Google Play Store Case Study
What makes a popular app?

By Bradford Murphy & Jayke Sudana

Introduction

Project Description

Google Play Store

- An app developer approaches us with plans of developing a new app and releasing it onto the Google Play Store. The developer does not know exactly what the app will look like in terms of size, functionality, category, etc.
- There are 3.04 million apps on the Google Play Store.
- <1% of apps are downloaded 1m+ times (“Android app download ranges 2018”).
- Our data set included 1.1 million rows of apps with 23 column attributes.

Project Objective

Project Goals

- ➢ Create a dependent variable which best describes an app's success.
- ➢ Model previous 2020 Google Play app installation behavior to analyze what combination of parameters makes a user more likely to install an app.
- ➢ Narrow down the data set to an accurate representation of the 2020 Google Play Store.

Analytics Questions

- Is there a correlation between the App's rating and the number of installs?
  - Does an app with a higher rating tend to have more installs? Are there outliers?
- Do free apps tend to be downloaded more than non-free apps?
  - What is the correlation between app price and the number of installs?
- Is there a correlation between app category and the number of installs?
- Does the app size have an effect on the number of downloads?
- Are apps released earlier in 2020 installed more on average than later releases?
- Is there a correlation between the content rating and the average number of downloads?
- Do apps that receive an 'Editors Choice' rating get more installs?
- Do ad support apps fare better than non-ad support apps?
- Are apps with in-app purchases more download than apps without in-app purchases?

Data Preprocessing

Handling Missing Data / Selecting Random sample

- Missing data
  - The Google Play Dataset had more than one million Records; a small percentage of the records had missing data.
  - Any record containing missing values was deleted. With such a large data set we felt this had no impact on our results.
- Getting a sample of 10,000 values
  1. Changing the time frame: we used only 2020 data and deleted data from all other years.
  2. Category selection: we included only the top 5 app categories (Education, Business, Lifestyle, Music, and Entertainment).
  3. Excel RAND() function: we created a new column containing random values and then sorted the dataset by this column. Selecting the first 10,00 records gave us a usable random sample.

Correlation Table & Summary Characteristics

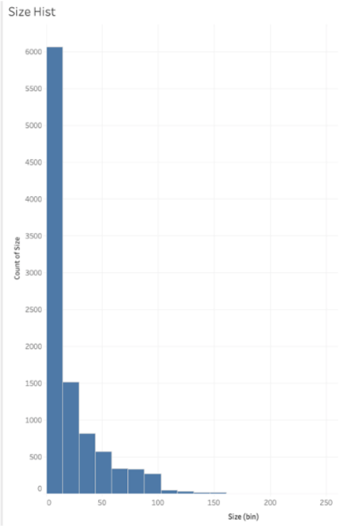
A	B	C	D	E	F	G
	Rating	*Rating Count*	*aximum Instal*	*Size*	*Days*	*nstalls per Day*
Rating	1					
Rating Count	0.09242658	1				
Maximum Ins	0.0833606	0.75439484	1			
Size	0.11211274	0.0110148	0.01028715	1		
Days	0.17666736	0.04458022	0.04632868	-0.05505	1	
Installs per Da	0.06408627	0.53607985	0.69877842	0.03506801	-0.0052175	1

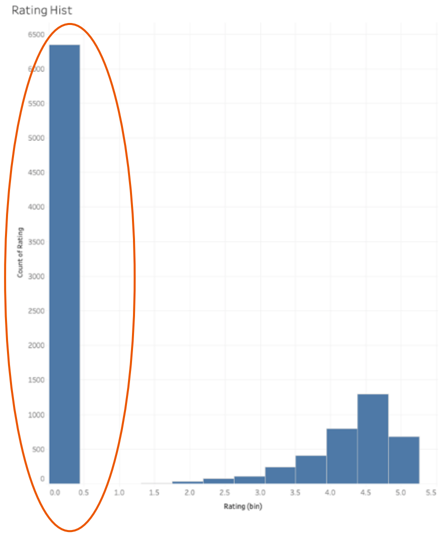
	Rating	Rating Count	Size	Installs per day	Min Android
Mean	1.56128	113.8076192	21.87	77.78349244	4.38168183
Med	0	0	9.80	1.463414634	4.1
St dev	2.09414	2874.429515	25.627345	1453.549222	0.57211063

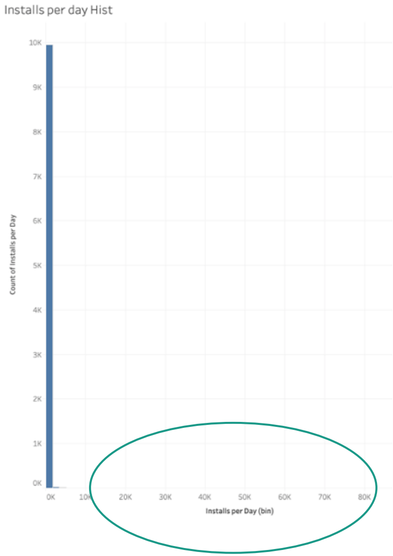
- Nothing found in the correlation table was alarming.
- The correlation values shown in the bottom row indicate that the numerical variables are likely to have an effect on our dependent variable.

- The standard deviation values of our numerical variables seemed large.
- This meant that some of our variables had values which were very spread out.

Histograms







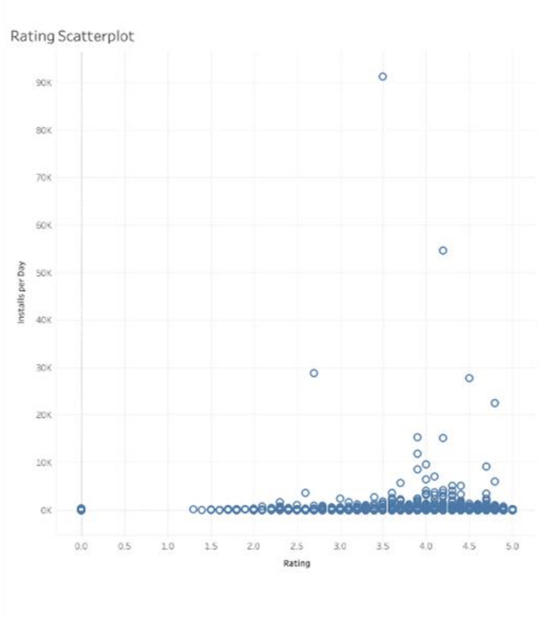
Issues Discovered:

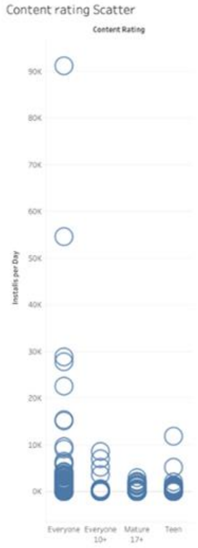
- Many values having a rating count of zero
- “Installs per day” variable has many outliers

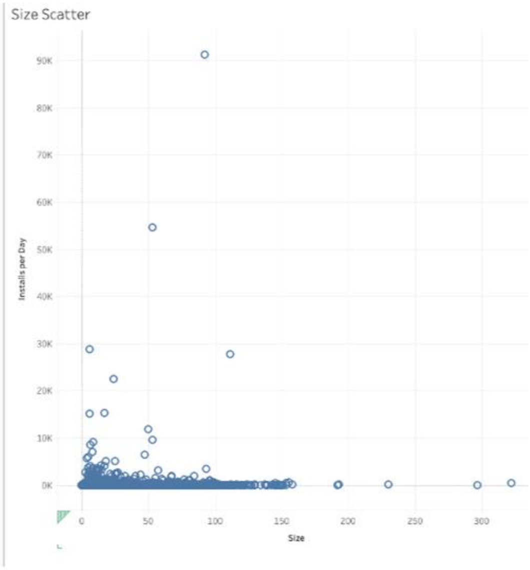
Solutions:

- Standardize data with log transformation
- Delete outliers

Scatterplots





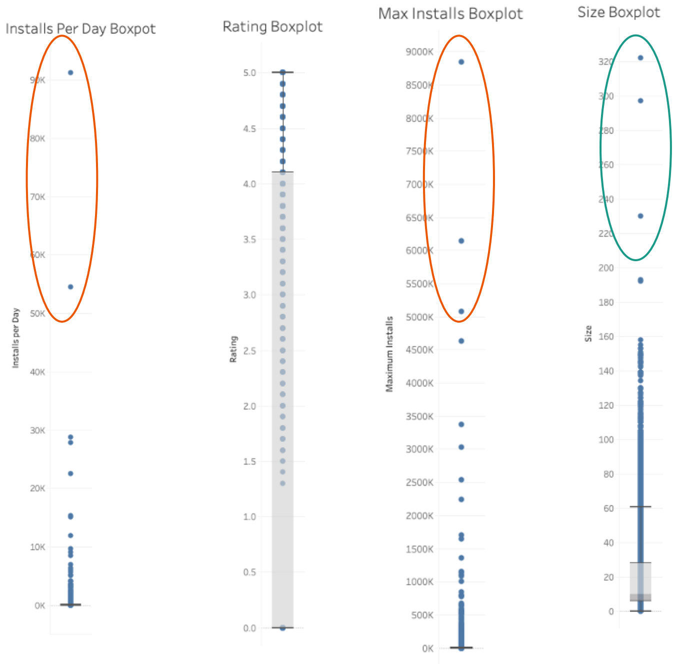


Correlations Found:

- Higher Rating ⇒ More Installations
- Larger App ⇒ More Installations
- No clear correlation between size and installs
- Apps rated “everyone” ⇒ More Installations

***should be noted that the majority of apps are rated “everyone”

Box Plots



Issues Discovered:

- Outliers in categories involving number of installs
- Outliers in “Size”

Solutions:

- Standardize data with log transformation
- Delete outliers

Model 1: Linear Regression

Why Linear Regression?

- We are using linear regression because our dependent variable is numerical.
- Linear regression gives us an understanding of which variables are important in determining the outcome of our dependent variable (installs per day).
- Variable Selection is automatic.

Results

Predictor	Estimate	Confidence Interval: Lower	Confidence Interval: Upper	Standard Error	T-Statistic	P-Value
Intercept	64.59743865	-3.473560729	132.668438	34.72373028	1.860325435	0.0628884
Rating Count	0.664416513	0.63880322	0.690029806	0.013065609	50.85231772	0
Size	1.497940029	0.307286298	2.68859376	0.607364948	2.466293179	0.0136797
Days	-0.501807028	-0.824929877	-0.178684178	0.164828353	-3.04442179	0.0023414

Model Interpretation

Equation

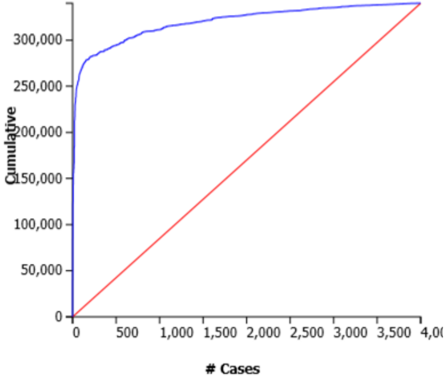
- **Installs per day = 64.60 + 0.6644RatingCount + 1.4979Size - 0.5018Day**

Interpretation

- Installs per day increases by 0.6644 when Rating Count increases by 1, holding other variables constant.
- Installs per day increases by 1.4979 when Size increases by 1, holding other variables constant.
- Installs per day decreases by 0.5018 when Days increases by 1, holding other variables constant.

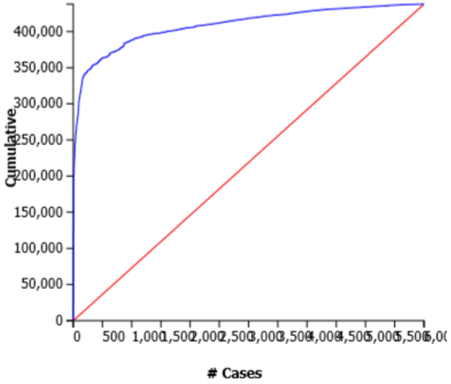
Model Performance

Training



Metric	Value
SSE	8611771471
MSE	1435056.069
RMSE	1197.938258
MAD	104.9263355
R2	0.302568959

Validation



Metric	Value
SSE	11097030239
MSE	2774257.56
RMSE	1665.610266
MAD	121.0292388
R2	-0.263918711

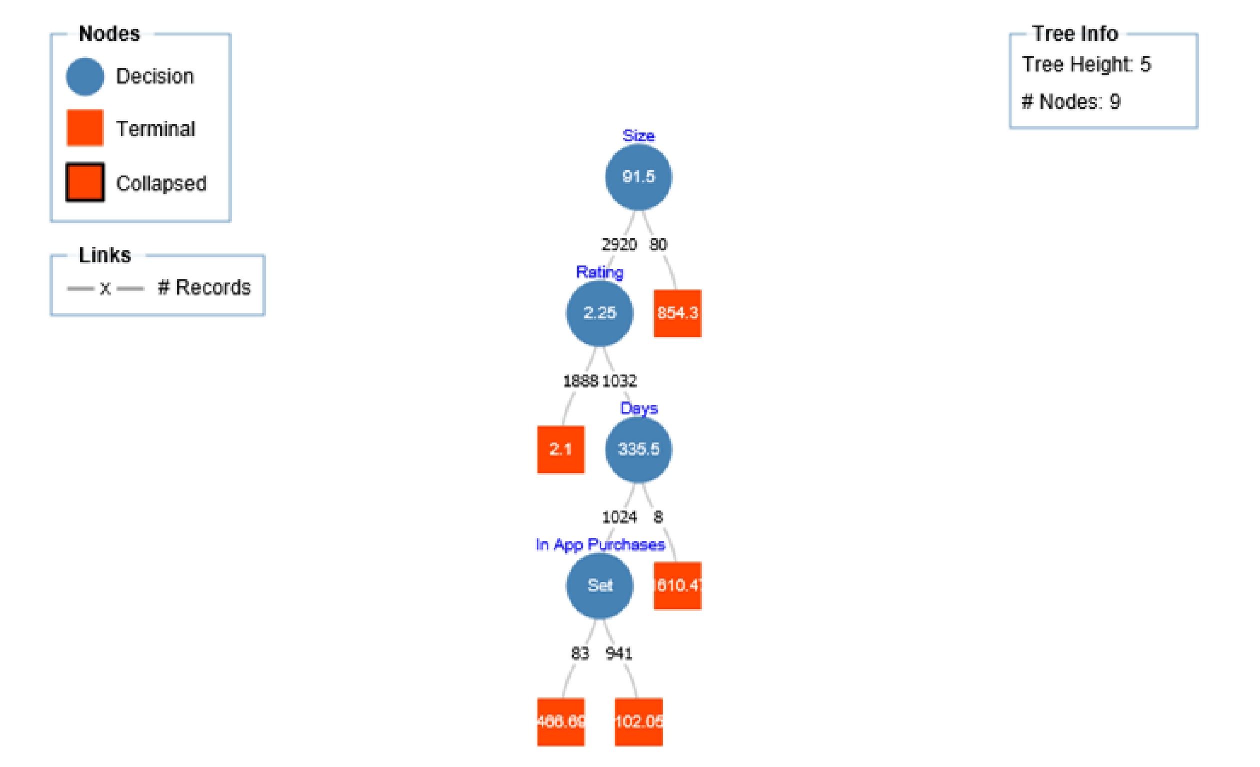
- The model performs very well as shown in the lift charts.
- When comparing training and validation data and lift charts, we saw that the AUCs for both charts were quite similar. The RMSE for validation (1197.9) was a bit lower than the training data's RMSE (1665.6), while the R^2 statistic for Validation (0.30) was higher than that of Training (-0.26).

Advice to Firm

All advice has our primary goal in mind: **increasing the number of installation the app receives**

1. Get more users to review the application.
    - Add push notifications asking users to leave a reviews.
2. Market heavily soon after the app is released to capitalize on the app's early "buzz".
    - Apps who receive many downloads soon after release get a "push" from the Google Play Store.
3. Make a high performing app, even if the app has to be large.
    - Users *don't care* if the app is large. They want the app to work well.

Model 2: Regression Tree



Model & Variable Selection

Purpose of Regression Tree

- Easy to interpret and understand.
- Tree gives simplified insight into certain variables that are most important to determining an app with high “Installs per day”.
- Variables are automatically selected.

Variable Selection

- Independent variables include category, rating, size, days, minimum android, content rating, ad supported, free, in app, and editors choice.
- Rating Count was excluded due to high correlation with dependent variable.

Fully Grown Tree:

Nodes

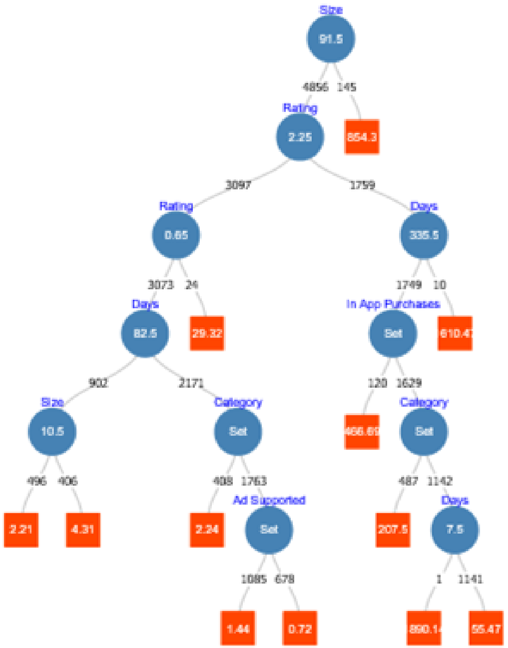
Decision

Terminal

Collapsed

Links

— x — # Records



Tree Info

Tree Height: 7

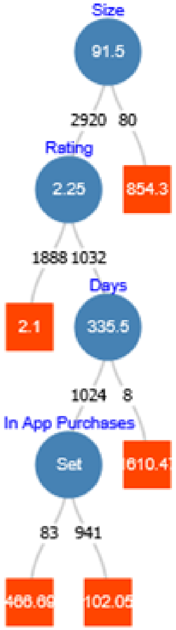
Nodes: 23

Best Pruned Tree:

Tree Info

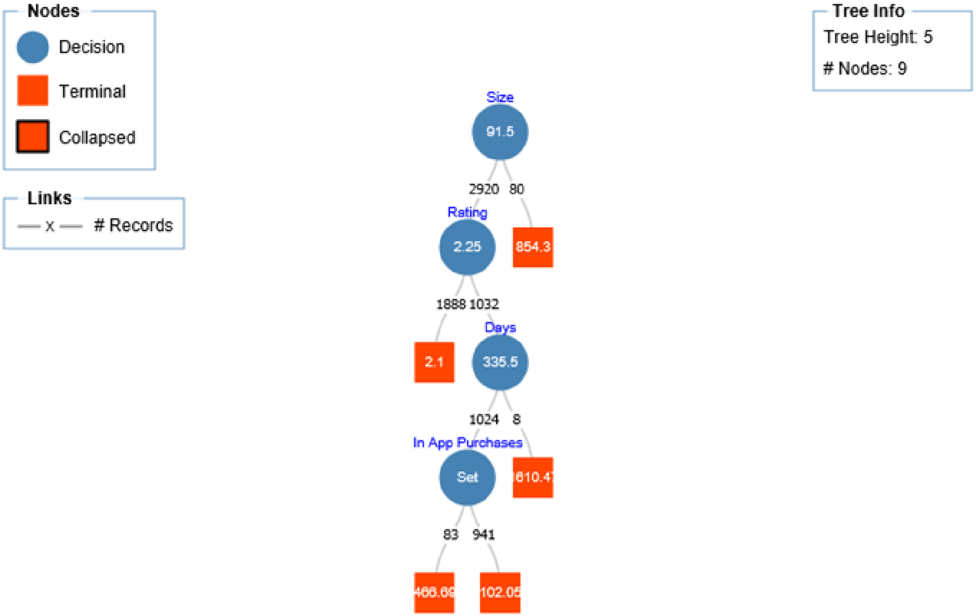
Tree Height: 5

Nodes: 9



Model Output & Interpretation

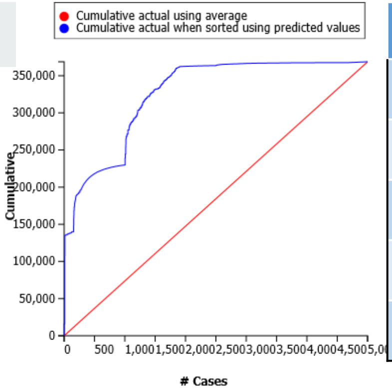
- Go left if Size < 91.5 and right if Size ≥ 91.5.
- Go left if Rating < 2.25 and right if Rating ≥ 2.25.
- Go left if Days < 335.5 and right if Days ≥ 335.5.
- Go left if In App Purchases is True and right if In App Purchases is False.



Training, Validation, and Test Results

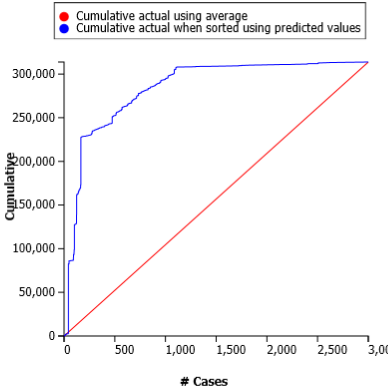
- The A.U.C for each model demonstrate that the model works well.
  - Validation with largest AUC, then Training, then Test.
- Validation displayed highest RMSE of 1894.02.
  - Training with 1387.74 and Test with lowest, 483.60.
- Validation and Training have similar R^2 values (0.014 and 0.015, respectively).
  - Training has lowest R^2, with -0.099.

Training



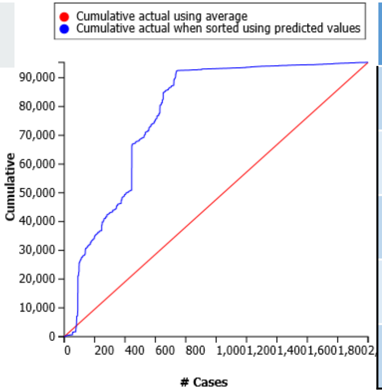
Metric	Value
SSE	9631093407
MSE	1925833.515
RMSE	1387.744038
MAD	115.2019085
R2	0.015102781

Validation



Metric	Value
SSE	10761901269
MSE	3587300.423
RMSE	1894.017007
MAD	146.1921555
R2	0.014428844

Test



Metric	Value
SSE	467734656
MSE	233867.328
RMSE	483.5983127
MAD	94.23957275
R2	-0.098582901

Advice to Firm

- According to our decision tree model, the apps which a developer should focus on include app size, rating, days the app has been available, and in-app purchases.
- Larger sized apps tend to receive more downloads.
    - A high performing app can take into account latest technologies, personalization, connectivity, business solutions, etc.
- Apps with high ratings tend to receive higher installs.
- The longer an app is available, the more likely it is to be installed.
- In-App purchases are a strong add-in for increasing downloads.
    - “Monetization is directly linked to engagement” (“Driving Buyer Behavior with In-App Purchases”)

Conclusion

Which Model Should Be Used?

We recommend using the decision tree model.

Justification for Decision Tree

- Regression tree addressed our business questions more fully.
  - Parameters relevant to our goals were addressed and included by the tree model more so than the linear regression.
  - The tree allowed us to provide a more holistic conclusion to the firm.
- The tree model does not assume normal distribution; this addresses the outlier issue we had with the linear regression model.
- The model handled outliers while providing accurate results as seen in the lift charts.
- The tree was easier to interpret and understand.

We Learned...

- How to create effective research questions.
- How to clean, process, and sample a dataset of over 1 million values. It Was Hard!
- How to effectively visualize data and find issues through visualization.
- How to use models to draw conclusions from data.
- How to analyze results and deliver said results to a boss/client.

Issues with the Dataset

- The dataset was *huge*. Getting a representative sample of 10,000 values was difficult.
- There were unexpected outliers and some were discovered well into our analysis.
- Some inter-variable correlations proved to be problematic.
- We had to derive our own independent variable (Installs per day) using the information we has.

Sources

- *"What factors contribute to the success of a mobile app?"* : https://appinventiv.com/blog/8-key-features-makes-mobile-app-successful/

- "Driving Buyer Behavior with In-App Purchases": https://medium.com/googleplaydev/a-kpis-guide-for-google-play-apps-and-games-driving-buyer-behavior-with-in-app-purchases-a9f88564cd86

- "Android app download ranges 2018": https://www.statista.com/statistics/269884/android-app-downloads/