, we want to get down to the smallest number of variables that are strong predictors of our response and that minimizes prediction widths; thus, we will use BIC. BIC tends to favor more parsimonious models as it has a harsher penalty for having more variables. We would use backward selection instead of forward selection since we want to include all possible predictors at the start and reduce it to the set of variables that are most significant; we only want to get down to those variables that have a significant effect in explaining the prediction of whether an app has 1 million+ installs.

Variables

Additional Variable Modifications

1. Number of installs

The number of installs(`Installs`) in the dataset is currently just a categorical variable that either indicates "1m +" or "less than 1m". In order use it as a response variable for the logistic model, we would need to transform all the "1m +" to a numeric value 1, and all the "less than 1m" to a numeric value 0.

2. Days from last updated

The last updated date, in the form of a Date R object, is unfit to be considered as a quantitative predictor for a logistic linear regression. Rather, it would be useful to use the number of days between the last date of update and the date when the data was scraped. It is mentioned in the data description that the dataset was scraped in February 3rd, 2019. Therefore, we again transform the last updated date into a new variable(`Days_from`) that is the number of days since each app was last updated to the date the data was scraped (Feb 3rd, 2019).

Variables of Interest

Our main variables of interest are ratings, number of reviews, and days from last update as we have seen their relationships with the response in EDA. Ratings may be a significant predictor because an app with more than 1 million installs is probably highly appreciated by more people and is thus likelier to have a high rating. Number of reviews may be a significant predictor as well because the number of reviews would be highly associated with the number of installs. If a lot of people wrote reviews, it is more likely that the app was downloaded a lot of times in the first place. Days from last update may be a significant predictor as recently updated apps may have the latest features that would attract users. Moreover, a constantly used, popular app would call for the necessity of its developers to update the app according to user feedback. We also see an increase in advertisement and consequent attraction when it comes to app updates.

Besides these variables for which we can easily identify their potential significance in prediction, we also added category, size, type of whether the app requires payment, and content rating of the apps to the full model as variables worth exploring. Other variables were omitted as they are either redundant of the variables already added (variable for the price of the app is redundant with the type of payment requirements, and the genre of an app is redundant with category) or have little possibility of being analyzable with respect to the response variable (as in current version of the app or the Android version it supports).

Interaction Terms being Explored

We will also try and add three interaction terms based on what we think might affect the number of installs. The first is the interaction term between **ratings and number of reviews**. The value of a rating differs greatly by the number of people who participated in rating. For example, if only 5 people gave a rating of 5, this value would be less valuable in accounting for the actual ratings for the app for the whole population when compared to 500 people giving a rating of 5. Since we are under the assumption that apps that are viewed as "good" by the public are more likely to get more than a million installs, an interactive term is needed to control for the disproportionate effect of ratings based on the number of people who participated in the rating. Therefore, the interaction term should be introduced, with the number of reviews being the proxy for the number of people who actually participated in giving a rating for the app.

Another interaction term that we would like to explore is between **category and size**. Generally, we would expect some apps from specific categories to have a large size, say gaming apps compared to simple utility apps such as third-party file managers. This might change the effect size would have on the tendency of a user to download an app, depending on the expectation on the size the user might have for different app categories. Therefore, this could correlate to a different effect on the odds that an app reaches more than a million installs.

The last interaction term that we would like to explore is between **category and days from last update**. Apps of certain categories might not require as frequent updates than others, while remaining popular (and having a higher chance of having more than a million installs). For example, a very popular cooking recipe would require less frequent updates than an unpopular gaming app. In other words, although the cooking recipe app could have been updated longer ago, it could still have a higher odds of getting more than a million downloads. However, when controlled for the different categories, the more recently an app was updated, the more likely it is used consistently by the users, and thus the more likely it is to have higher odds of getting more than a million installs.

Model Selection

We therefore use the full model with predictor variables rating, number of reviews, days from last update, category, size, payment requirement type, content rating, and the interaction terms between reviews and number of rating; category and size; category and days from last update. As previously explained, backward selection with BIC will be used for our model selection.

The model that we obtain from this selection is (Model selection trace **here**)

$$\hat{Installs} \sim Rating + Reviews + Type + ContentRating + DaysFrom + Rating \times Reviews$$

We now examine the model assumptions and diagnostics to see if the model can be further improved.

Model Assumptions

1. Linearity: Initially not satisfied; requires transformation

To check for this assumption, we made empirical logit plots using our untransformed numerical predictor variables and our response variable, which can be found **here**.

The linearity condition is not met. As shown clearly in the empirical logit plot, all three predictors, ratings, number of reviews and days from update do not seem to show a linear relationship with the response variable (`Installs`). This is a critical violation of an essential assumption for the logit model. The seemingly hyperbolic relationship all the predictors have should be factored in to the model for a valid interpretation of the model.

The empirical logit plot revealed that ratings has a quadratic relationship with the response variable, while the number of reviews and days from update have a logarithmic relationship. To adjust for the linearity condition, a quadratic term for ratings ($Rating^2$) and log transformed terms for the number of reviews and days from update will be introduced in the final model.

2. Independence: Satisfied

Independence is most often violated if the data were collected over time or there is a strong spatial relationship between the observations. In this case, the data was scraped at once, and there cannot be a significant spatial relationship between the observations. Therefore, it is reasonable to conclude that the installs of each app is independent of one another, and the independence condition is satisfied.

3. Randomness: Satisfied

Whereas there is no specific indication as to how the apps in the data were selected among the many apps on Google Play Store, there is little reason to believe that the observations in the sample (apps in the dataset) differ systematically from the population of interest (all apps in Google Play Store). Therefore, the randomness condition is satisfied.

Model Diagnostics

Detecting Multicollinearity

We have tested for multicollinearity, which can occur when there are very high correlations among two or more predictor variables. We want precision in our regression coefficients' estimations; multicollinearity between our final selected variables can impede our ability to use our model for inference and prediction.

names	x
Rating	1.323912
Reviews	250.133696
TypePaid	1.189668
Content_RatingEveryone	481.998883
Content_RatingEveryone 10+	98.932213
Content_RatingMature 17+	147.337128
Content_RatingTeen	276.785463
Days_from	1.029065
Rating:Reviews	252.165719

Above is the Variance Inflation Factor (VIF) for our model, which is a measure of multicollinearity in the regression model.

Given that the VIF of the number of reviews (`Reviews`) is very high and similar to the VIF of the interaction term between ratings and number of reviews (`Rating * Reviews`), we are removing the interaction term from the model to account for this multicollinearity. We can see that the VIF scores for some of the Content Rating categories is high (> 10); however, since they do not have similar values, we are keeping them in the model.

Final Model Output

Final Model

term	estimate	std.error	statistic	p.value
(Intercept)	-35.947	4.138	-8.688	0.000
Rating	10.159	1.920	5.292	0.000
log_reviews	2.201	0.074	29.653	0.000
TypePaid	-3.554	0.385	-9.228	0.000
Content_RatingEveryone	2.837	1.603	1.769	0.077
Content_RatingEveryone 10+	1.657	1.626	1.019	0.308
Content_RatingMature 17+	1.732	1.623	1.067	0.286
Content_RatingTeen	1.674	1.609	1.040	0.298
log_daysfrom	-0.347	0.096	-3.622	0.000
I(Rating^2)	-1.518	0.243	-6.240	0.000

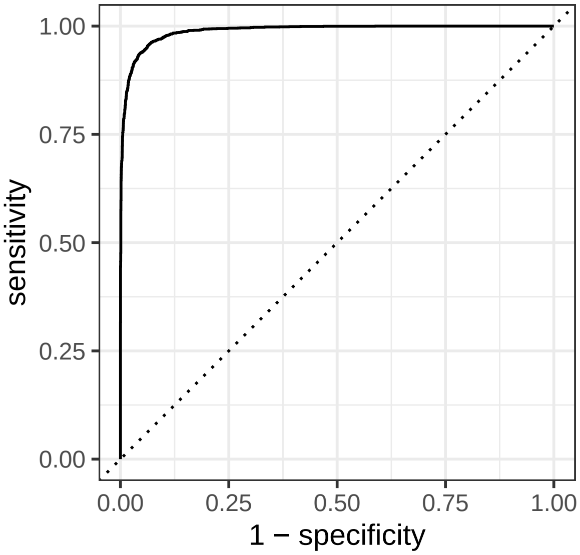
As can be seen above, our final model is as follows:

$$\begin{aligned}\hat{Installs} = -35.95 + 10.16 \times Rating + 2.20 \times \log(Reviews) - 3.55 \times Type\_Paid \\ +2.84 \times (ContentRating\_Everyone) + 1.66 \times (ContentRating\_Everyone10+) \\ +1.73 \times (ContentRating\_Mature17+) + 1.67 \times (ContentRating\_Teen) \\ -0.35 \times \log(DaysFrom) - 1.52 \times Rating^2\end{aligned}$$

Section 3. Results

Model Fit Statistics & Assessing Model's Predictive Power

We will assess how well the logistic model fits the data by generating a Receiver Operating Characteristic (ROC) curve and finding the area under curve (AUC) of the ROC curve. Below is the ROC curve of our final model.



We get an AUC of 0.9889. Because an AUC close to 1 indicates a good fit, we can say that our model fits the data very well.

From the ROC curve and by trial and error, we find out that 0.5 is the best decision-making threshold. Below is the confusion matrix with a threshold of 0.5.

Installs	pred_installs	n
1	1m+	2313
1	less than 1m	175
0	1m+	183
0	less than 1m	4355

The misclassification rate when using this threshold is 0.05. With this low misclassification rate, we again confirm that the model is a good fit of the data and has high predictive power; in other words, it can be used for making useful predictions.

Coefficient Interpretations

We interpreted some of the most significant coefficients of variables that seem to be predictive of the odds that an app has more than a million downloads.

If an app has double the number of reviews, the odds that the app has more than a million downloads is expected to multiply by a factor of $2^{2.2007}$, 4.597, keeping the ratings, content rating, app type and days from last update constant. If an app has more reviews, that may generally mean that it would've had more installs in the first place.

If an app's rating increases from 1 to 2, the odds that an app has more than a million downloads is multiplied a factor of $e^{-1.5182(2^2-1^2)+10.1587(2-1)}$, 271.537, keeping the number of reviews, content rating, app type and days from last update constant. Thus, if one's priority is to get more installs, then focusing on getting a higher app rating may be very beneficial.

The odds that a paid app has more than a million downloads is $e^{-3.5542}$, 0.029 times the odds that a freely available app has more a million downloads, keeping the number of reviews, content rating, ratings and days from last update constant. This number is small as all the paid apps tended to be around 0.99 dollars. Notwithstanding, this essentially makes sense since a paid app may not be accessible or affordable to as many people than a free app.

Another interesting point to note is that the coefficients for the content rating are all positive which means the odds that an app has more than a million downloads is higher than that of the baseline which are apps with the content rating of adults only 18+. This seems to be in line with what one would expect since apps meant for a larger audience would in theory have a higher tendency to obtain more installs than apps that are meant for a subset of the audience i.e. adults only. Thus, if one is aiming to get more installs on their app, then having it open to all age groups would most likely be most beneficial.

Section 4. Discussion

Summary of findings

Our research question was "For Google Play Store Apps, to what extent are the aspects of the application such as ratings, genre, pricing, size, and other variables in the dataset predictive of the odds of the app having more than 1 million downloads?" Installs is a categorical variable we have used to capture this distinction of an app having more than 1 million downloads. From our EDA itself, we noticed that certain aspects, such as the type of the app (whether it is paid or free) and the days since last updated seemed to have a relationship with the number of installs. We also found that the relationship between reviews and ratings was different for apps with 1 million + installs and apps with less than 1 million installs.

Overall, our calculation of p-values from our logit model revealed that our slopes for Ratings, Reviews, Days_From, and Type were below the alpha level of 0.05. In particular, we see that the odds of having greater than 1 million installs for apps on the Google Play Store is greater with higher ratings, a greater number of reviews, a recent update, and if the app is free. This highlights the statistical significance of our findings, and gives us enough evidence to reject the null hypothesis that none of the aspects of an app will have a relationship with the odds that it has 1 million+ downloads. It appears as though there are differences in the value of the variables listed below between apps with 1 million+ downloads and apps with less than 1 million+ downloads. An interesting finding was that effect of the combined action of the two variables Reviews and Ratings as captured in the interaction term is less than the sum of the individual effects, as can be seen from the negative coefficient of the interaction term. This has been captured in our EDA as well. Due to multicollinearity, we eliminated this term from our model; however, it would be interesting if we could further examine the content of reviews using sentiment analysis to understand this relationship better.

These findings are immensely insightful for those who endeavor to attain more than 1 million + downloads on apps they have developed and released on the Google Play Store, as the factors found to predict the odds of an app getting 1 million + downloads may be extremely helpful to focus on to attain this goal. Particularly, one can use strategies to increase number of reviews, have a higher app rating, keep the app as free to download and update the app with recent features.

Critique of method and analyses

1. Construct Validity

The initial dataset was scraped directly from the actual Google Play Store. Therefore, the raw data collected and used for interpretation is precisely the values we wanted to analyze. Such high data matching leads to stronger construct validity. Moreover, any mutation done on the variables does not rely on any inference from the raw data, but only simple algebraic transformations. Moreover, the variables that did not fit the linear condition were mutated for a logistic linear regression.

2. Internal Validity

The backward model selection model ensures that all the variables included in the full model has been accounted for before selecting the final model. Therefore, there is less chance of Omitted Variable Bias (OVB) present in the final model since all the variables and three interactive terms had been accounted for. However, there are still many factors that could influence the odds of an app getting more than a million installs that is not included in the dataset. Moreover, not all possible interactive terms from the data had been accounted for, but only the 3 we deemed important. There is still a possibility of OVB being present in the interpretation. Thus, the internal validity seems intact but not guaranteed.

3. External Validity

The evaluation of the randomness condition in the model assumptions concluded we have little reason to believe that the collected sample differs systematically from the actual population of Google Play Store apps. However, this assumption relies entirely on the fact that the scraping method used by the author of the original dataset incorporates appropriate random sampling of the population. Moreover, from the fact that not a single widely known app such as Facebook, Instagram, Twitter is included in the dataset, we may conclude that the external validity is not guaranteed.

4. Reliability

Reliability of the data is not in question since all observations will produce exactly the same results every time it is scraped, as long as the date the data was scraped does not change. Again, not a single widely known app such as Facebook, Instagram, Twitter is included in the dataset. This may indicate that there are significant missing data from the actual Google Play market. Therefore, the result might not be reliable if we scrape a completely different set of apps. However, within the data provided, the backward step selection process made all the most predictive variables get accounted for. Therefore, reliability is breached but contained within the provided dataset.

Scope for improvement

There is plenty of room for improvement regarding the dataset. This report has utilized a sample dataset given from another author, who scraped the set from the Google Play market. However, the scraped data seems to display some fundamental flaws. First and foremost, many observations had to be disregarded due to missing parts of data. Although there is no particular reason to believe that observations with missing parts of the data have systematic differences with the rest, the study's model could have made more use of the sample if there were none. Moreover, the missing data could have been filled in based on observations in the data set.

Moreover, we have reason to believe that the sample might not be entirely representative of the Google Play apps population. As discussed during the assessment of external validity, the sample used for modeling is not ideal. Though there might not be enough evidence to say that the sample exhibits significant systematic difference from the population, even just the simple absence of all universal social networking service apps from the sample may hint that our sample is not a perfect IID. Albeit perfect IID samples are near an impossibility in real practice, there is still room for improvement in our sample being more representative of the Google Play market. Lastly, it would be interesting to use sentiment analysis to look at reviews in particular, to better understand whether it is merely the number of reviews or the actual positivity associated with reviews that is predictive of the odds of an application getting 1 million+ downloads on the Google Play Store.

References

Google Play, Retrieved from https://play.google.com/store

Sharma S (2020 May 6). Top Google Play Store Statistics 2020-21. *Appventurez.* https://www.appventurez.com/blog/google-play-store-statistics/

Louis T (2013 Aug 10). How Much Do Average Apps Make?. *Forbes.* https://www.forbes.com/sites/tristanlouis/2013/08/10/how-much-do-average-apps-make/#7c1bfb4446c4

Mobisoft Team (2015 July 14). Top 34 Techniques to Get First 100,000 Downloads For Your App. *Mobisoft.* https://mobisoftinfotech.com/resources/blog/top-34-techniques-to-get-first-100000-downloads-for-your-app/

Gupta L (2019 Feb 03) (Retrieved 2020 Oct 21). "Google Play Store Apps". *Kaggle* https://www.kaggle.com/lava18/google-play-store-apps

Srivastav S (2018 March 23). How to Get Million Downloads On Your App. *appinventive.* https://appinventiv.com/blog/get-million-downloads-app/

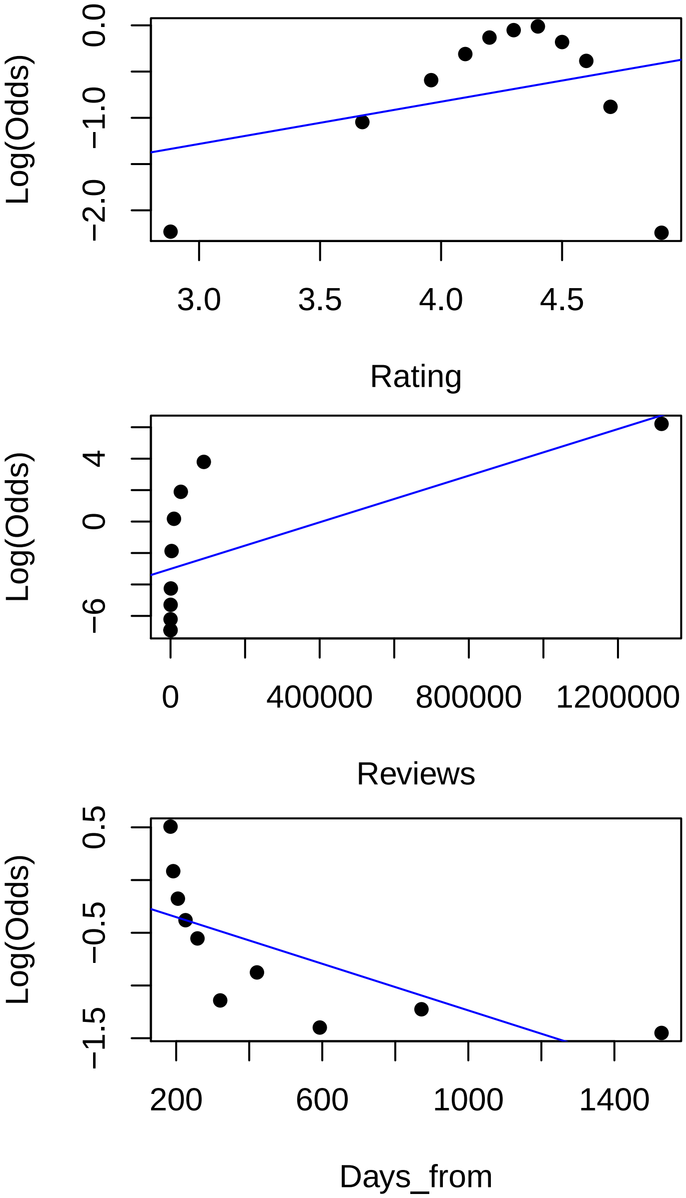
Appendix

Model Selection Process

```
## Start:  AIC=3099.98
## Installs ~ Category + Rating + Reviews + Size + Type + Content_Rating +
##     Days_from + Rating * Reviews + Category * Size + Category *
##     Days_from
##
##                       Df Deviance    AIC
## - Category:Size      32   2188.5 2852.8
## - Category:Days_from 32   2193.4 2857.7
## - Content_Rating      4   2183.1 3095.4
## <none>                    2152.2 3100.0
## - Type                1   2313.8 3252.7
## - Rating:Reviews      1   2505.5 3444.4
##
## Step:  AIC=2852.83
## Installs ~ Category + Rating + Reviews + Size + Type + Content_Rating +
##     Days_from + Rating:Reviews + Category:Days_from
##
##                       Df Deviance    AIC
## - Category:Days_from 32   2234.4 2615.3
## - Size                1   2188.9 2844.3
## - Content_Rating      4   2220.7 2849.6
## <none>                    2188.5 2852.8
## - Type                1   2350.6 3006.0
## - Rating:Reviews      1   2542.9 3198.4
##
## Step:  AIC=2615.26
## Installs ~ Category + Rating + Reviews + Size + Type + Content_Rating +
##     Days_from + Rating:Reviews
##
##                   Df Deviance    AIC
## - Category       32   2327.4 2424.8
## - Size            1   2235.1 2607.1
## - Content_Rating  4   2266.5 2611.9
## <none>               2234.4 2615.3
## - Days_from       1   2264.3 2636.4
## - Type            1   2406.2 2778.2
## - Rating:Reviews  1   2587.8 2959.8
##
## Step:  AIC=2424.85
## Installs ~ Rating + Reviews + Size + Type + Content_Rating +
##     Days_from + Rating:Reviews
##
##                  Df Deviance    AIC
## - Size            1   2327.4 2416.0
## <none>               2327.4 2424.8
## - Content_Rating  4   2378.2 2440.2
## - Days_from       1   2353.8 2442.4
## - Type            1   2497.5 2586.1
## - Rating:Reviews  1   2695.0 2783.6
##
## Step:  AIC=2415.99
## Installs ~ Rating + Reviews + Type + Content_Rating + Days_from +
##     Rating:Reviews
##
##                  Df Deviance    AIC
## <none>               2327.4 2416.0
## - Content_Rating  4   2379.9 2433.0
## - Days_from       1   2354.8 2434.5
## - Type            1   2497.6 2577.3
## - Rating:Reviews  1   2695.4 2775.1
```

```
##
## Call:  glm(formula = Installs ~ Rating + Reviews + Type + Content_Rating +
##     Days_from + Rating:Reviews, family = "binomial", data = apps_data)
##
## Coefficients:
##               (Intercept)                     Rating
##                -9.1319257                  0.3983152
##                   Reviews                   TypePaid
##                 0.0019780                 -7.4669634
##     Content_RatingEveryone  Content_RatingEveryone 10+
##                 4.6659561                  3.5433621
##   Content_RatingMature 17+         Content_RatingTeen
##                 3.6544807                  3.4181164
##                 Days_from             Rating:Reviews
##                -0.0008265                 -0.0003918
##
## Degrees of Freedom: 7025 Total (i.e. Null);  7016 Residual
## Null Deviance:       9133
## Residual Deviance: 2327  AIC: 2347
```

Empirical Logit Plots


Log(Odds)
Rating
Log(Odds)
Reviews
Log(Odds)
Days_from