What Determines an App’s Rating?

STAT 3220, Section 2

Kieran Cottrell, Nicolas Izurrategui, Alex Johnson, and Max Jones

Spring 2020

Introduction

We plan to create a model to predict an app's rating using the Mobile Apps Store dataset. Over the past decade, the usage of mobile apps has boomed--for example, according to *Statista*, mobile phones generated 52.2 percent of all website traffic worldwide in 2018. As a result of this growth, the profitability of apps has also increased significantly. Just as of April 2019, global app revenue reached $19.5 billion, an increase of 17% from April 2018 (Nelson). It is clear that app creation has the potential to be a highly lucrative venture, and one of the factors playing into whether an app becomes successful is its overall user rating. In fact, user ratings are often the first thing new customers look at to determine if an app is worth installing. Because of the importance of the user rating, we decided to look at exactly *what* leads to higher-rated apps. Perhaps certain app genres naturally have higher ratings, or maybe customers tend to prefer smaller apps.

We believe that figuring out what gives apps higher user ratings could be of great importance for anyone looking to make an app. For large companies or small developers looking to create successful apps, our findings could be greatly useful. Even developers just looking to make useful apps with no intention of making money could use our findings to better satisfy app users.

In order to respond to our research question, we decided to implement a multiple linear regression model. That is, we will model the linear relationship between our categorical and continuous explanatory variables, and our response variable. We have decided to utilize this technique for two reasons. Firstly, modeling a multiple linear regression will provide us with an effective interpretation of the relative influence of individual factors on our response variable. By looking at the strength of each of our variables we can see which of these factors is more responsible for the variability in our response variable. Secondly, according to Will Kenton, an author at Investopedia, using a multiple linear regression model is the most popular statistical technique used by researchers and companies because of how these models can be easily extended to predict outcomes on new data sets. This means that if we were to utilize this model on a new data set to predict the average rating of a new app, we would be able do so with no major adjustments to our model.

Although we find several advantages for our chosen technique, there are also disadvantages for utilizing a multiple linear regression model. The first of these disadvantages is that by utilizing a multiple linear regression technique, we are assuming that the underlying relationship between our variables is strictly linear. Sciencing Publishers says that this assumption may lead to an oversimplification of the relationship between our factors. That is, utilizing this technique would limit the model to only account for simple linear relationships and not of any other kind. Another major disadvantage of multiple linear regression is the high number of assumptions required before making inferences about the data. Because we are utilizing this technique, we will have to check and correct our analysis for the violation of common assumptions, such as the assumptions of the equal variance, independence, non-correlation and normality of errors.

Data Description

Our dataset comes from the Google Play Store, obtained from Kaggle at **https://www.kaggle.com/gauthamp10/google-playstore-apps**. The dataset contains data for over 267,000 apps from the Google Play Store from April 2019. The data was extracted from the Google Play Store using Python web scraping tools. Since the data comes directly from Google, we can be reasonably sure of its validity. The data contained both a sample of 267,000 apps and a sample of 32,000, so for simplicity we used the sample of 32,000 apps for our dataset. We also removed 3 observations from the dataset because those observations got corrupted. Additionally, we removed several variables because they were either unchangeable or did not contribute to an app's rating (such as the app's version number or when the app was last updated). We would have liked to have included the size of the app, but unfortunately, some of the values were missing (varied depending on device), so they were unusable. Additionally, the number of installations for each app contained numerous categories, so for our data analysis, we concatenated these into three categories: under one million installs, one million to ten million installs, and over ten million installs. We also had to change the number of genres, grouping them from their original very specific 49 genres to more broad groups (totaling 4) to make the analysis more manageable. Finally, to simplify the number of levels for the content rating variable, we took a sample of only apps rated "Everyone" and "Teen", thus making the data more manageable. Our group is interested in seeing what variables cause the greatest influence on an app having a higher average user rating in the app store so we can create a model to predict an app's average user rating. Each observation in the data set represents a single app with its various predictor variables. The variables are outlined and explained in the table below:

Data Dictionary

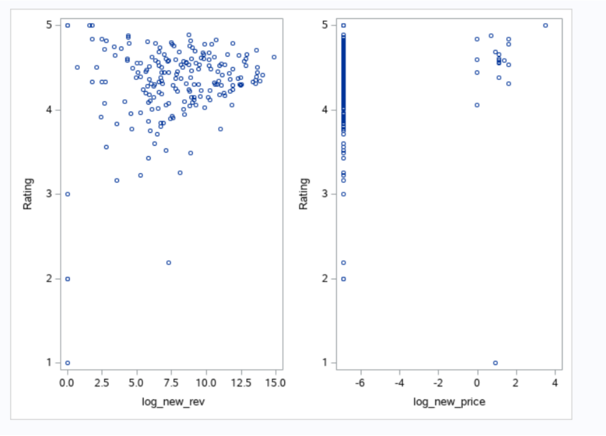
Variable	**Description**	**Quantitative/Categorical**
Rating	Average user rating of the app (0 to 5 stars)	Quantitative
Reviews	Total number of user reviews for the app	Quantitative
Installs	Number of installations for the app (under one million installs, one million to ten million installs, and over ten million installs) To represent installs we will have 2 dummy variables with under one million installs as the base level.	Categorical

Price	Price of the app (in dollars)	Quantitative
Category/Genre	Primary genre of the app, ranging between these groups: entertainment, lifestyle, media, and productivity To represent genre we will have 3 dummy variables with entertainment as the base level	Categorical
Content Rating	Content rating (age recommendation) for the app, ranging between Everyone and Teen To represent Content Rating we will use 1 dummy variable with the everyone rating as the base level.	Categorical

Early Data Analysis

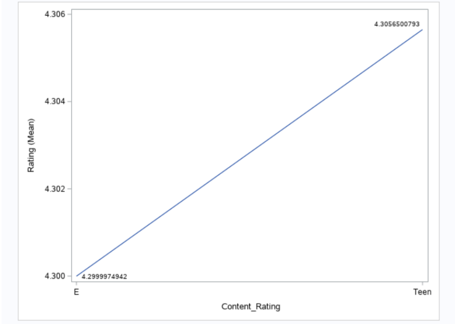
While performing our early data analysis, we noticed that our quantitative data presented a lot of 0s and was skewed, so we performed a log transformation to make the quantitative variables more linear. After that, we then looked at the quantitative and categorical variables relationship with our response variable, rating, to check if effects seemed present.

Quantitative Variables Relationship

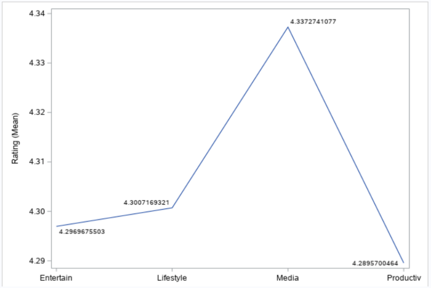


The left-hand graph shows how our response rating relates to our explanatory variable, reviews, and the number of reviews received by app. There is an apparent relationship between the number of reviews and the rating an app receives. We can see that apps that are highly rated, tend to receive more reviews than those that are low rated. The right-hand graph shows the apparent relationship between rating and our explanatory quantitative variable price. Our initial analysis indicates that low rated apps are generally free and those who charge a high price tend to be more highly rated.

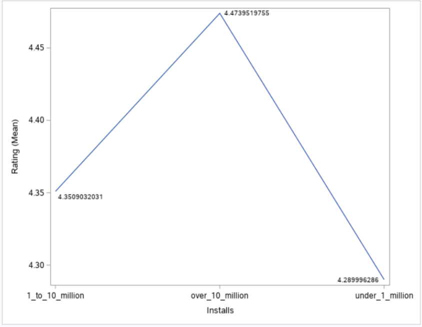
Categorical Variables Relationship



This graph shows the difference in mean Rating, between Content rating categories. Apps rated for adults only scored the highest while apps that were not rated scored the lowest. The difference in mean rating shows that there is a relationship between Content rating and Rating. This relationship makes the variables appropriate for analysis through multiple linear regression.



This graph shows a difference in means between the category/genre of an app, and its user rating. This observed difference means that there is a relationship between category and app rating, and thus using multiple linear regression for this data is appropriate.



This graph shows a difference in means between the number of installs for each app and its respective user rating, meaning that a possible relationship exists between them, thus making it appropriate for multiple linear regression.

Variable of Interest

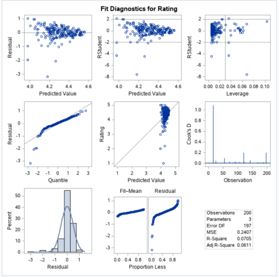
One factor that we are particularly interested in is an app's price. For price, we're curious to see if having to pay more for an app will result in a significant change in the user rating--perhaps pricier apps will have better features than cheaper apps and will thus have higher ratings, or maybe consumers will generally view pricy apps as ripoffs and give them lower ratings. Either way, we suspect price to be a fairly significant factor.

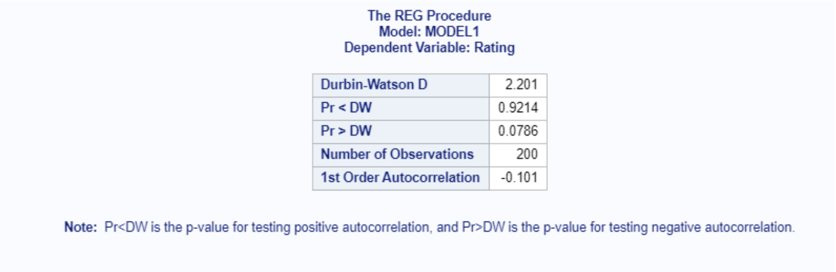
Analysis

The goal of our project is to predict User Score using multiple linear regression. User score is a continuous and quantitative variable, making it an appropriate response variable when using multiple linear regression. Our dataset represents a sample of apps selected from the population of Google Play store, as the Google store has about 2.8 million apps in total. Most importantly our data satisfies the requirements needed to conduct multiple linear regression. Firstly, our dataset contains 31,997 observations, thus meeting the requirement of having at least 200 observations. Secondly, our data set has at least 5 qualitative/quantitative variables with at least one qualitative variable. Specifically, the quantitative variables for this dataset are price and number of reviews, and the qualitative variables are number of installs, genre, and content rating.

Regression Assumptions

Looking at the following graphs, we checked each of the regression assumptions:





The REG Procedure
Model: MODEL1
Dependent Variable: Rating

Durbin-Watson D	2.201
Pr < DW	0.9214
Pr > DW	0.0786
Number of Observations	200
1st Order Autocorrelation	-0.101

Note: Pr<DW is the p-value for testing positive autocorrelation, and Pr>DW is the p-value for testing negative autocorrelation.

1. E(ε) = 0

This assumption is met because we see no clear trends in the residual graphs.

2. Homoscedasticity

This assumption is met because while there was some slight fanning in the residual graphs, the data was mostly concentrated in a horizontal band near 0.

3. The errors are normally distributed for any setting of the independent variables.

This assumption is met because the histogram is fairly normal, and the points on the QQ plot mostly follow the line.

4. The errors are independent of each other (uncorrelated)

To check this assumption we can do the Durbin-Watson test, and for negative autocorrelation we rejected the null hypothesis, meaning that the errors are autocorrelated. However, we choose not to add the autocorrelation term to correct the model.

Model Building

Stage 1:

In the first stage of model building, we dealt with quantitative variables. We transformed the explanatory variables using log, because price and reviews were heavily skewed. The log of 0 is also undefined, so we changed any zeros in the data to .001 before the transformation. We also didn't believe that our data had any quantitative-quantitative interactions.

This was the stage 1 model:
Predicted Rating = 4.15688 +(0.02632)*ln(price) + (0.03950)*ln(reviews)
Global F-test:
p-value: .0007 < .1
Since .0007 < .1, this model is significant.

Stage 2:

For the second stage, we included the qualitative variables.
This was the stage 2 model:

Predicted Rating = β_0 + β_1*ln(price) + β_2*ln(reviews) + β_3*DummyContentB + β_4*DummyInstallB β_5*DummyInstallC + β_6*DummyCatB + β_7*DummyCatC + β_8*DummyCatD

However, when we ran the nested F-test to see if the qualitative variables were significant, we got a p-value of .63607. .63607 is greater than .1, therefore we remove the qualitative terms from the model, and we have the same model as stage 1.

Predicted Rating = 4.15688 +(0.02632)*ln(price) + (0.03950)*ln(reviews)

Stage 3:
We only have qualitative terms, so there are no qualitative-quantitative interactions to add, and we retain the same model from stage 2.
Predicted Rating = 4.15688 +(0.02632)*ln(price) + (0.03950)*ln(reviews)

Added Techniques/Robust Model

We didn't choose to investigate added techniques, and instead choose to investigate using a robust model for our data. Unfortunately, using a robust model did not improve our model, since the r square for the robust was .0395, compared to the .0705 of our MLR model.

Assessing with Top Three Criteria

For MLR, we thought that the most significant criteria were p-value from the Global F-test, Adjusted R-Squared, and Root MSE.

For our model:
Global F-test p-value: .0007
Adjusted R-Squared: .0611
Root MSE: .49065

From the p-value for the global F-test we determined that our model was significant. However, a .0611 Adjusted R-Squared meant that only 6.11 percent of the variation in the predicted rating can be explained by the model. This is fairly low, and we believed it was an indication that our model was not that good. The Root MSE was .49065, meaning that we expect our prediction to be within 2*(.49065) = 0.9813 stars of the true rating. With an interval of almost a full star, that leaves a large amount of uncertainty.

Final Model

For each stage of the model building process we had the same model, and the global F-test deemed this model significant. Therefore, the model we choose to go with was:

Predicted Rating = 4.15688 +(0.02632)*ln(price) + (0.03950)*ln(reviews)

Conclusion

From our analysis, we determined the best model to come from multiple linear regression, with the following as our final prediction equation for an app's rating:

Predicted Rating = 4.15688 +(0.02632)*ln(price) + (0.03950)*ln(reviews)

While this model is statistically significant, it is unfortunately not significantly useful. The beta values associated with the final variables (price and reviews) were too small to really tell us anything of importance, meaning that they had a relatively negligible effect on the predicted user rating. Additionally, with an adjusted R-Squared of only 0.0611, the final model only explained approximately 6% of the variation in user ratings with price and number of reviews present--with 94% of the variation left unexplained, we did not believe this model to be very useful or significant in predicting user rating. Additionally, the the root MSE was .49065, meaning that we expect our prediction to be within 2*(.49065) = 0.9813 stars of the true rating--with an interval of almost a full star, the predictions could not be heavily trusted as accurate.

While we were not interested in any one app's rating per se, if we were to actually use this model for prediction, we might try to see what an app costing $1.00 with 1000 reviews (both within the experimental region) would recieve for a rating. In that case, running that in the model would yield the following result:

$$\text{Predicted Rating} = 4.15688 +(0.02632)*\ln(1.00) + (0.03950)*\ln(1000) = 4.4297$$

Another problem this example illustrates is that according to this model, an app's rating cannot be lower than 4.16. Since an app cannot be negative in price or number of reviews, this equation suggests an app's lowest possible rating is 4.16, when that is obviously not the case. Clearly, our model is not very effective in determining an app's rating.

For future research, we believe this model could be improved mainly with the inclusion of new variables. One such variable could be the size of the app--perhaps larger apps tend to have more content to satisfy the user, or something along those lines. While we did not have access to each app's size, future researchers could feasibly use additional web scraping tools to obtain that data from the Google Play Store. Another realistic variable that could have an effect may be the country of the user who gives the review--maybe culture or demographics play a part in what a person likes in an app or whether he or she determines to be an appropriate review; perhaps some countries' populations tend to leave more reviews than others. Again, this sort of data is likely available to the public, so future researchers could also use this.

One more variable that would likely have an effect on an app's rating would be the number of major incidents that happen to an app. However, "major" is somewhat subjective, and defining it would be slightly difficult--ideally, it would track how many times an app has a malfunction or problem such as a widespread outage or privacy breach and tally those up. This tends to be the cause of many apps' low ratings; for example, the stock brokerage app "Robinhood" had a fairly decent rating (around 4.5 stars) before it had a day-long outage, thus preventing its users from buying or selling anything for a full trading day (USA Today). After that outage, of course, the app's rating fell to about 3 stars and has since slowly gone up to about 3.2 stars. If future researchers could devise a way to track major app incidents, maybe by developing AI to scan through app news articles, then we could see what kind of effect major incidents may have on an app's rating compared to our other variables, thus making the model more accurate.

Additionally, as earlier stated in the analysis section, our data violated the autocorrelation assumption. To correct this assumption and make the model more accurate, future researchers could add an autocorrelation term to the model.

Methodology Comparison

For our methodology comparison section, we will compare the technique of multiple linear regression with the Lasso regression procedure. According to Charles Zaiontz at RealStatistics.com, the Lasso technique is named after an acronym that stands for Least Absolute Shrinkage and Selection Operator. Lasso is used to simplify linear models in order to avoid overfitting and multicollinear. It works by adding an L1 Regularization term to an existing linear model.

Given that the Lasso procedure will operate as a minor addition to a linear model, there are no new assumptions to be tested when one is to perform the operation. That is, all of the assumptions of the Lasso regression are derived from the ordinary least squares (OLS) regression technique. According to Ashish Dutt from R-Bloggers, one would utilize the Lasso technique in order to avoid two common statistical modelling problems: overfitting (a model that is unstable due to its high complexness), multicollinearity (correlation between two or more terms in the model), or both. By adding a penalty term that is equal to the absolute value of the magnitude of coefficients, L1 regularization simplifies the model by making every parameter being closer to 0. In contrast with the Ridge procedure, the Lasso technique does eliminate insignificant parameters in the model if they are close enough to 0.

As we can see, the Lasso regression technique is a relatively simple process that can be used to create an improved model in the mentioned variety of circumstances. It is an effective way to reduce a model's complexity, preventing overfitting and proves as a useful addition to models with high levels of multicollinearity. The Lasso technique of shrinkage can also be used to automate the parts of the model building process, such as eliminating parameters and the select variables. In this sense we can also consider the Lasso procedure as an automatic method for the selection of features in our model. Like any other statistical technique, the use of the Lasso technique can present some drawbacks for our estimation. As we discussed, Lasso regression is great at creating a simplified model, but a simplified model is not always a better model. When dealing with predictors that are highly correlated, Lasso regression chooses just one of the predictors rather arbitrarily, not taking into account any theoretical considerations, instead on a completely statistical basis. Because of this, Lasso regression leaves no room for human interpretation in the selection of predictive variables. Another downfall of Lasso regression technique is that there is no straightforward way to calculate and contrast the P-value of the parameters. Finally, Lasso regression is mainly performed by utilizing statistical softwares and the performance of the operation on each of these can differ slightly. That is, one might get two different models when performing Lasso regression in R and in SAS.

In summary, Lasso regression is a helpful way to create simpler models under some circumstances. This does not mean that theoretical foundations and human decision in the selection of parameters should be abandoned all together, as this, in most circumstances will yield a worse result in our modelling efforts of our response variable.

SAS Code

```
*quantitative EDA;
proc sgscatter data=mydata.GOOGLE_PLAYSTORE_DATA;
plot Rating*(Reviews Price);
run;

*take only the E and Teen groups;
data mydata.new_E;
 set mydata.GOOGLE_PLAYSTORE_DATA;
 where Content_Rating = 'E';
 run;

data mydata.new_T;
 set mydata.GOOGLE_PLAYSTORE_DATA;
 where Content_Rating = 'Teen';
 run;

*Take a sample;
proc surveyselect data= mydata.new_E method=srs n = 100
seed=180 out=mydata.new_sample_E;
strata Content_Rating;
run;

proc surveyselect data= mydata.new_T method=srs n = 100
seed=180 out=mydata.new_sample_Teen;
strata Content_Rating;
run;

*regression;
proc reg data = mydata.combo_new plots=none;
model Rating = log_new_rev log_new_price;
run;

*calculate p-value from f statistic;
data cutoff;
fcritical=quantile("F",.90,6,191);*quantile("F",1-alpha,k-g,n-k-1);
pval=sdf("F",0.71859505548,6,191); *sdf("F",test statistic,k-g,n-k-1);
proc print data=cutoff;
run;

*log transformation;
data mydata.combo_new;
   set mydata.combo;
   if Price = 0 then new_price = .001;
```

```
  if Price ~= 0 then new_price = Price;
  if Reviews = 0 then new_rev = .001;
  if Reviews ~= 0 then new_rev = Reviews;
run;

data mydata.combo_new;
  set mydata.combo_new;
  log_new_price = log(new_price);
  log_new_rev = log(new_rev);
run;

*make dummy variable;
data mydata.combo; *original data table;
        set mydata.combo; *set the original data table (add new columns);
        DummyContentB=0; *create a variable called DummyGenreB where every observation
will have a value of 0;
        if Content_Rating = 'E' then DummyContentB=1; *similar to above;
run;

data mydata.combo; *original data table;
        set mydata.combo; *set the original data table (add new columns);
        DummyInstallB = 0; *create a variable called DummyGenreB where every observation
will have a value of 0;
        DummyInstallC = 0;
        if  Installs = 'under_1_million' then DummyInstallB=1; *similar to above;
        if  Installs = '1_to_10_million' then DummyInstallC=1; *similar to above;
run;

data mydata.combo; *original data table;
        set mydata.combo; *set the original data table (add new columns);
        DummyCatB = 0; *create a variable called DummyGenreB where every observation will
have a value of 0;
        DummyCatC = 0;
        DummyCatD = 0;
        if  Category = 'Lifestyle' then DummyCatB=1; *similar to above;
        if  Category = 'Media' then DummyCatC=1; *similar to above;
        if  Category = 'Productiv' then DummyCatD=1; *similar to above;
run;

proc sgplot data= mydata.combo;
        vline Reviews / response=Rating datalabel stat=mean;
        *vline quantitative variable / response = response variable datalabel stat=mean;
        *stat = mean gives the means value of y for each category of the qualitative variable;
Run;
```

```
proc sgplot data= mydata.combo;
        vline Category / response=Rating datalabel stat=mean;
        *vline quantitative variable / response = response variable datalabel stat=mean;
        *stat = mean gives the means value of y for each category of the qualitative variable;
Run;

proc sgplot data= mydata.combo;
        vline Installs / response=Rating datalabel stat=mean;
        *vline quantitative variable / response = response variable datalabel stat=mean;
        *stat = mean gives the means value of y for each category of the qualitative variable;
Run;
```

Works Cited

Dutt, Ashish. "Basic Assumptions to be Taken Care of When Building a Predictive Model." *R-Bloggers*, 17 Jan. 2017, www.r-bloggers.com/basic-assumptions-to-be-taken-care-of-when-building-a-predictive-model-2/.

Goodwin, Jazmin. "Trading Restored on Robinhood After Facing Its Third Outage This Week." *USA Today*, 9 Mar. 2020, www.usatoday.com/story/money/2020/03/09/ robinhood-hit-massive-outage-markets-plummet/5001963002/.

Nelson, Randy. "Percentage of All Global Web Pages Served to Mobile Phones from 2009 to 2018." *Sensor Tower,* 22 July 2017, www.sensortower.com/blog/app-revenue-and-downloads-q1-2019.

Statista. "Percentage of All Global Web Pages Served to Mobile Phones from 2009 to 2018." *Statista*, 22 July 2017, www.statista.com/statistics/241462/global-mobile-phone-website-traffic-share/.

Weedmark, David. "The Advantages & Disadvantages of a Multiple Regression Model." *Sciencing*, 13 Mar. 2018, www.sciencing.com/advantages-disadvantages-multiple-regression-model-12070171.html.

Wesleyan University. "Lasso Regression Limitations" *Coursera,* n.d., coursera.org/ lecture/machine-learning-data-analysis/lasso-regression-limitations-64xnZ.

Williams, Ken. "Multiple Linear Regression–MLR Definition." *Investopedia*, 14 Apr. 2019, www.sciencing.com/advantages-disadvantages-multiple-regression-model-12070171.html.

Zaiontz, Charles. "Real Statistics Using Excel." *Real Statistics*, n.d., www.real-statistics.com/

multiple-regression/ridge-and-lasso-regression/lasso-regression/.