Analyzing Steam Games by single-player vs. multiplayer stats

Rae Chen, Jason Gao, and Jack Sui

Research Questions:

1. Predicting the Rating for a game out of all total players who own this game with a given type, price, release date, and whether or not an "app" contains multiplayer content. This research can provide publishers a tool to predict the rating of their new game and maybe help them find a suitable price so that the rating won't be too low.
   (need join ACHIEVEMENT_PERCENTAGES and APP_ID_INFO by appid for this)
   - We found only a slight correlation between our selected features and the rating of a game.

2. How whether or not an "app" contains multiplayer content affects Play Time. Do they have a significant difference? We are trying to make a graph to show if there is a relationship between Play Time and whether or not an "app" contains multiplayer content.
   (GAMES_1 & ACHIEVEMENT_PERCENTAGES)
   - We found that multiplayer content does, in general, increase the amount of time people spend playing a game.

3. How do other aspects of player engagement change in multiplayer vs single-player titles? We could look at other subdivisions, achievement completion, playtime, etc. We can answer this question by graphing each relationship.
   (GAMES_1 & ACHIEVEMENT_PERCENTAGES)
   - We found that single player games have higher average achievement completion.

4. How might engagement with games change over time? (Game data by release date). We can do this by looking at how achievement completion and playtime change throughout the years as trends over time in a graph.
   (GAMES_1 & ACHIEVEMENT_PERCENTAGES)
   - We found that there is no change over time of engagement with games based on just their release dates.

Motivation and Background:

How players perceive games influences how they perform in the market. Analysis of player data could potentially reveal how multiple factors influence a game's reception and activity. If common trends can be identified, we could see more clearly how to make a successful game. By analyzing player data that is strictly quantitative we can also see if there are any higher-level trends in the market. It could be possible to measure success based solely on numbers instead of relying on player reviews or sentiments, and this data could dictate how a game publisher or development company acts towards releases and features.

Dataset:

Steam Dataset
This dataset has data scraped from Steam services for someone's conference paper. The data was taken in 2016. It is not the most recent data but it is comprehensive.

Methodology:

Filter Data:

We downloaded and decompressed the data archive, then imported it into a local MySQL server. The database exceeds 100 GB in size, and is far too big to process directly in Python. We used SQL queries to randomly sample games, and used the result to further distill data in other tables. This way we get a fairly distributed dataset that is small enough for us to work on it.

A number of aggregation queries on the dataset were implemented and ran in MySQL instead of Python pandas, because MySQL can better deal with gigantic tables. One of the game activity tables contain more than 300 million rows, another contains more than 700 million.

For the ML model:

1. Using both ACHIEVEMENT_PERCENTAGES and APP_ID_INFO.
2. Clean up the database down to what we need for this ML and drop all missing values.
3. Split the dataset into 70% training data and 30% test data.
4. Use the model to make predictions and create visualizations and perform further analysis.
5. Separate the data into usable features(Type, Price, Is_Multiplayer) and labels(Rating).
6. Use a decision tree regressor to train the training data and test it with test data.
7. Come up with a square mean error to see how it did.

For Graphs:

1. multiplayer content v.s. Play Time

Using both GAMES_1 & ACHIEVEMENT_PERCENTAGES.
Clean up the database down to what we need for this graph and drop all missing values.
Plot a scatter plot on Is_Multiplayer and the sum of playtime_forever for that game.
Given it a title and axis names
Analyzing the result by answering the research question.

2. engagement change in multiplayer vs single-player titles
    Look at other metrics besides simply playtime or rating
    Filter data to look at achievement percentages as game completion
    Plot both by year, side by side to see how achievement completion changes over time for multiplayer and singleplayer games

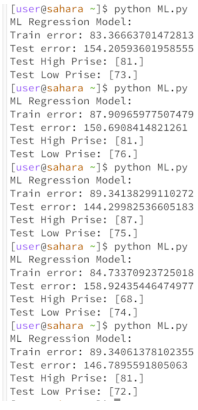
3. Engagement vs Play Time
    Using both GAMES_1 & ACHIEVEMENT_PERCENTAGES.
    Filter the database down to achievement completion, total playtime, and year
    Plot 2 graphs comparing how averages for each change based on the release date
    Add a title and axis names
    Analyze the result by answering the research question.

Results:

1. ML model

Our ML model was not really successful. We found only a slight correlation between our selected features and the rating of a game, so the model can't accurately predict rating based on a given feature.

We used a DecisionTreeRegressor to complete this ML. We used 70% of our dataset for training and 30% of our dataset for testing. After the model is trained, the mean square error for the training set is around 85, and the mean square error for the testing set is around 170. Both of them are high. One possible explanation is that even if a game has the same price, release month, and multiplayer content, the rating still can vary a lot. We imagine that if it is possible to get "game quality" data, the ML will do better. But quality can be hard to measure.

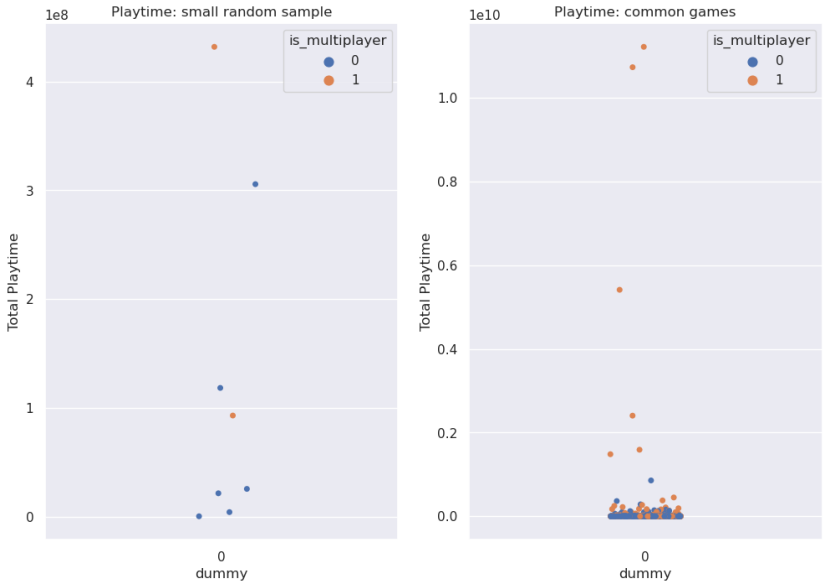


Since the rating data in the dataset are "Metacritic scores" computed by gaming blogs and media, we used another formula from SteamDB, a well-known Steam analytics website, to compute a score on the same scale for user-generated review rating. The model did not perform well on user rating either.

Before we wrote the ML, we thought that price might have a big impact on the rating, but the result turns out unexpected. We tested a high price($100) and a low price($0) model with every other feature the same, and get a predicted rating of high price around 80 and low price around 75. The difference is insignificant, which overthrew our expectations. The fact that a regression model could not predict a game's rating shows that a game's rating does not depend on price, release time, etc., as much as it depends on the quality of the game itself.

2. Multiplayer vs Playtime

In the below graphs, blue dots represent single-player games, and orange dots represent multiplayer games. We found that multiplayer content does, in general, have higher playtime.



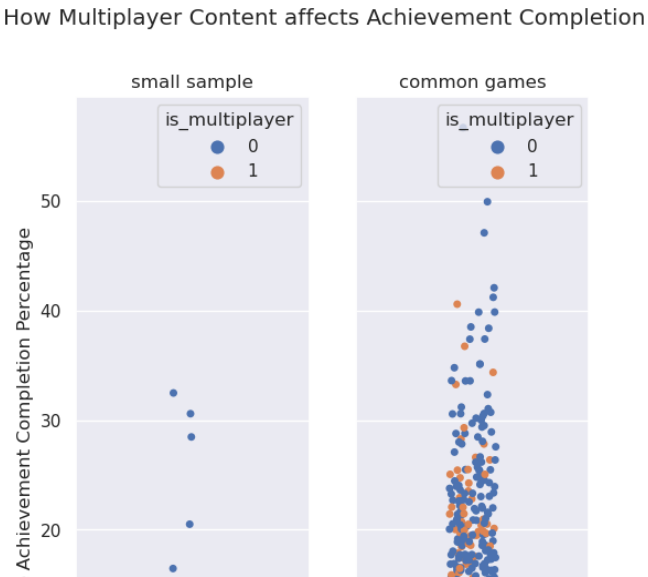
By interpreting those two graphs, we can see that for the small sample, the playtime difference between multiplayer and single-player is not that clear. But for the large sample one, we can see that the orange dot(multiplayer games) is stacking on top of the blue dot(single-player games). Also, all the extremely long playtime is mostly orange.

The result is not surprising. We expect that multiplayer games will have more playtime in general. Single-player games tend to have a plot and after the plot is over, players are not likely to replay it too many times. Whereas multiplayer games are more likely to have less to no plot but more competitions between players, so players can play endlessly to get props.

3. Multiplayer vs Achievement Completion

Whether or not a game contains multiplayer elements does impact how players interact with achievements. Specifically, single-player games have higher achievement completion rates compared to multiplayer ones.

The graph shows a small random sample of games. The single-player games (blue dots) all have higher average achievement completion than



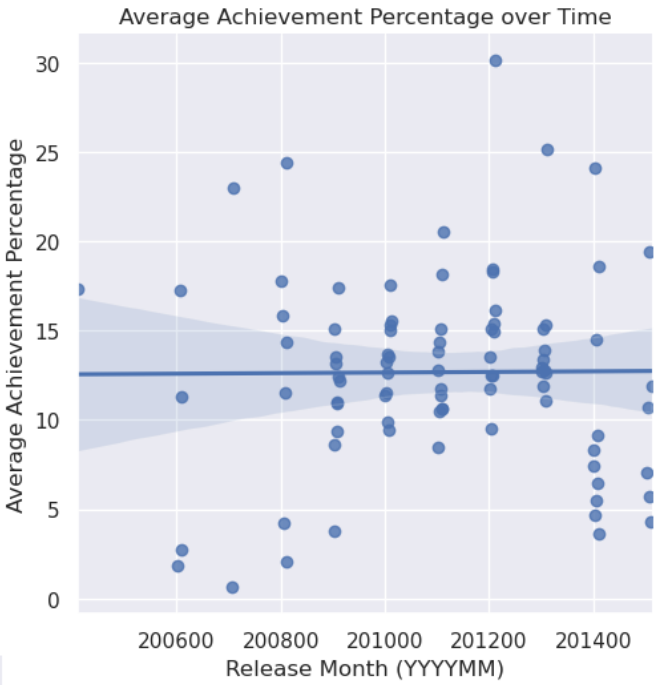
the multiplayer ones (orange dots). This trend is slightly less clear when considering a set of common games, with many more games included in it. However, it is still visible that the top half contains more blue than orange.

Does this reflect game design, or how players play multiplayer vs single-player games? On one hand, single-player games may be designed to have more accessible achievements. Multiplayer games may have achievements that encourage players to spend larger amounts of time, or put forward a greater commitment in order to get achievements. Multiplayer games could also be generally larger and have more achievements overall. This would also lead to a lower completion rate.

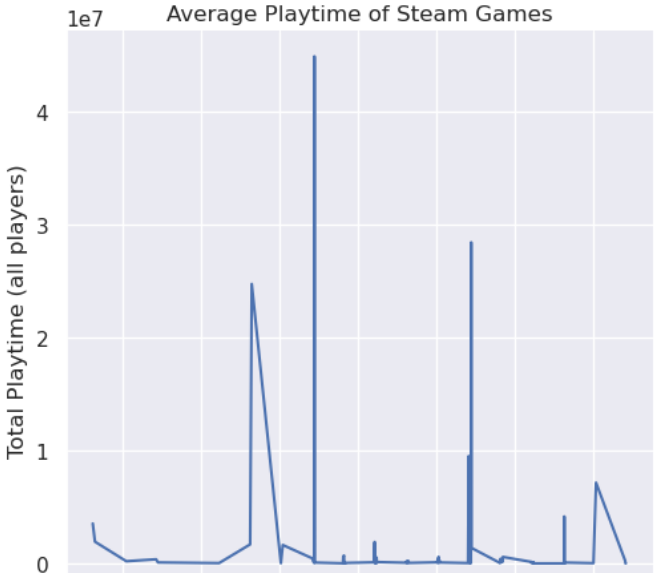
On the other hand, players of single-player games might be more incentivized to play through an entire game and complete all of its achievements. There could be more success to completing a game that can be completed, like a single-player title, than to completing achievements for a game that has no definite end, like a multiplayer title.

4. Playtime and Achievement Completion over Time (release date)

We considered playtime and achievement completion as metrics for player engagement. We found that by these metrics, there has been no real change throughout times between games.



First, by looking at the achievement completion, we see that a scatter of the points of games with achievements seems random. A regression line applied to this arrangement of points, as is visible on the graph, is completely flat. This shows that there is no trend in achievement completion based on release date. Looking at the average playtime of games



compared to their release dates, there is also no trend. It does not seem that there could be a trend of games getting more engaging, or being played more. What does stand out are the few games that massively disrupt the average values because they have been played significantly more. A cursory inspection of the

data reveals that some games have been played for thousands of times more hours than others. These massive outliers also make it more difficult to see how the data trends on a smaller scale.

These two graphs reveal two things. First, the release date of a game has little effect on how it performs. It is not possible to tell how a game interacted with the Steam environment from these metrics alone. Second, these graphs show that there has been no trend in the Steam market. Developers are not making games that give more incentive to complete achievements, nor are developers making games that players are playing more and more.

Or are they? It would seem reasonable that with the growth of the gaming industry, playtime measurements should just trend upwards. This is sensible but is not visible in the data because while newer games might have more players, older games have had more time to be played, and there is no change. Perhaps a trend could be clearer with more recent data, or further into the future.

Challenge Goals:

Our planned challenge goals were:

> ***Messy data****: The dataset is not in .csv format, instead it is in .sql. The data also has many different categories and fields of data that were gathered. We will need to sort through the data to find the data we need, and then convert it to a format that we can work with. We think this will be an achievable goal because the data is already somewhat organized, we simply need to find what we need.*
> ***Machine Learning****: It seems likely that a machine learning model could help predict how the Steam market and provide useful information about the games on it and how different aspects of games interact to influence each of the statistics that we examine. This goal could be more difficult but the dataset is quite large and that means there will be a good deal of data to train a model on.*

For messy data, this was definitely a challenge. Not only was the original dataset very large (multiple gigabytes), but not all the data that we needed was in the same place. We had to filter the data down to what we needed to work with, in a reasonable size to work with. We began with examining one hundred random games taken as a sample, but this was not enough. We ended up working with around one thousand games selected randomly from a pool of more common “active” games. We also had to deal with missing data. For achievement completion, many games did not have achievements so had NaN values for completion percentage. There were also edge cases. In one instance, a game had an invalid time value, so was listed as the start of time (1970-01-01). Even after all the data was cleaned and prepared, we had to convert time values to something that plotting libraries could use for the investigations into change over time.

Machine Learning was an interesting goal. While we originally thought that it would be possible to predict how a game performs based on data about other games, we saw that the features we selected had little correlation to how a game was rated. We tested with 100 games at first, then 1000, but our model's mean-squared error was always about 80. Other variations, such as adding release month as a feature, or switching to user score as a label caused the model to have an even higher error. So while we thought that with this amount of data we could train a model well, there does not seem to be a correlation between how a game is rated and its features in the market.

We also wrote a web scraping script with BeautifulSoup to get the user score mentioned in the previous paragraph. It was very basic but should be mentioned because it was part of another challenge goal, learning a new library.

Work Plan:

Our original work plan was:

> *Filter data: 1-2 hours*
> *Analyze and plot data: 2-3 hours*
> *Create ML model - 3 hours*
> *Write report:- 1-2 hours*
>
> *After the database is filtered, the information we need will mostly be set, so we won't spend too much time cleaning the database afterward. We can then each tackle an ML or plot question from above and come up with the initial solution. If time allows, we will switch the question we get and the initial solution to another group member for improvement. Our code will be written collaboratively in an Ed workspace in multiple .py scripts. Each file will be a script aiming to answer one of our research questions.*

Work Plan Evaluation:

Filter data:

The filter data part is mostly accurate since it was done before the work plan was created. We did spend a little more time on getting the data we want from the huge dataset and other places to better fit our research questions.

Analyze and plot data:

Our estimate was not so good for this part. We spent much more time analyzing and plotting data than expected. We encountered problems like panda's groupby method returning Series which matplotlib and Seaborn did not like, data frame returning empty after certain operations, and plots not updating after running the code. We also found our first attempt to look at achievement data was unsuccessful since there were too many none values. We ended up returning to the data filtering part to get more valid data. Then getting the graphs correct, with

informative labeling and juggling between matplotlib and Seaborn graphs was also time consuming. In total, this step took 5-6 hours instead of the 2-3 we forecasted.

Create ML model:

The code for the ML model is fairly simple, partly due to the good organization of the data. However, we spent a lot of time trying to see which features were better to use in the model. We also tried rating datas from another source to see if there could have been something wrong with the rating data in our original set since the model's error turned out to be high. Overall the time consumed is close to our prediction.

Write report:

Writing the report was more time consuming that predicted. While the actual process was simple, going back and forth between the code and the report was tedious. This was more of a problem while the code portion was unfinished, afterwards it was easier to put all the information down into this report. The report ended up having a lot more parts than previous steps as well. We spent approximately 4 hours on the report.

Testing:

Our code was not algorithmic in nature. We did not implement much functionality; we just interacted with a large volume of data and attempted to analyze it and identify trends via visualizations and other methods. We did try to test the code on a smaller dataset (the previously mentioned 100 games), then went larger to more clearly identify trends in the data. We have manually examined the cleaned data before plotting them to make sure there is nothing wrong with it. For this reason most of our testing was done with print statements to verify that code behaved correctly, and objects were consistent in their types and their content. Otherwise, the results we found were reliant on code from well document libraries. There are still some testing lines in the comments of our code.

Collaboration:

We did use online resources to help us. We referred back to the seaborn documentation to plot our graphs in an appropriate way. The matplotlib documentation and other library documentation was helpful as well.
This video on scraping the web with BeautifulSoup and the requests libraries was helpful for us to get more rating information on web: https://www.youtube.com/watch?v=ng2o98k983k