Capstone 2 Final Report: Predicting Video Game Success

Mike Stacey

Problem Statement

The video game industry in Canada is a multi-billion dollar industry that is growing in recent years. Sales of video game hardware are going up and both adults and teens are turning to gaming to help with the mental health issues being caused by pandemic-related isolation. As this market grows, so does the incentive for video game companies to capitalize on the increased demand. As games cover a wide range of genres and platforms, I wanted to see if I could find a way to predict what contributes to player enjoyment and determine if a game would be popular or not.

By using data from over 400,000 video games, I created a model to predict the popularity of a game using the average number of users who suggested a game as a target for popularity. I decided to use a classification model to determine if a game would receive above the average number of suggestions or not. My RandomForest classification model was able to predict above average games with 78.46% precision and was able to predict games that were equal to or below average with 92.81% recall.

Data Wrangling

The dataset came from the RAWG game database. It included multiple columns that could be used to determine player enjoyment as well as columns that would be features such as the platform of the game, the genre and release year.

I began by dropping irrelevant columns such as the website of the game and extra identifying numbers. I also dropped columns that were missing too many entries to be useful as well as any games that had not been released yet. I felt that I could not impute missing data without impacting the results and given the sheer size of the dataset, I felt comfortable dropping rows that were missing values.

Nearly all of the games had zero achievements so the number of achievements was not useful. As such I converted that column to a boolean value of whether or not there were any achievements. I did the same for whether or not a game was part of a series for the same reason.

I now was left with the name of the game, the release year, how many suggestions it received, the platforms it was available on, the developers behind the game, the genres the game fit under, and whether or not it had achievements and was part of a series. The platform and genre columns were strings containing a list of each applicable value, so I had to explode those columns so that each value would be separate. I then created dummy variables for them so I could use them as numeric features. Finally I dropped the name and developer columns as the name of the game is only relevant if it is part of a series and the developer column is not something that can be changed and thus is not relevant. I now had all numeric features and one target to classify based on.

Exploratory Data Analysis

I began by looking at how common achievements and multi-game series were. 4.6% of games have achievements and less than 1% of games are part of a series.

I then looked at how many games were released each year. With a few exceptions, more games are released each year than the previous one. I plotted this with a histogram.

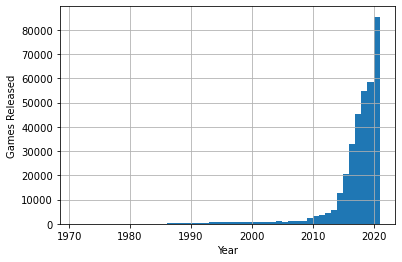


Figure 1: Total number of games released per year

I also wanted to see if there was a pattern to the average number of suggestions each game released in a given year was. Plotting this data in a line graph shows that the number of suggestions per game peaked in 2007 with spikes in 2021 and the early 1970s. These spikes are explained by the small number of games released in those years that are included in the dataset.

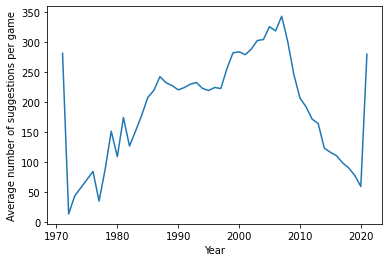


Figure 2: Average number of suggestions per game each year

I then determined the average number of suggestions for games with and without achievements and games that are and are not in a series. Games with achievements received an average of 313 suggestions compared to 89 for games without achievements. Games that are part of a series received an average of 383 suggestions compared to 98 for stand-alone games.

Finally I generated a heatmap to see which features appear to be correlated. Judging by the heatmap, the features most positively correlated with suggestions are games released on the PC, indie games, and games with achievements. Web-based games and the release year are the most negatively correlated with suggestions. I also see that platforms that were available at around the same time are positively correlated with each other, suggesting many games were released on multiple platforms. The heatmap is too large to include here and be readable, but it is included in the EDA jupyter notebook.

Preprocessing and Modelling

I prepared to run the models by converting the suggestions column to a categorical variable of above average suggestions or equal and below average suggestions. I then split the data into 70/30 training/test datasets and used a standard scaler to scale the data.

My first two models were entropy and gini impurity decision trees with the default parameters. I printed out the maximum depth of each tree and used grid search cross-validation to test each potential maximum depth up to that number to find the optimum depth for the trees. I ran the first two decision trees again but used the optimum maximum depth. Both models outperformed the basic versions in every metric except for recall score when predicting that a game will not be above average.

I then decided to see if a random forest classifier could outperform the decision trees. I ran a model with default parameters to determine the range of max depth to check and then tried a grid search to check various parameters of max depth, number of estimators, gini or entropy criterion, and whether or not to bootstrap the model. However due to the size of the dataset, this would have taken several days to run so I stopped the model and instead used a randomized search to optimize the parameters. The optimized model outperformed the basic model in every metric, however it was beaten in a few metrics by other models so I needed to decide which metrics would be most important.

In a real world situation, when predicting that a game will have above average suggestions, false positives would lead to costly investment in a game that might not be popular enough to make back the money. A false negative in this situation would mean that money would not be invested in a particular game that is likely to be successful, however the company would likely choose a different game idea that is identified as likely to succeed, and thus would still make a profit. Therefore it is important to avoid false positives when predicting success and inversely avoid false negatives when predicting failure. Using this logic, the most important metrics are precision for above average games and recall for equal or below average games. For this reason I selected the optimized random forest model as it performed the best of all the models in these metrics and it was also the most accurate.

Takeaways

In order to understand what the model was predicting, I defined a function to output a bar graph of the ten most important features used to make the predictions. In order to determine if the features had a positive or negative impact, I then printed out the value counts of games above and equal or below the average number of predictions when the values of those features were true. I did not do this for the release year as past releases are not something that can be controlled in the future.

From this it appears that web, PC, and iOS based games are more likely to be unsuccessful than successful, which could mean that console games could be a safer investment. It also appears that games with achievements are overwhelmingly more likely to be popular.

As for the genre, casual and RPG games appear more likely to be popular than not, while adventure games are slightly more likely to be unpopular but are close to evenly split. Platformers are far more likely to be unpopular, suggesting that that genre is a risky investment.

Lastly, I looked at indie games. A game developer is either independent or not, so this feature cannot be controlled by the developer. However as the data shows independent games to be far more likely to be popular than unpopular, this could indicate a useful topic of future study. Clearly the independent developers are doing something correctly and it could be useful to the larger developers to learn what that is so it can be incorporated into mainstream games.

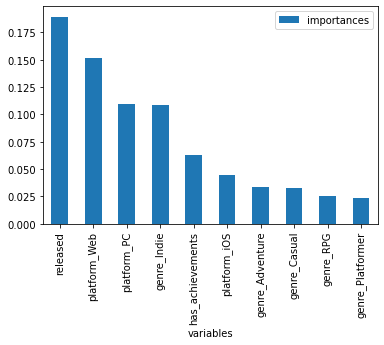


Figure 3: Ten most important features for RandomForest model