Kaede Yoshikawa

CSE 163

Final Project Deliverable

Analyzing Digital Games and its Gamers

Executive Summary

Which features of a mobile app correlate to the average rating?

None of the app features (such as the price, whether it contains ads, or app size) correlated to the rating, but some categories (such as the genre and the content rating) did have some correlation with regards to the ratio of reviews.

Are negative reviews worth considering in considering how addictive a game is?

Negative reviews can be an indicator for whether the game will be addictive relative to other games, but a high ratio of negative reviews will not necessarily indicate how worthy it is to play.

To what extent do social media trends influence how people interact with mobile games?

A lower mode value of the trend, which indicates the presence of a sudden surge in the number of searches, seems to have a higher number of installs.

Can machine learning predict the average rating of a game, and how accurately can the average rating be predicted?

The educated model was able to predict the average rating of a game with an error of about 0.34, which is pretty good.

Motivation

There has been a rise in the number of digital games in recent years. Because it is difficult for a person to decide whether a game will be good before playing, I wanted to investigate what elements of a game contributes to a "good game". By analyzing the features of a game and the rating system itself, there can be a more thorough consideration regarding virtual game stores and their systems.

During this, I will make a general assumption that the rating system imposed by most virtual game stores do a decent job of capturing the mostly honest feedback.

Dataset

- Google Play Store data
- Google Trends (US)
    a. Data obtained via pytrends
- Steam Store data

Method

Which features of a mobile app correlate to the average rating?

By plotting individual features of a game (such as the price) over various factors of a rating of a game, I can identify factors that correlate to the rating, which I can then use to evaluate the machine learning model later on. For this analysis, I will generally assume that most reviews are honest (not spam or rated by people that have not played).

Method

1. Download and unzip the Google Play Store data
2. Read and convert the JSON file into a pandas DataFrame. Use the list: ['title', 'minInstalls', 'score', 'ratings', 'reviews', 'price', 'inAppProductPrice', 'size', 'genre', 'genreId', 'contentRating', 'containsAds'] to filter any columns not used for analysis. Save a copy of the "histogram" column (contains lists of the number of reviews for each star, with index 0 being the 1-star rating) to extract the number of 1-star reviews and the number of 5-star reviews. In the end, there should be:
    a. Title of the game (string)
    b. Number of installs in log base 10 scale (integer)
    c. Average rating (float)
    d. Number of reviews in log base 10 scale (integer)
    e. Number of 1-star reviews (integer)
    f. Number of 5-star reviews (integer)
    g. Price (float)
    h. In-app product price (is null if there is none, float)
    i. Size (If the value is "Varies with device", then set to None, but else, convert to a float value after removing the last character ("M"), float)
    j. Genre ID (string)
    k. Content rating (string)
    l. Whether contains advertisements (Boolean)

   Save this new DataFrame as a new csv file to use later again.
3. Plot the following plots with seaborn:
    a. Scatter plot of number of installs (x) vs average rating (y)
    b. Scatter plot of number of reviews (x) vs average rating (y)
    c. Scatter plot of in-app product price (x) vs average rating (y)
    d. Scatter plot of size (x) vs average rating (y)
    e. Scatter plot of price (x) vs average rating (y)
    f. Scatter plot of overall price (price + in-app product price * 0.05) (x) vs average rating (y), colored by whether or not it contains ads
    g. Box plot of genre ID (x) vs average rating (y)
        i. Use groupby beforehand to get the average average rating
    h. Box plot of content rating (x) vs average rating (y)
        i. Use groupby beforehand to get the average average rating
    i. Above graphs repeated with the number of 1-star reviews and 5-star reviews instead of the average rating

Are negative reviews worth considering in considering how addictive a game is?

By plotting the playtime against the number of positive reviews, the number of negative reviews, and the percentage of negative reviews, I can identify whether negative reviews are reflected in the playtime, which will indicate the "addictiveness" of a game. In this analysis, I will assume that the "addictiveness" is a better indicator of a goodness of a game compared to the number of reviews, which allows me to analyze the rating system implemented by Steam based on this assumption.

Method

1. Download the Steam Store data and convert to a pandas DataFrame
2. Create a calculated column of the ratio of the mean playtime to the median playtime and the percentage of negative reviews.
3. Plot the following scatterplots with seaborn:
    a. The percentage of negative reviews (x) vs the mean playtime on log scale(y)
    b. The percentage of negative reviews (x) vs the median playtime on log scale(y)
    c. The percentage of negative reviews (x) vs the mean/median playtime on log scale(y)

To what extent do social media trends influence how people interact with mobile games?

By plotting the number of installs against metrics that would indicate the trend of the data, I can identify whether the trend, indicated by the popularity of a search, reflects people's actions (number of installs) or inclinations (the average rating).

Method

1. Using the CSV file created from the first research question, obtain the search trend for 5 years for each game using pytrends. If there is an error due to too many requests, split the list of games into smaller pieces, generate a csv, and concatenate all the csv into one.
2. Calculate:
    a. Q3 (75 percentile location)
    b. mode of the interest values
    c. standard deviation of the interest values
3. Plot the calculated columns (x) against the number of installs on log scale and the average rating.

Can machine learning predict the average rating of a game, and how accurately can the average rating be predicted?

By educating an algorithm to make an estimate of the rating of a game, one can possibly judge how good a game is, even if it has minimal reviews.

Method

1. Using the CSV file created from the first research question, educate a Scikit-learn Decision Tree Regressor with:
    a. Labels as the average rating
    b. Features as:
        i. Overall price
        ii. Size
        iii. Genre
        iv. Content Rating

v. Whether or not it contains ads

Save 25% of the data for testing. Limit the max depth to 5.

2. Test the accuracy of the model using the test data.

Results

Which features of a mobile app correlate to the average rating?

When I initially thought of investigating this question, my prediction was that although there may be some jittering due to the real-world data, some of the data points like the price and whether it contained ads would somewhat correlate to the average rating, for they were things that I thought would be of frustration to consumers. This did not turn out to be the case. For all the scatterplots plotted, none of them seemed to have any relationship, and seemed like random data. The presence of ads does seem to somewhat correlate with the maximum rating being lower, but it was not as large of a significance as I expected.

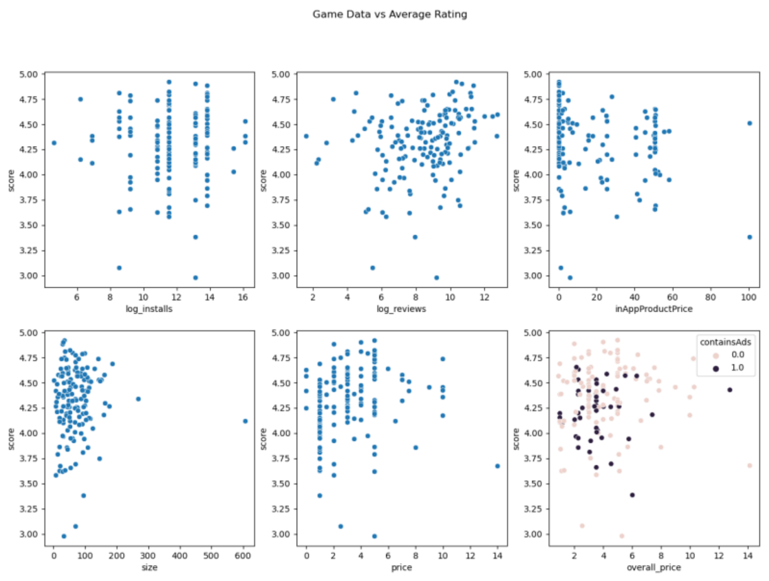
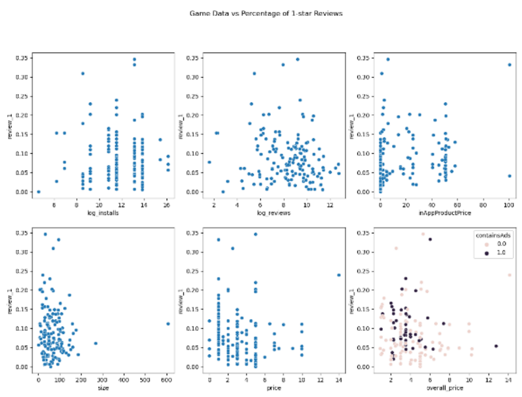


Figure 1: Plot of the data attributes (installs, reviews, in-app price, size, price, and overall price) vs the average rating. Though there are general trends in the maximum or minimum values (size vs rating, overall price), there is no strong relationship between them.



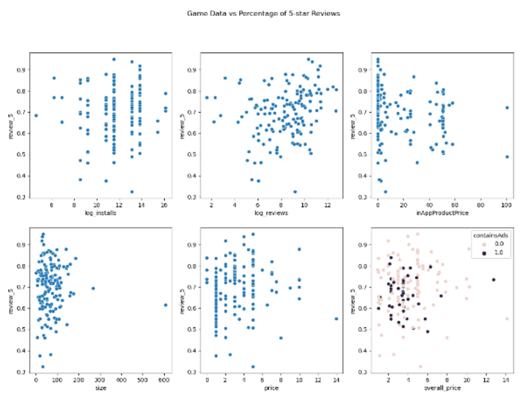


Figure 2, 3: Plot of the data attributes vs the percentage of 1-star and 5-star ratings. The results of the 5-star reviews are similar to the average rating, and the results of the 1-star ratings looks like an inverted version of the plots produced in the 5-star ratings.

As for the bar charts produced, I initially did not expect them to have much difference across the categories because I expected things like the genre or the content rating of the games played would be more of a personal preference, and therefore would not have a considerable difference. Although this was generally true for the average rating, the plots for 1-star and 5-star ratings had significantly different results for some of the categories. For example, card, casino, and puzzle games had a significantly lower ratio of 1-star ratings compared to other genres, and the number of 1-star ratings were the highest for a content rating of 17+ and lowest for a content rating of teen.

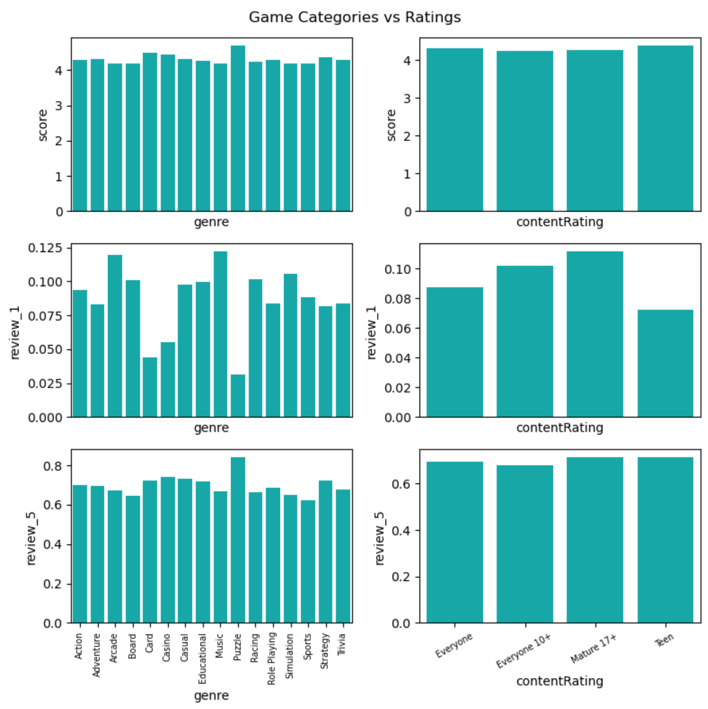


Figure 4: Plot of the game categories (genre and content rating) vs the average rating and the ratio of 1-star and 5-star ratings. The largest difference can be seen in the row for the ratio of 1-star reviews.

From this result, I can conclude that the features of a game do not contribute directly to the rating of a game, but the category of the game will affect the ratio of reviews to some degree. This means that for a game with a set category, there is no considerable difference in the rating due to the increase in price or the increase in the size, which will allow a creator to focus on their creative work rather than worrying about the impact of raising the price to make a living.

A potential reason for the features not affecting the rating could be because of a relatively wealthy population using the Google Play Store service. With more financial allowance, people can have better electronics and have a higher tolerance for spending money, so even if the game is not as good as expected, people will not be frustrated as much because of how much resources were spent.

Potential reasons for some of the categories having lower percentage of 1-star ratings could be because of the population playing the games and the playing habits that they would employ. Games like puzzle games or card games tend to have simple rules and a shorter playtime (or a long playtime that can be broken up into shorter playtimes) per game, such as a game of chess or solitaire. As these kinds of games tend to be thought-heavy, less people that are inclined to be enraged by games may play the game. Also, they may be played during leisure times in buses or waiting lines since they can be played for a short amount of time. Because the purpose is to have something to do, there would not be as much focus on the content of the game, which would create less frustration.

Are negative reviews worth considering in considering how addictive a game is?

When I initially though of this question, my prediction was that the results would be inversely proportional, for would stop playing a game if it is bad, causing games with a higher ratio of negative reviews to have a lower average or median playtime. Though this was somewhat true, it was not as drastic of a difference as I expected. Even for games with mostly negative reviews, the maximum and most frequent median/mean playtime in log scale was closer to 2, which indicates a median/mean playtime of 100 hours, which is a significant amount of time to play a game. For games with minimal negative reviews, the mode of the values was around 2 as well, but with a higher maximum mean/median playtime. Games with minimal negative reviews also had a higher ratio of mean to median, which indicates that there are more people that are more “addicted” to the game. For any rating ratio, there still will be a game with a relatively low mean/median playtime.

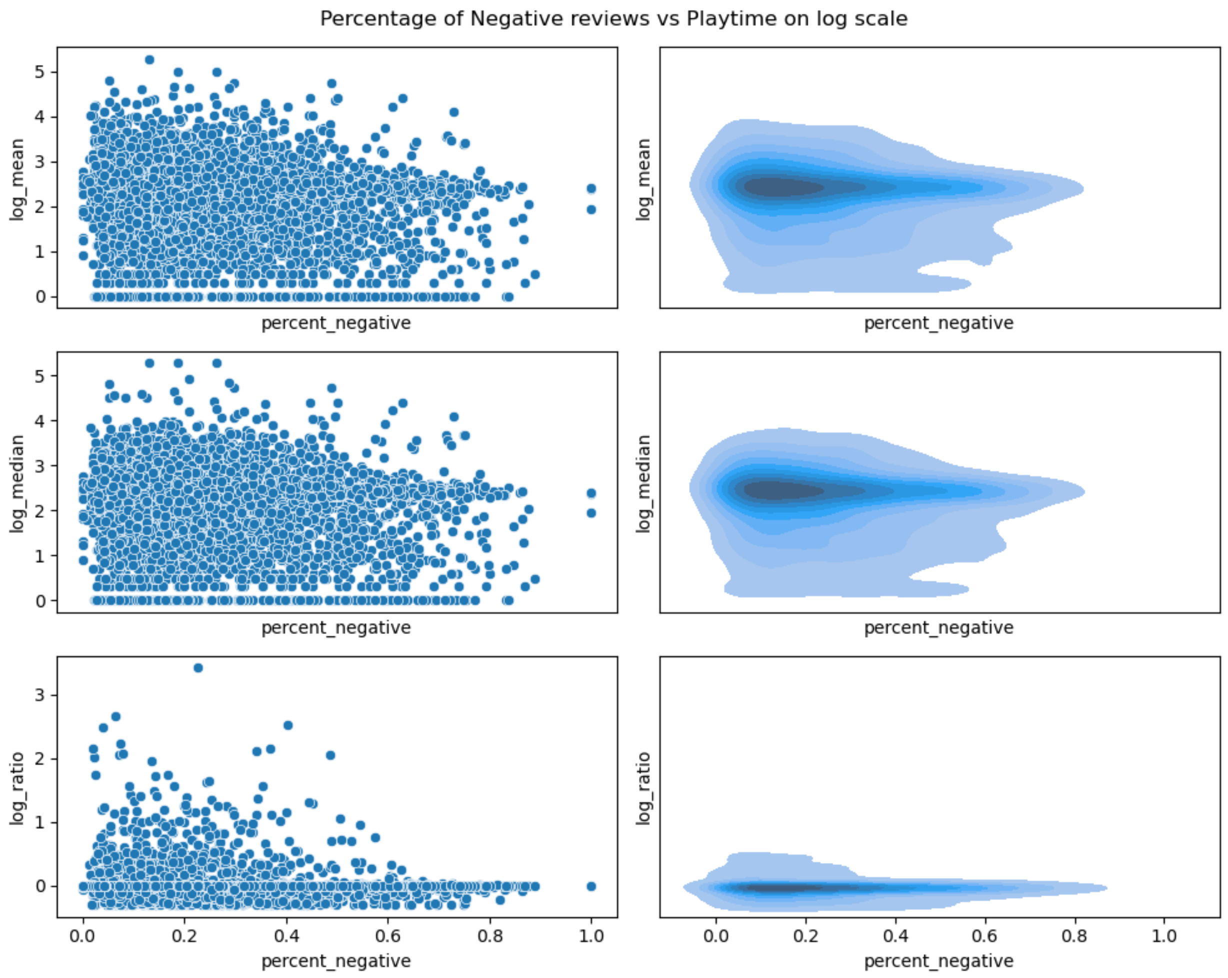


Figure 5: Plot of the percent of negative reviews vs the median playtime, mean playtime, and a ratio of the two in log scale. A horizontal trend can be seen at around 2 for the mean and median playtime.

In conclusion, the ratio of negative reviews can give some indication as to how much more "addictive" a game is relative to other games but does not strongly indicate the overall addictiveness (aka the playtime). A potential reason for this may be because of how the Steam Store reviewing system works. Because it can only be written by people who purchased the game, there would be at least some playtime before the player deems the game good or bad. For a population of gamers, it would be more likely that the player will rate the game after they have completed the game or have been totally stuck on it for a while, which would increase the mode median/mean playtime.

To what extent do social media trends influence how people interact with mobile games?

Initially, I was expecting the mode of the trend values to be inversely proportional to the popularity (whether it be the number of reviews or installs) because if there is a sudden surge in the number of searches (a buzz), then the absolute maximum value will be higher than the rest, causing most of the values to be low with some points being very high. This was somewhat true with the mode and the number of reviews being loosely inversely correlated, but all other graphs seem to be random.

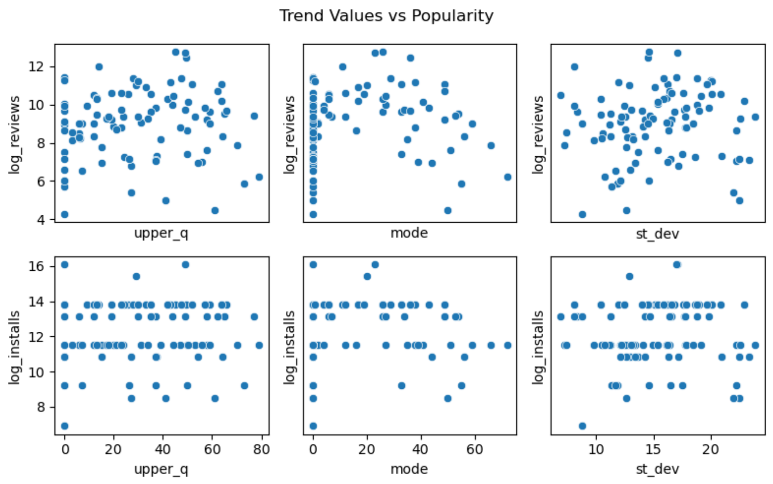


Figure 6: Plot of the trend values (75 percentile, mode, and standard deviation) vs the popularity (number of reviews and installs). There is a slight correlation in the mode vs reviews graph.

Therefore, social media trends only affect the popularity loosely, and will not significantly impact the overall installs or reviews as much. This can potentially be because even if there is a trend "buzz", there will be a limited number of people who will install, play, or review the game.

Can machine learning predict the average rating of a game, and how accurately can the average rating be predicted?

Before starting the project, I was assuming that the machine learning would do a pretty good job of estimating the average rating, for I believed that all the attributes will have some correlation to the rating. Additionally, even if the correlation is loose, there have been examples that had good results even if the data is somewhat foggy, so I was not as worried. But after I investigated the first research question, I was estimating that it would possibly have a large error because the datapoints seemed so random. Contrary to that new prediction, the educated model was able to predict the average rating of a game with an average root-mean-squared error of about 0.16 for the test dataset and 0.06 for the training dataset, which good guess for a 5-star rating system.

A rating prediction system can have real-life applications, such as the sorting algorithm for an online game store, for if the search can bring up better games first, the user may be more inclined to purchase a game from the list. A concern with this application would be manipulation of the algorithm. If a company was able to identify one factor in getting a better score (and thus a higher placement), they can use that knowledge to make their product more appealing to the algorithm than to humans or create low-quality apps that are favored by the algorithm to prevent other companies from getting customers. They can also create an imitation of a popular app, and if they were able to get a better placement than the original, then they can trick people into using their fake app, which can lead to data theft.

In conclusion, machine learning can predict the average rating of a game accurately to about 0.2 error but should not be implemented as the sole indicator of relevance in real-life situations to prevent malicious manipulation of the algorithm.

Challenge Goals

- Multiple Datasets
- Messy Data
- Machine Learning
- External Library

The challenge goal of using an external library did not turn out as expected, as I was initially planning to use plotly to make the plots nice. But, since there was no way for me to scrape the Google Trends data from the website, I had to use an external library called pytrends to obtain that data.

Work Plan Evaluation

My proposed work plan estimates were not accurate. My estimates were far from reality because I was not able to accurately assess the messiness of the data (Google Play data), and I devoted too much time trying to make the figures look nice using plotly. I also had a sudden issue with matplotlib, which caused me to search for solutions on how to resolve that instead of writing code for analysis. Also, I usually did not mind how much time I take to do a coding task, so I did not have a good basis for my estimates.

Testing

I did not use assert statements or smaller data files to test my code. While coding, I used multiple print statements, show statements for graphs, and many error statements that python threw at me to make python do what I intended to do. This was mostly because of laziness, but it was also because I did not know which parts of the data would not behave as expected and I did not have a clear stance as to what to do with all the data.

Collaboration

I referenced multiple Q&As by Stack Overflow users, library documentations, and this article, which I used to obtain the percentage to multiply to the in-app content price for the overall price.