Brooke Ann Coco
Project Luther Summary
Predicting Tabletop Game Success
April 22, 2019

Overview:

Despite the growing trend towards digital singularity, tabletop games have seen unprecedented success. In 2017 alone, more than 5,000 board games were introduced into the U.S. market.[1] Given the growing success of tabletop games within recent years, this project seeks to construct a linear regression model that predicts the success of tabletop games. The application of this project lies with tabletop game designers and publishers alike, who can use this model to produce games that are optimized towards consumer desires.

Data:

The data was obtained via scraping information from boardgamegeek.com (BGG), a massive online games database with information and metrics on over 81,000 tabletop games. The site also contains an online forum where board game enthusiasts can rate, review, and discuss tabletop games. For this project, I scraped information for 8,712 tabletop games. After cleaning the data to account for missing data and outliers, I was left with 5,537 observations.

Because tabletop game sales were not readily available to the public, I utilized average community rating as my target variable. Though the BGG community is not a perfect representation of the general public nor is average rating a perfect indicator of game success, it is the closest available proxy in the absence of sales data.

Tools:

Platform	Purpose	Tool
Python	Web Scraping	BeautifulSoup4
		Selenium
	Cleaning/ Feature Engineering	Pandas
		Numpy
	Modeling	Scikit-learn
		Statsmodels
	Visualization	Matplotlib
		Seaborn
Powerpoint		

[1] Oliver, Marie Elizabeth. “Join the Party: Board Game Popularity Just Keeps Growing | Produced by Advertising Publications.” *The Seattle Times*, The Seattle Times Company, 14 Feb. 2018, www.seattletimes.com/explore/shop-northwest/join-the-party-board-game-popularity-just-keeps-growing/.

Modeling/Feature Engineering:
This project focused on the use of linear regression to predict tabletop game success. The models were evaluated using 5-fold cross-validation with r^2 as the primary scoring metric. R^2 measures the amount of variability that any given model accounts for. Because the goal of this project was to predict BGG ratings, it was important that my model account for as much variability as possible (without overfitting) to yield more accurate predictions.

My MVP model received an r^2 score of 0.412, which indicated that I was likely underfitting the dataset. As a result, I did not need to utilize regularization methods, but instead, needed to add more features to my model. However, I first took a closer look at the features in which I was using to predict BGG ratings.

Upon further inspection of what qualified as an expansion on BGG, I discovered that this metric was much more expansive than I had initially imagined. For instance, depending on the game, an expansion may refer to either the addition of a single card, the addition of new player pieces, a new game that relies on the base pack, or even unofficial 'fan expansions'. In my mind, it was unfair to quantify the addition of a new player piece as equivalent to the reimagining of a whole new game. For this reason, I decided to remove the number of expansions as a metric in my model.

The next step towards improving the predictability of my model was to clean the data further. For one, this meant accounting for outliers in the data. For instance, because average rating was my target variable, I filtered out games who had less than 100 community ratings, to lessen the chance of outliers skewing the average. I also decided to cap the maximum number of players at 12 and the release year at 1950, as I suspected that the more extreme numbers in these features would have little to no effect on predicting the success of a game. However, instead of removing these extreme values from the dataset, I combined them with the capped values. Finally, before retraining my model, I scaled the data. As a result of cleaning and scaling the data, my model's r^2 score jumped up to 0.5191 from 0.412. Not bad!

Next, I looked to feature engineering to improve the predictability of my model. First, I added more features by converting game genre (a categorical variable) into dummy variables. This boosted the model's r^2 score to 0.5548. After that, I played with several interaction features, most of which were unsuccessful at improving the predictability of my model. The few that did improve the predictability of my model, had multicollinearity issues. From there, I tested the use of log and square transformations on a few of my target variables. Nearly all of these feature transformations failed to improve the predictability of my model.

Apart from adding genre to my model, the only bit of feature engineering that improved my model was adding the log of the difference between maximum and minimum playtime. Adding this feature while removing some of the less significant genre features (as determined by correlation to ratings) brought the model's r^2 score up to 0.56734 on my training set. Finally, I retrained the model and fit it to my test set, which yielded an r^2 score of 0.5660.

Conclusions:
Given the highly subjective nature of both my target variable (average community rating) and one feature variable (complexity), I was fairly happy to have constructed a model that was able to account for nearly 57% of the variation.

A review of the final model's standardized coefficients reveal that the log of the difference in playtime (coef = -0.401) appeared to have the most predictive of success. More concretely, with every increase of 0.418 (one standard deviation) in the log of the difference in playtime, the average community rating of a game increases by 0.044, or the standardized coefficient of the log of the difference in playtime (0.401) multiplied by the standard deviation of ratings (0.950), assuming that all other feature variables remain constant.

Though standardized coefficients make it difficult to interpret categorical dummy variables, they are particularly useful when trying to illustrate the relative importance of each variable in a regression model. As such, the scaled coefficients indicate that whether a tabletop game is classified as a children's game (coef= -0.341) has the second most predictive power in the model, followed by whether a tabletop game is classified as a wargame (coef= -0.198) as well as the complexity rating of a game (coef=0.130). In summary, the BGG community appears to enjoy fairly complex games that have a somewhat large variability in the time it takes to complete. In terms of genre, the most highly rated games tend not to be categorized as either children's games or wargames.

Perhaps the most interesting result of this study was illustrated by the residual error plots of my MVP and the final model. Upon inspecting the residual errors of my MVP model, the data points converged into two rough clouds. After finalizing the model, the separation between these clouds became much more pronounced, indicating that my model is missing an important piece of information. Upon locating the source of this divide, I would likely be able to significantly improve the predictability of my model.

Future Improvements:

Future improvements of this project would include scraping more data. This would include scraping not only more observations, but also more features. For instance, I would have liked to have taken factors such as country of origin (America, Europe, Asia, other), the number of games already released by the creator and the publisher, as well as the type of game (board, card, or dice) into consideration. With more time, I would be able to more deeply navigate BGG as well as other sources of information.

Next, I would have liked to delve into different transformations of my feature variables. Though I played with a few log and square transformations, most of my time feature engineering was spent exploring feature interactions. In the future, I would like to examine correlation plots more carefully to determine better transformations of the data.

As discussed in the **Conclusion** section, given more time, I would also like to identify the source of the two residual clouds. Clearly my final model is missing a variable that significantly separates portions of my dataset. With further examination, I may discover the source of this variation and further improve the predictive power of my model.

Finally, I would like to investigate whether my dataset meets the assumptions required to apply linear regression models. It is entirely possible, in fact likely, that my dataset does not meet these assumptions, meaning that a linear regression may not be the most accurate method through which to predict success. Upon future work with this data, I would like to ensure that my data meets these basic assumptions.

Overall, my knowledge of linear regression, project organization, and workflow has increased tremendously throughout the course of this project. Though I struggled a bit with grasping which steps should be taken in which order, the whole process has become much more clear in my mind. If I were to complete this project again, my workflow would be much more stream-lined, leaving me more time to focus on improving my model.