MGT 6203 - Team 56 Final Report

My Duong; Anthea Mitchell; Stephanie Yie; Jose Santana

April 16, 2023

Contents

1 Introduction 1

2 Methodology & Innovations 1
- 2.1 Data Collection & EDA 1
- 2.2 Comparing Regression Models 2
- 2.3 Recommender System 2
  - 2.3.1 Content-Based Filtering 2
  - 2.3.2 Collaborative Filtering 3
- 2.4 Market Segmentation 4
  - 2.4.1 Text Clustering Approach 4
  - 2.4.2 Numeric Clustering Approach 4

3 Experiments & Evaluation 5
- 3.1 Regression Models 5
- 3.2 Market Segmentation 7
  - 3.2.1 Text Clustering Approach 7
  - 3.2.2 Numeric Clustering Approach 8
- 3.3 Recommender System 9
  - 3.3.1 Content-based Filtering 9
  - 3.3.2 Collaborative Filtering 9

4 Conclusion & Discussion 10

References 11

Appendices 12

A Appendix A 12

B Appendix B 12

C Appendix C 13

D Appendix D 13

E Appendix E 14

F Appendix F 14

1 Introduction

The market for board games was worth an estimated USD 13.1 billion in 2019 and is predicted to continue expanding.(See20, ee20) The market for board games is also predicted to reach USD 30 billion by 2026 as a result of growing acceptability and appeal among consumers despite competition from digital entertainment sources.(Mor22, or22) According to a survey done by PrintNinja, 41% of current boardgamers purchased 5-10 new games or expansions in the past year alone. Due to the ever-expanding board game market making it difficult to sift through all the games, how do these gamers find their next favorite board game? The top three ways consumers hear about new games are through word-of-mouth and browsing social media sites such as BoardGameGeek (BGG).(LW21, W21)

Our project focuses on identifying the characteristics and user statistics that most accurately describe a successful and well-liked board game by analyzing a large dataset of board games. As demonstrated by Netflix, YouTube, and other online public platforms, recommender algorithms have significantly improved user experience. When consumers utilize the recommender system, we want to concentrate on creating a fantastic user experience and on providing individualized recommendations to each and every user. Our identified characteristics and recommender system would hopefully boost the sales of board game creators by 10% by exposing their games to a larger audience overall in the board game industry. Hasbro, one of the largest and most profitable board and card game companies in the United States, profits most from a card and tabletop RPG game. This segment alone produced $419.8 million in revenue in 2022, meaning a 10% increase in only this sector would result in a $41.9 million advantage for that company.(Mil12, il12) We hope the results of our project can be used by current board game creators and marketers as recommendations to bring in new customers and increase demand and interest in their games.

2 Methodology & Innovations

2.1 Data Collection & EDA

There are two datasets being used for this project; both are from Kaggle and were originally scraped from BGG website. The core dataset contains 81 attributes and ratings for more than 94.000 board games and expansion. It was cleaned by filtering out any columns with more than 10% missing data. The game type was fixed at "board game" to filter out any expansions or minor components. Multiple columns: *game.type*, *details.image*, *details.thumbnail*, *stats.median*, *stats.bayesaverage* and *stats.stddev* were removed since they either provide unimportant information or statistical data of average rating which is already included in the final dataset. The remaining columns were cleaned to remove any outliers by examining each column distribution. The second dataset has all ratings for all BGG board game IDs with username.

There are over 411,000 unique users and more than 19 million ratings. This dataset is solely used to build collaborative filtering (CF) recommender system.

In our exploratory data analysis (see appendix A), we found that the highest count of user-ratings (67,655) and most commented (13,841) game on BGG was Catan. The game with the most interest is Terra Mystica with 10,920 BGG users having it on their wishlist. The highest-rated game overall was Pearl Lands (9.83), the two highest-rated games with more than 100 user-ratings was Mythic Battles: Pantheon (9.34) and Gloomhaven (9.14), respectively. Additionally, looking at the relationship of all numerical variables, we observe that there are two groups of variables that are strongly correlated with each other. The first group of variables represents the popularity and demand of games consisting of wishing, wanting, trading, owned, number of comments, etc. The second group consists of variables that represent the characteristics of games like average rating, weighted average, total attributes, etc.

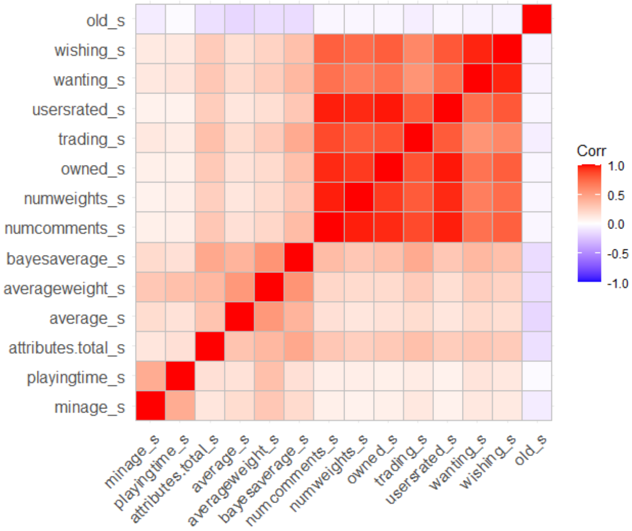


Figure 1: Correlation Matrix

2.2 Comparing Regression Models

When looking at factors that are most important for a board game, we used a linear regression model to predict the demand for board games. First, factors that had no impact on our predictions and redundant variables were removed. Other cleaning steps include renaming factors and consolidating the *wanting* and *wishing* factors into a single comprehensive factor, which we named *wishlist*, this would serve as our response variable since a *wishlist* is a good indicator for a user's interest. The board game *description* factor was dropped after being transformed from a string into a numerical sentiment value using NLP's sentiment analysis package, which provides amalgam scores for a game description based on the words used within. Despite the dataset only containing two categorical variables, converting them into dummy variables as is would have resulted in 10,000 variables, making it difficult to interpret. Therefore, we took several steps to transform and organize the data into a 32-factor dataset including dummy variables for our categorical data.

To simplify the analysis, board game categories were combined into 10 broader categories and were then condensed and made into dummy variables. Similarly, for board game publishers, there were an overwhelming number of unique publishers, exceeding 10,000. To simplify this, publishers were grouped into publisher sizes based on the percentage of board games produced by each company. The condensed publishers were also turned into dummy variables. Finally, to represent the game's published year, we calculated the age of each game and converted it into a continuous variable, capturing the time-related property in a way that is relevant to our model.

Next, we trained different models in order to get a general sense of how our linear regression compared to other regression models in terms of fit. Regression models such as: nonlinear models, nu and epsilon SVM regression models from the e1071 SVM package in R, and a random forest model with 3 variables at each split and 500 trees. Finally, we trained a linear regression model to derive a linear equation that could be used to make predictions on a sample dataset.

2.3 Recommender System

Content-based filtering and collaborative filtering approaches are utilized in this project in designing the board game recommender system. Profile attributes are the foundation for content-based filtering. However, collaborative filtering solely uses past interactions and does not take any attributes or item qualities into account.

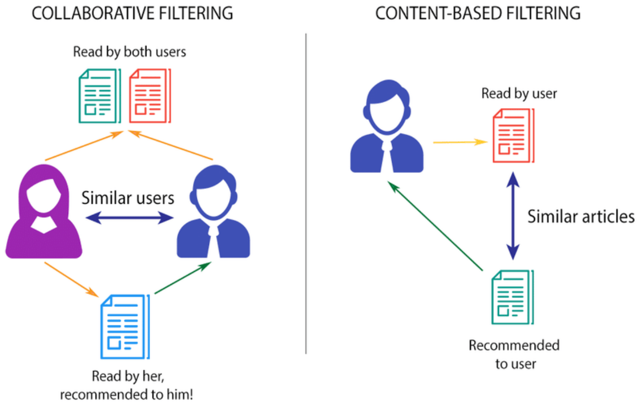


Figure 2: Collaborative vs Content-based Filtering

2.3.1 Content-Based Filtering

To create the recommender system using Content-Based Filtering, only the top 8000 games with highest average ratings and number of ratings were used. Term frequency-inverse document frequency was used (TF-IDF) to process the text data present in multiple columns our dataset, such as game *publishers*, *name*, and *game description*. For each data point, the text data was combined into a main document string and was pre-processed by stripping punctuation and tokenized using a lemma tokenizer to reduce the number of synonyms. Next, the stop-word corpus was updated to include any fragmental words resulting from lemmatizing the documents. The result of our transformation is a sparse matrix for each game to record its normalized term frequency.

The cosine similarity index calculates how similar two vectors in an inner product space are to one another. It establishes whether two vectors are roughly pointing in the same direction by calculating the cosine of the angle between them. In text analysis, it is frequently used to gauge document similarity. (HPYM04, PYM04) Since we recommend board games based on their mentioned combined document text, comparing their cosine similarity of each game with chosen games can help us rank how these games match a specific user's choices. After computing cosine similarity to for each game to each of other games. We can set weights for a user choice to represent the magnitude of each of that user choices. The higher the weight the more relevant of that choice to our final recommendation. The equation to compute final cosine similarity for a recommended game is illustrated below:

$$Best\ reccomended\ game\ similarity = \frac{\sum_{i=1}^{number\ of chosen\ games} S_{highest} * w_i}{number\ of chosen\ games}$$

Figure 3: Best Recommended Game Similarity Formula

Where $\mathrm{S}_{\mathrm{highest}}$ is the highest similarity by computing cosine similarity calculations on all games with a specific game chosen by user, and w_i is the weight applied to our chosen game's similarities with any other games. By computing the cosine similarity of each game, we can create a content-based application based on their chosen games and customizable weights.

To experiment with different way to compute similarity between two game, different distance metric can be used such as: Manhattan distance. The results are discussed in detail in the experiments section.

2.3.2 Collaborative Filtering

Unlike content-based filtering, the collaborative filtering technique organizes users based on their past usage patterns or preferences, and suggests products that has been been seen or liked from users belonging in the same group. The age of the users, the game's genre, or any other information about persons or products are not considered while calculating the similarity. It is exclusively dependent on the implicit or explicit rating a user provide for a product. Therefore, only the second dataset is needed for this model. There are many collaborative filtering methods, but memory-based and model-based are 2 main types. Memory-based methods use user rating historical data to compute the similarity between users (UBCF) or items (IBCF), while model-based methods use machine learning algorithms to predict users' rating of unrated items. The most frequent model-based approaches are matrix factorization models, such as using an SVD to reconstruct the rating matrix. We will compute all 3 models (UBCF, IBCF annd SVD) to make comparison on their performances based on root-mean-squared error (RMSE) and machine learning run time.

The first step is to prepare the dataset. Since the original user-rating dataset is huge (19M rows) and even much larger after we comprise a user-item matrix. It would be too large relative to the available computing resources. We decided to only include games with rating of 7 or higher that had at least 52 ratings with users who had provided at least 5 ratings. The cleaned dataset has about 100,000 ratings. Next, the dataset was converted into a sparse matrix called 'realRatingMatrix' which is about 9 times more efficient in conserving memory than a traditional matrix object.

With the realRatingMatrix in place, we can now define parameters that will be used by the recommender algorithm to train the model. The technique is configured to use a single test dataset and train on an 80% random sample of the data. The recommender algorithm is given 4 items per user in the test set, while the remaining test user's items will be used to compute rating prediction error. The model is built using the train data. There is also test data accessible for examination, both *known* and *unknown*. Instead of forecasting test performance, these *known* records are utilized to calibrate the test user's similarity to the trained records, identify and weight its nearest k neighbors, and then create item ratings or recommendation predictions. The anticipated ratings or recommended items from these *known* data points are compared to each test user's remaining hidden items. As a result, these *unknown* test user items will be used to determine the model's prediction error. It is important that the given parameter is smaller than the minimal number of rated items available per user, so that *unknown* test data is available for each test case to quantify rating prediction error.

For memory-based methods, commonly used similarity metrics include the Pearson correlation, the Jaccard similarity coefficient, and the cosine similarity. The first two are not good options if using unary ratings but work well for this scenario. For this model, we use cosine similarity as a metric.

2.4 Market Segmentation

2.4.1 Text Clustering Approach

To perform unsupervised learning on our board games, we need to further clean our data by filtering any game with less than 500 reviews since the distribution of number of ratings per game is heavily skewed towards low number of ratings. This is because most games in this dataset are not rated by many users. Thus, if we include these games in our clustering analysis, it will skew our cluster toward less popular game which is on the contrary with our intention. TF-IDF transformation was then applied to our combined string again with a similar manner mentioned in section **2.3.1** to generate required features for clustering. Since the number of features is more than 18,0000 due to sparsity of our matrix, latent semantic analysis (LSA) was applied. Following the construction of the occurrence matrix, LSA finds a low-rank approximation to the term-document matrix. (MM12, M12) The low-rank matrix was generated using single value decomposition with 200 components. Figure 4 illustrates the reason for choosing such a number.

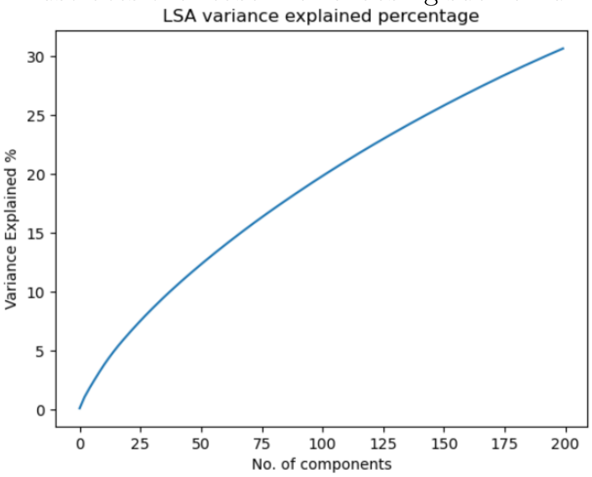


Figure 4: LSA Explained Variance Percentage

As we can see with 200 components, our explained variance is not increasing significantly, an indication of most variance of data is capture with those components.

The LSA decomposition yields the feature space required for our k-means and mini-batch k-means clustering methods. These centroids are estimated with mean of distance metrics at first with their respective nearest neighbors K-means method clusters feature space based on distance of their respective centroids. It is an iterative method where the initial position of centroid is continuously computed until optimal cluster is defined. Usually, inertia and silhouette metrics are considered to determine if a k-mean algorithm has successfully quantized our feature space. These metrics will be discussed in the experiment section. Mini-batch k-means is a variation of k-means that utilizes small batch of data during fitting. It focuses on improving model fit time and convergence.

Both k-means and mini-batch k-means method were used for market segmentation to divide our board games into sub-groups which will be ranked based on user average ratings and popularity. Our game publisher, name and description will be clustered into different niche markets with different key words to describe their thematic board game representation. Board game producers then can use these clusters to decide if a theme is likely to succeed in the market and gain popularity with the fan base.

2.4.2 Numeric Clustering Approach

To take advantage of the continuous values in our data, a second approach for market segmentation was applied that used principal component analysis (PCA) to reduce factors into two dimensions and cluster them to gain more insight. The cleaning steps differed from other modelings in that columns with 25% or more missing values were eliminated, as were all categorical variables to focus on the numeric factors, leaving 76,685 observations and 20 variables. Moreover, additional imputation of data was required to maximize and simplify important attributes. Rows with all missing values in *max/min players*, *min age*, and *max/min playing time* were removed, and from the remaining observations, the variable *players* was created. The *players* variable was set equal to the value of *minimum players*, the variable with the fewest missing values. If minimum players was equal to 0, players was set equal to the value of maximum players, and if both were missing, the value was filled with the mean of minimum players.

Prior to clustering, the data was transformed using PCA. Outliers were identified and removed using the interquartile range (IQR) method because results are sensitive to outliers when performing PCA. For the minimum age, the IQR suggested the threshold be 18 years old, but 21 was chosen as a better threshold to capture subgroups of games targeted for adults (e.g. drinking games or games with a gambling theme). All games have a playing time outside the 97.5 percentiles threshold, 10 or more attributes, 5 or more players were considered an outlier. After the data cleaning process, 46,642 observations (60.8% of total observations of interest) and 15 variables remained.

In figure **1**, the correlation matrix suggests that there are two main sets of variables; a popularity set (wishing, wanting, trading, owned, number of comments, etc.) and a rating set (average rating, weighted average, total of attributes, etc.). To capture clustering results on a scatter plot while preserving all original factors, PCA was done to reduce dimensionality and combine variables correlated with each other. The data was simplified from 15 factors into 2 factors that captured characteristics from the popularity and rating set. The popularity set was reduced into one by keeping only the top principal component that explained 84% of the variability of the data, the rating set was reduced into the first principal component that explained 57% of the variability of the data. The next step was to take reduced data points, standardize them and perform k-means clustering. An elbow plot (see Appendix B) was used to determine the optimal number of clusters, in our case it was 4 clusters, the cluster results can be seen below:

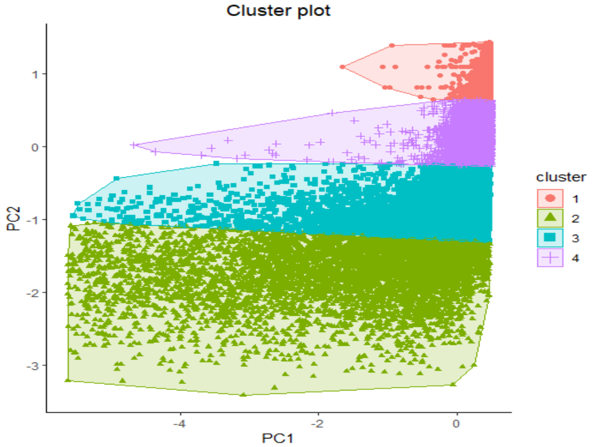


Figure 5: Clusters Groups

3 Experiments & Evaluation

3.1 Regression Models

Both the *average rating* and *wishlist* were compared as dependent variables based on our factors. In this dataset, *wishlist* is measure of how many BGG users saved a game to their collection to indicate their interest, and *average rating* is the average rating given to a board game by BGG users. However, when comparing the two models, the R^2 value when using *average rating* as the dependent factor was much lower at 0.378. Therefore, it was determined that *wishlist* would be the proper dependent variable since it resulted in a higher R^2 value which explains the demand better. Figure below plots our residuals vs. fitted values for our multiple regression model. Although the adjusted R^2 value is quite high at 0.7086, it suggests that our data follows a nonlinear relationship based on the non-normal shape of our Q-Q plot.

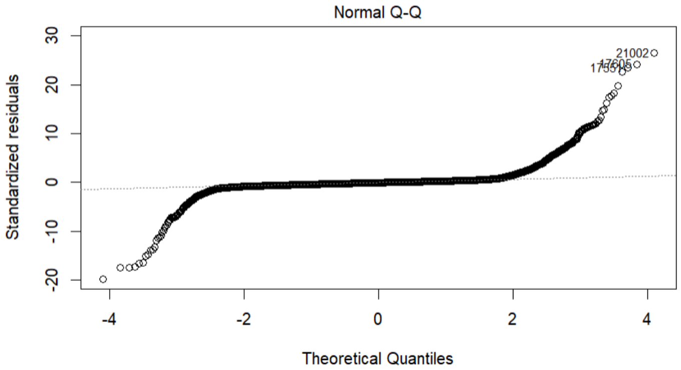


Figure 6: Multiple Linear Regression Residuals vs. Fitted Values Diagnostic

Thus, we decided to explore other types of non-linear regression models and advanced techniques such as random forest and SVM to further improve the model. Other scopes for improvement such as outlier detection were explored, but these data points were not removed from the model because highly desirable games would be considered an outlier. However, its factors were important in contributing to the dependent variable of the regression model. Appendix C shows a more detailed view of how each factor is represented. Factors were removed based on the p-value of the base model, denoted with a **. The comparison of R^2 values of various regression models can be found in below table.

Table 1: R^2 Values between Various Regression Models

Regression Model	Factors Removed	R^2	Adjusted-R^2
Linear-Linear **	×	0.7089	0.7086
Linear-Linear	$x_1, x_{14}, x_{21}, x_{22}$	0.7089	0.7086
Log-Linear	$x_1, x_{14}, x_{21}, x_{22}$	0.5936	0.5931
Linear-Log	$x_1, x_{14}, x_{21}, x_{22}$	0.3284	0.3277
Log-Log	×	0.8217	0.8215
Random Forest	×	0.8099	–
SVM Nu	×	0.2266	–
SVM Eps	×	0.2779	–

When comparing the R^2 of all the models, the *log-log* regression model yielded the highest adjusted R^2 value of 0.8217. For the SVM-nu model, the emphasis is placed on how many support vectors the user wants, while in the epsilon version, it places emphasis on reducing error regardless of how high the resulting SVM parameter ends up being. With both kernels set to radial, and number of support vectors for nu equal to 11370 and for epsilon equal to 3467, neither approach yielded a well fit model. R^2 values for both were significantly lower than the results from other regression models. Based on the R^2 value of 0.8099, the random forest regression model also performed well. However, in terms of interpretability, the random forest model offers far fewer insights for only 0.2 difference in R^2 value and thus would not easily inform major marketing changes. Figure below shows that the *log-log* model accomplishes numerous things: transforms our data to achieve a more linear relationship, makes the distribution more normal, makes the variance more constant, and achieves a higher R^2.

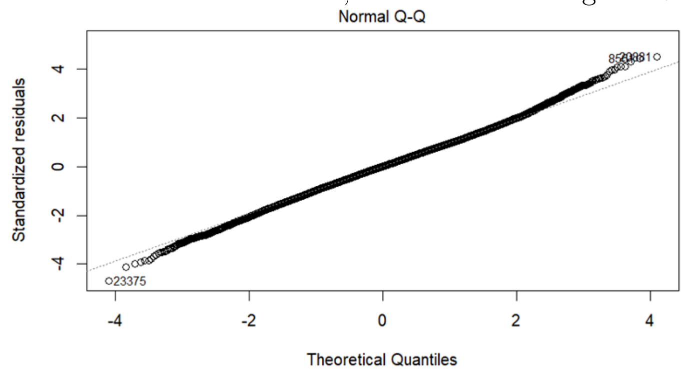


Figure 7: Log-Log Regression Residuals vs. Fitted Values Diagnostic

Because of the non-linear data and the high R^2 value of 0.8217, we decided to explore the *log-log* model more in-depth to assure that the data fit the model well enough to explain a significant portion of variation, and our predictions are likely to be reliable. Various adjustments were made to the log-log model such as removing insignificant factors based on its p-value and multicollinearity factors based on VIF values greater than 5. When looking at the VIF values for the factors in log-log model, there some factors (*numcomments*, *numweights*, *trading*, and *userated*) that had a VIF value that was greater than 5, signifying the presence of multicollinearity in our regression model (see Appendix D). The performance table below shows the resultant R^2 values after removing factors that were not statistically significant and factors with a high VIF value.

Table 2: Comparisons of log-log models after removing for non-significant factors and multicollinearity

Factors Removed	Multi-Collinearity	R^2	Adjusted-R^2
×	×	0.8217	0.8215
x_8, x_{23}	×	0.8217	0.8214
x_8, x_{23}	x_7, x_9	0.8184	0.8182
x_8, x_{23}, x_{25}	x_7, x_9, x_{10}	0.8151	0.815
$x_8, x_{11}, x_{23}, x_{25}, x_{27}$	x_7, x_9, x_{10}	0.8151	0.815
x_8, x_{23}, x_{25}	x_6, x_7, x_9	0.7989	0.7987
$x_8, x_{13}, x_{23}, x_{25}$	x_6, x_7, x_9	0.7989	0.7987

The table above shows that removing *trading*(x_9) and *numweights*(x_9) together results in an R^2 value that is only $3.0e-3$ less. We were then left with two variables, *numcomments*(x_6) and *usersrated*(x_10) that had a VIF value greater than 5. However, only one needed to be kept because both measured the amount of engagement a board game received: the number of comments on a game and the number of users who rated the game on BGG. Both variables help explain the popularity of a board game because the more comments or users who rate the game, the more popular it would be. We decided to keep the variable that resulted in the higher R^2 value, which was *numcomments*. Although the R^2 value decreases from 0.8217 to 0.8151 after these changes, the overall decrease is negligible at only $6.3e-3$. The trade-off of addressing the multicollinearity issues in our model far outweighs the slight decrease in the resultant adjusted-R^2 value. Appendix F shows there are no more factors with a VIF greater than 5 meaning multicollinearity problems in our model were addressed.

After adjusting for non-significant factors and multicollinearity, Appedix F illustrates *log-log* regression model summary. With large F-statistic at 5206, small p-value and high adjusted-R^2 value of 0.815, we can therefore conclude that there is strong evidence suggesting a relationship does exist between our dependent variable and factors. With the coefficients estimated by the *log-log* regression model, the final formula of the log transformed response and predictor variables can be written as below.

$$
\begin{aligned}
\log(\text{wishlist}) = {} & -3.398 + 0.002{*}\log(\text{maxplayers}) + 0.023{*}\log(\text{minage}) \\
& - 0.206{*}\log(\text{minplayers}) + 0.022{*}\log(\text{playingtime}) + 2.343{*}\log(\text{average\_rating}) \\
& + 0.448{*}\log(\text{averageweight}) + 0.849{*}\log(\text{numcomments}) - 0.133{*}\log(\text{War}) \\
& - 0.151{*}\log(\text{Strategy\_Deduction\_Puzzle}) + 0.178{*}\log(\text{History\_Politics}) \\
& + 0.392{*}\log(\text{Fantasy\_SciFi\_Horror}) + 0.306{*}\log(\text{Adventure}) + 0.208{*}\log(\text{MediaBased}) \\
& + 0.132{*}\log(\text{Nature}) + 0.194{*}\log(\text{IndustryBased}) - 0.153{*}\log(\text{Educational\_Kids}) \\
& - 0.142{*}\log(\text{Other}) + 0.152{*}\log(\text{Medium\_producing\_company}) \\
& - 0.240{*}\log(\text{Self\_published}) - 0.471{*}\log(\text{Age\_of\_game})
\end{aligned}
$$

Figure 8: Log Transformed Response and Predictor Variable Formula

A negative intercept indicates that if all the factors were not present, there would be no demand as the *wishlist* prediction would be negative. According to the coefficients, among the 20 factors, the three most effective are: *numcomments*, *averagerating*, and *averageweight*. This means that for a particular board game, for a 1% increase in the number of comments, the *wishlist* count increases by 0.849% while all other variables in the model are held constant. Similarly, a 1% increase in average rating and 1% increase in game complexity increase the interest for a game by 2.343% and 0.448% respectively. Alternatively, the age of the game, self-published, and the minimum number of players were three least effective because of the negative coefficient values. A decrease in 0.471% of *wishlist* when the age of a game increases by 1% indicates that newer games generally have less demand which makes sense since newer games need to have more users try it first and generate buzz before it garners more interest. Similarly, a self-published game is less likely to generate more interest than a game published by a larger company as well as the interest in a particular game decreases as more players are required for a game.

As a case study, we ran our prediction on a game that has yet to come out to see what kind of attention we might expect it to receive once it is available on the market. Previously a video game, now a tv show, and soon to be a board game, ”The Last of Us” tabletop game is slated to be released in December of 2023. It was Kickstarter game and got much of its funding from the public which makes it an interesting example for prediction. Variables were estimated based on the makers description, and unknowns were imputated with average values from the original dataset. We experimented with both low and high values for *numcomments* (one of the more important predictors) to see how changing buzz and community activity around a game might influence the interest in a game. The high value was set to the number of comments currently on the Kickstarter, and the low number set to half that value (1545 and 772 respectively). With a great deal of commentary around the game, it is expected that 23 people will have the game on their list after 1 year of being released. With half the comments, this number drops to 17. This result would support a marketing strategy of centering community engagement, such as social media involvement, feedback from future consumers, or back-and-forth with developers like online AMAs with creators.

3.2 Market Segmentation

3.2.1 Text Clustering Approach

To tune our k-means and mini-batch k-means number of clusters, model inertia versus k plots were constructed for both algorithms.

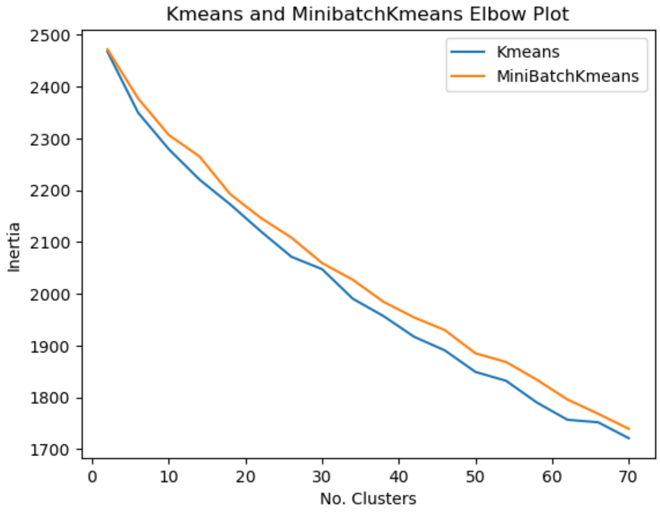


Figure 9: K-means and Mini-Batch K-means Inertia vs Number of Clusters

It can be easily seen that our inertia is not plateau yet even at k = 70. This is because these algorithms did not reach an optimal solution for cluster with the provided dataset’s text content, but the measurement of inertia starts to trend down slower around 30-40 clusters. That suggests k = 30 can be chosen for our data set cluster model.

Using k = 30, different metrics such as homogeneity, completeness, adjusted rand-index and silhouette are plotted for kmeans with and without LSA data and the same for Mini-Batch Kmeans with and without LSA data. Cluster labels are compared to the primary category of these board game which is provided in the original data set. These categories serve as ground truth.

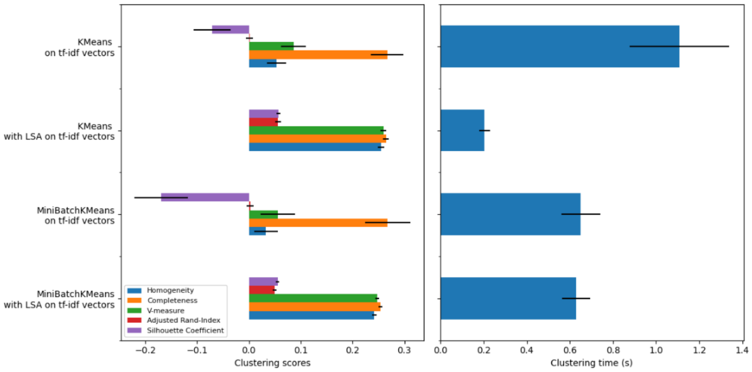


Figure 10: Comparison of Kmeans and Mini-Batch Kmeans model score and fit time

It is observable that without LSA data, both k-means and mini-batch k-means clusters have negative silhouette metric which suggests that our clusters are collapsing to each other. Moreover, the homogeneity, completeness, and adjusted rand-index are computed by comparing cluster labels with primary categories provided in our dataset. Note that both k-means and mini-batch with LSA k-means yield very similar results (27% match between cluster labels and predetermined categories). K-means fit time is much faster than mini-batch k-means, but the error bar is much higher. This is an indication of variability in fitting. Overall, mini-batch k-means with LSA is more stable and provides relatively similar results.

Figures 11 and 12 below show ranking of cluster labels from mini-batch k-means model by average rating and number of users rated (popularity) respectively. This allows a comparison between these cluster and give us insights on how competitive each cluster in the board game market is.

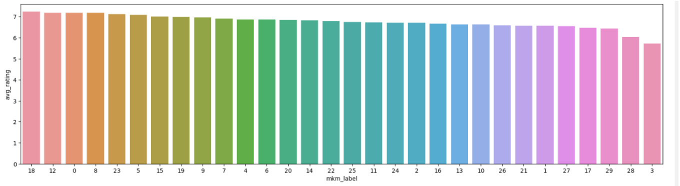
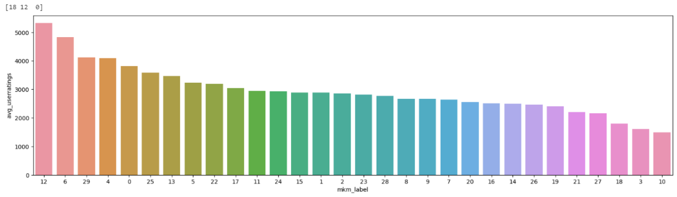


Figure 11: Avg. Rating versus Cluster Group number



```
[18 12  0]
```

Figure 12: Number of User Rated versus Cluster Group Number

Figures 13 and 14 show the best two clusters with the highest average ratings. Cluster no. 18 with highest average rating has words like worker, field, work, factory, pay, town, builder, cacao, season, coin. This implies that the highly evaluated game genre is the worker placement game with agriculture theme. Cluster 12 with words such as civilization, age, technology, history, military, nation. This is a clear sign for civilization and historical theme with technology tree, so games like: Through the Ages: A New Story of Civilization is referred.

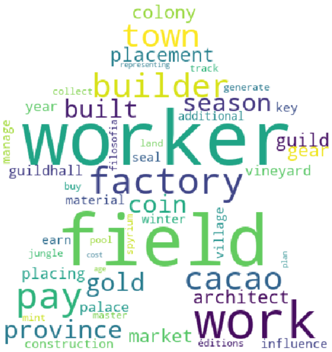


Figure 13: Cluster No.18 word cloud

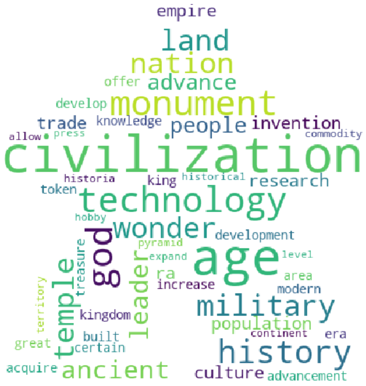


Figure 14: Cluster No.12 word cloud

Figure 15 and 16 below show the second and third best clusters with number of users rated. Cluster no.6 with highest number of users rated has words like alien, mission, planet, star, ship, galaxy, trek, colony, marine, and race. This is a clear indication for a space theme with inter-dimensional travel. On the other hand, figure 16 with cluster 29 with words like: train, rail, railway, railroad, route, ticket, station, passenger, line, a clear indication of transportation games like Ticket to Ride or Colt Express. Overall, each cluster does belong to its own game genre, and by exploring these clusters, the game producer can easily make the right decision to pursue their next board game project.

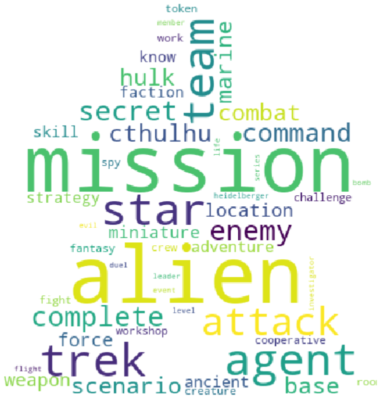


Figure 15: Cluster No. 6 word cloud

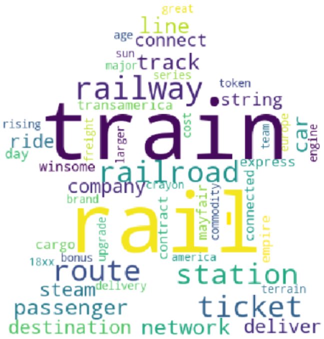


Figure 16: Cluster No. 29 word cloud

Improvements on this clustering method can be made by collecting more demographic data from the user who rates each game. Information like location of users, locations of publisher, gender or age can be used to rank our cluster better within a niche user group. For example, having a teenager category could be used to rank these clusters to target that group.

3.2.2 Numeric Clustering Approach

After extracting only the principal components,the box-and-whisker plots below was created from the clustered groups which highlighted some key findings. Cluster 4 had the highest median average rating of 6.47 which suggests that board game companies should concentrate on games similar to those in cluster 4. Other box-and-whisker plots show that for cluster 4, the median minimum age is 12 years old, quite different from the other clusters with a median minimum age range between 8 and 10 years old. Our cluster of interest also had a higher median number of attributes, 6, as opposed to the other clusters with the median ranging between 4 and 5. The median playing time of 60 minutes for cluster 4 was double the median of the other clusters at 30 minutes. These clustering results showed that games created for the following demographic usually result in a higher than average rating: for preteens and older, a higher-than-average complexity, and require at least 60 minutes of overall playtime.

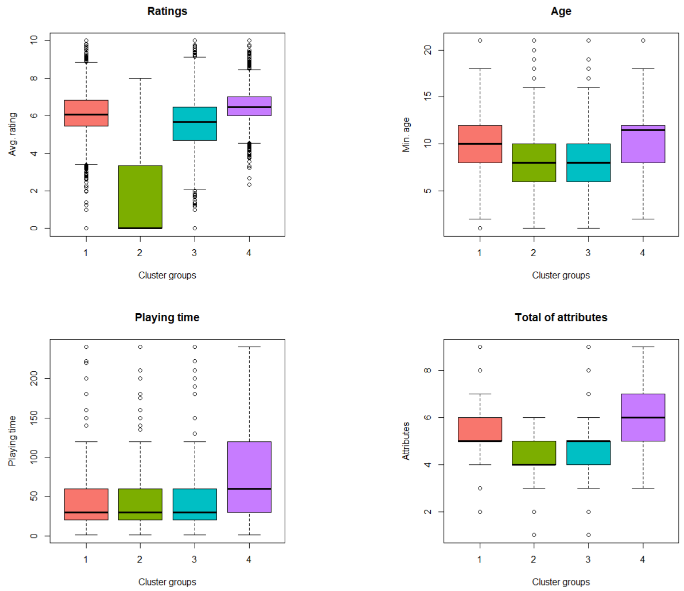


Figure 17: Box-and-whisker plots of rating, minimum age, playing time, total attributes vs cluster groups

3.3 Recommender System

3.3.1 Content-based Filtering

Figures 18 and 19 below compare two different metrics for content-based recommender. The cosine similarity of the top 5 games with highest rating were chosen by theoretical users were computed along with their weighted average. These games are Clue, Monopoly, Dixit: Journey, Love Letter and Stone Age (weighted lowest to highest).The Manhattan distance was also computed for the same 5 games. On the x-axis, there are 5 games that have the highest Cosine similarities or lowest Manhattan distance with the chosen 5 games. We can see that cosine similarities yield slightly different results from Manhattan distance. Comparing to cosine similarity, Manhattan distance between these games is much closer to each other which can be hard to distinguish between games. To better rank games based on their content-based matrices, cosine similarity should be chosen to maximize the difference between games.

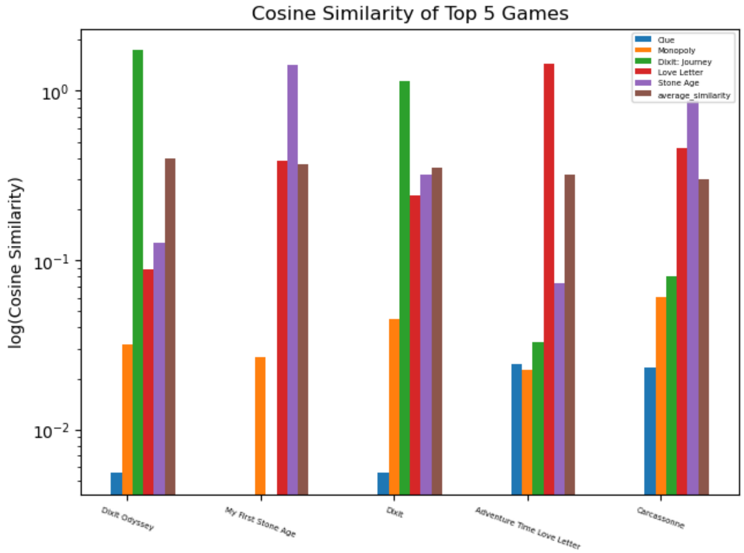


Figure 18: Cosine Similarity for top 5 games

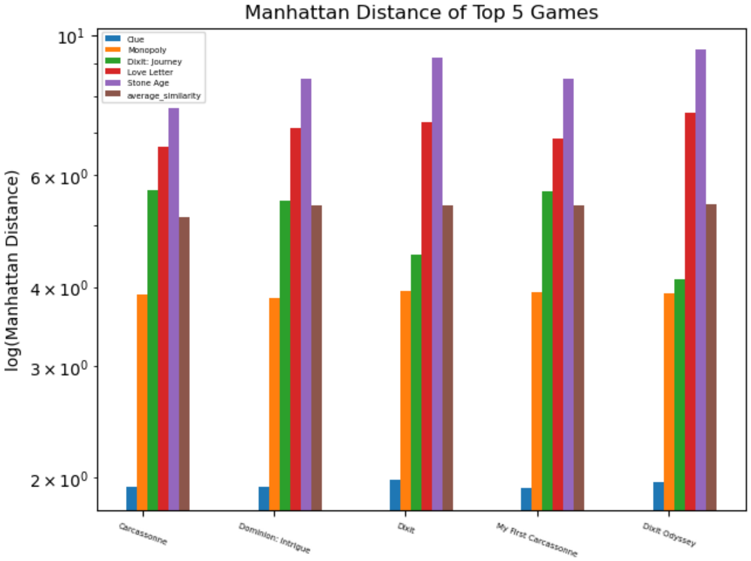
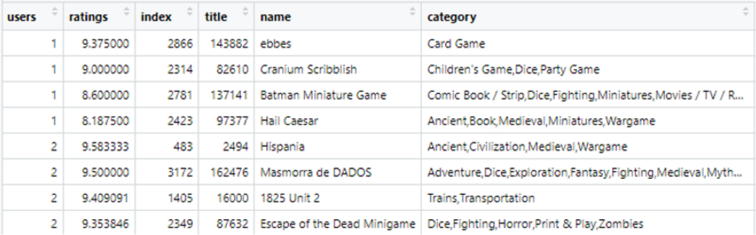


Figure 19: Manhattan Distance for top 5 games

3.3.2 Collaborative Filtering

We use the *known* part of the test users' item data (4 items for each user) to make predicted ratings to predict top 4 items. Below are output for first two users for UBCF model:



users	ratings	index	title	name	category
1	9.375000	2866	143882	ebbes	Card Game
1	9.000000	2314	82610	Cranium Scribblish	Children's Game,Dice,Party Game
1	8.600000	2781	137141	Batman Miniature Game	Comic Book / Strip,Dice,Fighting,Miniatures,Movies / TV / R...
1	8.187500	2423	97377	Hail Caesar	Ancient,Book,Medieval,Miniatures,Wargame
2	9.583333	483	2494	Hispania	Ancient,Civilization,Medieval,Wargame
2	9.500000	3172	162476	Masmorra de DADOS	Adventure,Dice,Exploration,Fantasy,Fighting,Medieval,Myth...
2	9.409091	1405	16000	1825 Unit 2	Trains,Transportation
2	9.353846	2349	87632	Escape of the Dead Minigame	Dice,Fighting,Horror,Print & Play,Zombies

Figure 20: Top 4 games predicted ratings for first 2 users

The RMSE and run time were used to evaluate the effectiveness of the recommender systems and the improved algorithm's suggestion accuracy. The RMSE value was calculated by comparing the actual rating to the expected rating among users. With declining RMSE value, the suggested approach becomes more accurate. Using the calcPredictionAccuracy method, we can now test the prediction error of UBCF, IBCF and SVD models on the *unknown* test user ratings. The results below focus on RMSE with the errors calculated per test user on their *unknown* data.

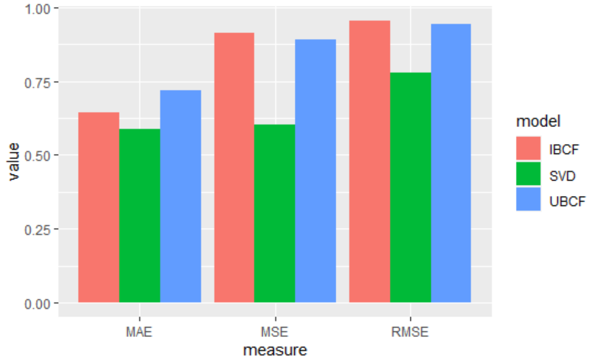
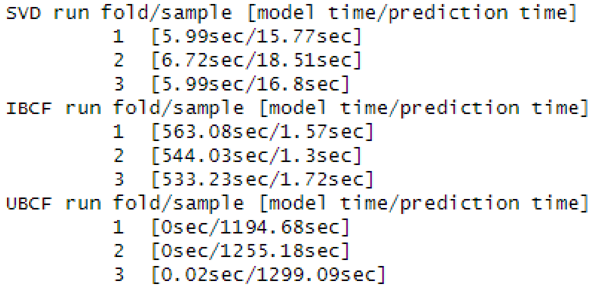


Figure 21: Comparison of RMSE between models

UBCF model performs the worst with the biggest RMSE value while SVD model outperforms all other models. Next, let's check out model and prediction run time between 3 models to see which model runs most efficiently. The below figure obviously shows that SVD method has the best average of model and prediction run time, while IBCF and UBCF models took considerable run time in training model and prediction respectively.



```
SVD run fold/sample [model time/prediction time]
         1  [5.99sec/15.77sec]
         2  [6.72sec/18.51sec]
         3  [5.99sec/16.8sec]
IBCF run fold/sample [model time/prediction time]
         1  [563.08sec/1.57sec]
         2  [544.03sec/1.3sec]
         3  [533.23sec/1.72sec]
UBCF run fold/sample [model time/prediction time]
         1  [0sec/1194.68sec]
         2  [0sec/1255.18sec]
         3  [0.02sec/1299.09sec]
```

Figure 22: Comparison of run time between models

To conclude, Singular value decomposition model performs better than than the collaborative filtering family (UBCF and IBCF) in this board game setting. It's not unexpected, given that well-known large internet businesses like Netflix and YouTube all used SVD as their recommendation system until recently. Despite the simplicity of implementation and understanding, it is also worth noting that both memory-based (UBCF and IBCF) and model-based (SVD) have limitations. Firstly, these algorithms do not too work well on very sparse ratings matrices. Secondly, they are computationally expensive because the full user database must be analyzed as the foundation for forming recommendations, limiting their ability to handle customers who have made no purchases or things with no single purchase (the cold-start problem). In these scenarios, content-based filtering approach is preferred.

4 Conclusion & Discussion

Based on the non-linear data and an adjusted R^2 value of 0.815, we are able to conclude that a log-log regression model does the best job in explaining our data. Most important factors in increasing the demand of a game are high number of comments on BGG, average rating, and complexity rating. These factors support our initial hypothesis that supporting a marketing strategy centering around community engagement, e.g. social media involvement, feedback from future consumers, or back-and-forth with developers like online AMAs with creators, would be the most effective in marketing a new board game. However, some limitations were encountered in the sentiment analysis approach, and p-values of the converted sentiment score were not good contributors to the model, indicating that sentiment analysis on descriptions did not contribute to determining which games individuals are most attracted to.

Unsupervised learning (clustering) and recommender system methods were utilized to perform market research on our board game data. This not only enables board game producers to gain deep insights on certain board game theme to help them make the most out of their investments, but users can also find their favorite board games. To overcome limitations of collaborative and content-based filtering approaches such as cold start and the data paucity troubles, a hybrid recommendation system (combination of these two methods) is considered to be a perfect solution and should be further experimented.

References

Jiawei Han, Jian Pei, Yiwen Yin, and Runying Mao. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. *Data mining and knowledge discovery*, 8:53–87, 2004.

Mark Lutter and Linus Weidner. Newcomers, betweenness centrality, and creative success: A study of teams in the board game industry from 1951 to 2017. *Poetics*, 87:101535, 2021.

G Wayne Miller. *Toy wars: The epic struggle between GI Joe, Barbie, and the companies that make them*. Crown Business, 2012.

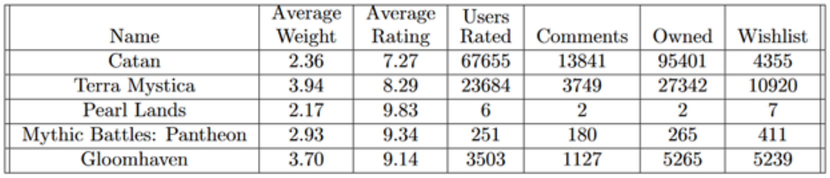
Ivan Markovsky and Ivan Markovsky. Algorithms. *Low Rank Approximation: Algorithms, Implementation, Applications*, pages 73–106, 2012.

Josiah Mork. A strategic social media marketing plan to launch a new tabletop role-playing game. 2022.

S Seetharaman. How big is the board game market. *Pipe Candy Blog. Recuperado de https://blog. pipecandy. com/board-games-market*, 2020.

Appendices

A Appendix A



Name	Average Weight	Average Rating	Users Rated	Comments	Owned	Wishlist
Catan	2.36	7.27	67655	13841	95401	4355
Terra Mystica	3.94	8.29	23684	3749	27342	10920
Pearl Lands	2.17	9.83	6	2	2	7
Mythic Battles: Pantheon	2.93	9.34	251	180	265	411
Gloomhaven	3.70	9.14	3503	1127	5265	5239

Figure A.1: Board Game EDA

B Appendix B

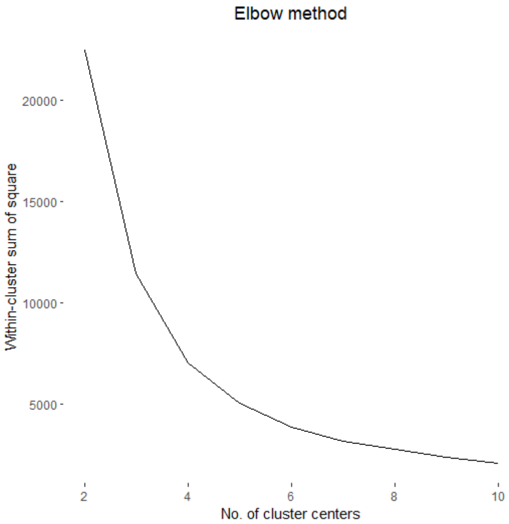


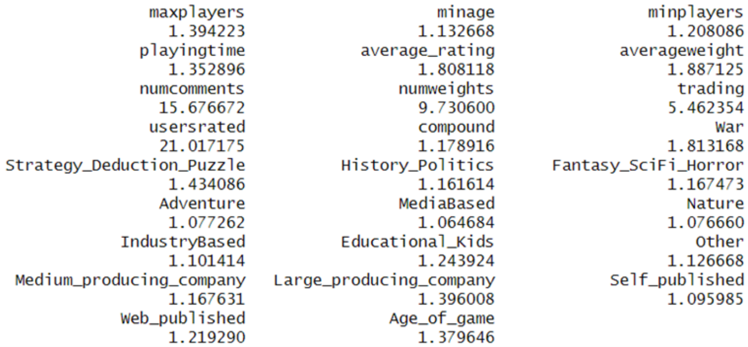
Figure B.1: Elbow method for optimal number of clusters

C Appendix C

Table 3: Variable Representation

Variable	Factor Represented	Description
x_1	maxplayers	Maximum number of players.
x_2	minage	Minimum age requirement.
x_3	minplayers	Minimum number of players.
x_4	playingtime	Average length of a game.
x_5	averageweight	Rating for how difficult a game is to understand.
x_6	numcomments	Number of users who commented on the game.
x_7	numweights	Number of users who contributed to the weight.
x_8	owned	Number of users who own the game.
x_9	trading	Users who have the game listed for trade.
x_{10}	usersrated	Number of users who rated the game.
x_{11}	compound	Compound sentiment score of game description.
x_{12}	wishlist	Number of users who have the game on their wishlist.
x_{13}	War	Primary/Secondary Board game category.
x_{14}	Strategy_Deduction_Puzzle	Primary/Secondary Board game category.
x_{15}	History_Politics	Primary/Secondary Board game category.
x_{16}	Fantasy_SciFi_Horror	Primary/Secondary Board game category.
x_{17}	Adventure	Primary/Secondary Board game category.
x_{18}	MediaBased	Primary/Secondary Board game category.
x_{19}	Nature	Primary/Secondary Board game category.
x_{20}	IndustryBased	Primary/Secondary Board game category.
x_{21}	Educational_Kids	Primary/Secondary Board game category.
x_{22}	Other	Primary/Secondary Board game category.
x_{23}	Small_producing_company	Size of company that published the game.
x_{24}	Medium_producing_company	Size of company that published the game.
x_{25}	Large_producing_company	Size of company that published the game.
x_{26}	Self_published	Game was self-published.
x_{27}	Web_published	Game was web-published.
x_{28}	Age_of_game	Years since 2023 the game has been released.

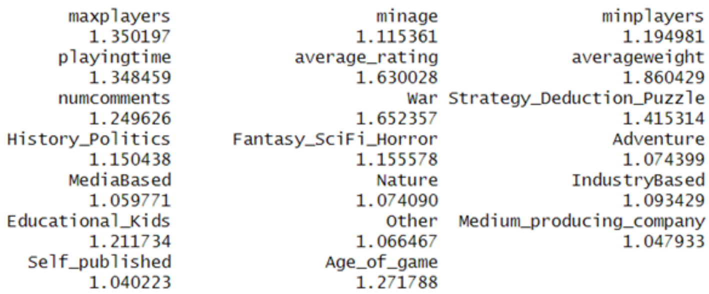
D Appendix D



```
               maxplayers                    minage               minplayers
                 1.394223                  1.132668                 1.208086
              playingtime            average_rating            averageweight
                 1.352896                  1.808118                 1.887125
              numcomments                numweights                  trading
                15.676672                  9.730600                 5.462354
               usersrated                  compound                      War
                21.017175                  1.178916                 1.813168
Strategy_Deduction_Puzzle          History_Politics     Fantasy_SciFi_Horror
                 1.434086                  1.161614                 1.167473
                Adventure                MediaBased                   Nature
                 1.077262                  1.064684                 1.076660
            IndustryBased          Educational_Kids                    Other
                 1.101414                  1.243924                 1.126668
 Medium_producing_company   Large_producing_company           Self_published
                 1.167631                  1.396008                 1.095985
            Web_published               Age_of_game
                 1.219290                  1.379646
```

Figure D.1: VIF Values of Log-Log Model (multicollinearity not addressed)

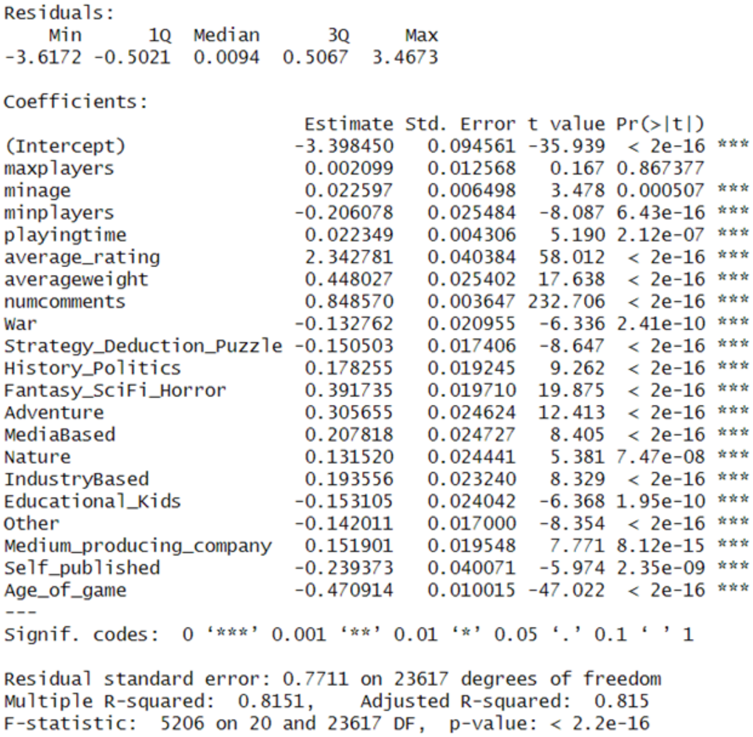
E Appendix E



```
             maxplayers                        minage                   minplayers
               1.350197                      1.115361                     1.194981
            playingtime                average_rating                averageweight
               1.348459                      1.630028                     1.860429
            numcomments                           War Strategy_Deduction_Puzzle
               1.249626                      1.652357                     1.415314
        History_Politics         Fantasy_SciFi_Horror                    Adventure
               1.150438                      1.155578                     1.074399
             MediaBased                        Nature                 IndustryBased
               1.059771                      1.074090                     1.093429
         Educational_Kids                         Other      Medium_producing_company
               1.211734                      1.066467                     1.047933
           Self_published                   Age_of_game
               1.040223                      1.271788
```

Figure E.1: VIF Values of Final Log-Log Model (after multicollinearity addressed)

F Appendix F



```
Residuals:
    Min      1Q  Median      3Q     Max 
-3.6172 -0.5021  0.0094  0.5067  3.4673 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              -3.398450   0.094561 -35.939  < 2e-16 ***
maxplayers                0.002099   0.012568   0.167 0.867377    
minage                    0.022597   0.006498   3.478 0.000507 ***
minplayers               -0.206078   0.025484  -8.087 6.43e-16 ***
playingtime               0.022349   0.004306   5.190 2.12e-07 ***
average_rating            2.342781   0.040384  58.012  < 2e-16 ***
averageweight             0.448027   0.025402  17.638  < 2e-16 ***
numcomments               0.848570   0.003647 232.706  < 2e-16 ***
War                      -0.132762   0.020955  -6.336 2.41e-10 ***
Strategy_Deduction_Puzzle -0.150503   0.017406  -8.647  < 2e-16 ***
History_Politics          0.178255   0.019245   9.262  < 2e-16 ***
Fantasy_SciFi_Horror      0.391735   0.019710  19.875  < 2e-16 ***
Adventure                 0.305655   0.024624  12.413  < 2e-16 ***
MediaBased                0.207818   0.024727   8.405  < 2e-16 ***
Nature                    0.131520   0.024441   5.381 7.47e-08 ***
IndustryBased             0.193556   0.023240   8.329  < 2e-16 ***
Educational_Kids         -0.153105   0.024042  -6.368 1.95e-10 ***
Other                    -0.142011   0.017000  -8.354  < 2e-16 ***
Medium_producing_company  0.151901   0.019548   7.771 8.12e-15 ***
Self_published           -0.239373   0.040071  -5.974 2.35e-09 ***
Age_of_game              -0.470914   0.010015 -47.022  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7711 on 23617 degrees of freedom
Multiple R-squared:  0.8151,	Adjusted R-squared:  0.815 
F-statistic:  5206 on 20 and 23617 DF,  p-value: < 2.2e-16
```

Figure F.1: Log-Log regression model summary after adjusting for non-significant factors and multicollinearity)