

Wheat from Chaff, Catan-Style:

*Predicting the General Quality of Board Games
using Binary Classification Algorithms*

Aaron Fehir

Abstract

This paper documents the utilization of various binary classification algorithms (SVM, Random Forest, and Logistic Regression) on the largest boardgame database in the world (available on Kaggle and created from boardgamegeek.com using API/JSON-scraping). The goal was to train three machine learning algorithms (SVM, Random Forest, and Logistic Regression) to reliably determine whether a boardgame would receive above or below average ratings on boardgamegeek.com (a good proxy for commercial success, depending on the scale of operations), without playing the game, and ideally without reading the rule, in order to cut down sorting time for reviewers.

Introduction / Business Understanding

In this scenario, a fictitious boardgame company, Cardboard Kings (CK), is swamped with submissions for ideas to make new games. While this might seem beneficial, the sheer number is such that they cannot look at each submission and are worried that potentially lucrative designs might slip through their fingers. With the increased popularity of boardgames in recent years, this is a reasonable fear; struggles for market share have become increasingly tense, and game designers -- especially new or first-time designers --- often pitch their designs to multiple companies (Browne, 2017, Martin 2017)). As a result, the ability to sort "the wheat from the chaff" as fast as possible is extremely valuable for board game companies. In fact, depending on the size of the company, publishing one or two hits in a year is the difference between thriving and bankruptcy (Woods, 2012). Phrased as a Business Problem: which games are worth being reviewed in-depth, and which should be immediately discarded?

Data Understanding and Preparation

While it would certainly be possible to phrase this as a regression problem designed to predict the average rating a game will receive on boardgamegeek.com, a more reliable translation of this business problem into an analytics problem is via binary classification. Specifically, can a model reliably (at least 80% of time, as measured by specificity and sensitivity) correctly predict whether a design submission will be above or below the average game on boardgame geek ($x < 6.9$ | $x > 6.9$), using categorical and numeric design-decisions as features? If successful, sorting new designs with this model could potentially halve the amount of work needed to review the same

design-input, or make CK's current review process twice as fast, depending on one's outlook. A priori assumptions are that as time increases, the frequency of boardgame production increases, boardgames become more complex, and that ratings improve.

Original Data and Pre-emptive Dimensional Reduction

As stated in the abstract, the database used for this project was scraped in 2016 from boardgamegeek.com (BBG), which is the largest boardgame enthusiast site/ database in the world. This was done by Baldassare (2017) for the purposes of market segmentation, it was later uploaded to Kaggle, which is where it was downloaded for this project. After importing the data from its initial SQLite format, the data was coerced into a data.frame composed of 90,400 observations, and 81 features. For a complete list of the initial features, which were fairly evenly split between <chr> and <dbl>, see fig 1. Many of these features, including the majority of the polling features, and all stats located [45:55] were discarded immediately, as they were missing a substantial portion of their values (see fig 2), and not relevant to the business question; CK would not have access to the community's response to a game's design prior to publication. Additionally, all features beginning with stats.family were removed, as was anything that made the database semi-structured, like the attributes.t.links., details.thumbnail, and details.image features. The game.description feature was also removed for this reason, since it both required natural language processing to become workable, and didn't directly relate to design features. The game.type feature and attributes.boardgameexpansion were also removed, but not before they were used to help refine the dataset, as they allowed arcade/computer games and boardgame expansions to be removed, which prevented over-representation of mechanics and categories shared by an original game, and its expansions.

[1] "row_names"	"game.id"
[3] "game.type"	"details.description"
[5] "details.image"	"details.maxplayers"
[7] "details.maxplaytime"	"details.minage"
[9] "details.minplayers"	"details.minplaytime"
[11] "details.name"	"details.playingtime"
[13] "details.thumbnail"	"details.yearpublished"
[15] "attributes.boardgameartist"	"attributes.boardgamecategory"
[17] "attributes.boardgamecompilation"	"attributes.boardgamedesigner"
[19] "attributes.boardgameexpansion"	"attributes.boardgamefamily"
[21] "attributes.boardgameimplementation"	"attributes.boardgameintegration"
[23] "attributes.boardgamemechanic"	"attributes.boardgamepublisher"
[25] "attributes.total"	"stats.average"
[27] "stats.averageweight"	"stats.bayesaverage"
[29] "stats.family.abstracts.bayesaverage"	"stats.family.abstracts.pos"
[31] "stats.family.cgs.bayesaverage"	"stats.family.cgs.pos"
[33] "stats.family.childrensgames.bayesaverage"	"stats.family.childrensgames.pos"
[35] "stats.family.familygames.bayesaverage"	"stats.family.familygames.pos"
[37] "stats.family.partygames.bayesaverage"	"stats.family.partygames.pos"
[39] "stats.family.strategygames.bayesaverage"	"stats.family.strategygames.pos"
[41] "stats.family.thematic.bayesaverage"	"stats.family.thematic.pos"
[43] "stats.family.wargames.bayesaverage"	"stats.family.wargames.pos"
[45] "stats.median"	"stats.numcomments"
[47] "stats.numweights"	"stats.owned"
[49] "stats.stddev"	"stats.subtype.boardgame.bayesaverage"
[51] "stats.subtype.boardgame.pos"	"stats.trading"
[53] "stats.usersrated"	"stats.wanting"
[55] "stats.wishing"	"polls.language_dependence"
[57] "polls.suggested_numplayers.1"	"polls.suggested_numplayers.10"
[59] "polls.suggested_numplayers.2"	"polls.suggested_numplayers.3"
[61] "polls.suggested_numplayers.4"	"polls.suggested_numplayers.5"
[63] "polls.suggested_numplayers.6"	"polls.suggested_numplayers.7"
[65] "polls.suggested_numplayers.8"	"polls.suggested_numplayers.9"
[67] "polls.suggested_numplayers.Over"	"polls.suggested_playerage"
[69] "attributes.t.links.concat.2..."	"stats.family.amiga.bayesaverage"
[71] "stats.family.amiga.pos"	"stats.family.arcade.bayesaverage"
[73] "stats.family.arcade.pos"	"stats.family.atarist.bayesaverage"
[75] "stats.family.atarist.pos"	"stats.family.commodore64.bayesaverage"
[77] "stats.family.commodore64.pos"	"stats.subtype.rpgitem.bayesaverage"
[79] "stats.subtype.rpgitem.pos"	"stats.subtype.videogame.bayesaverage"
[81] "stats.subtype.videogame.pos"	

Figure 1: Initial Features

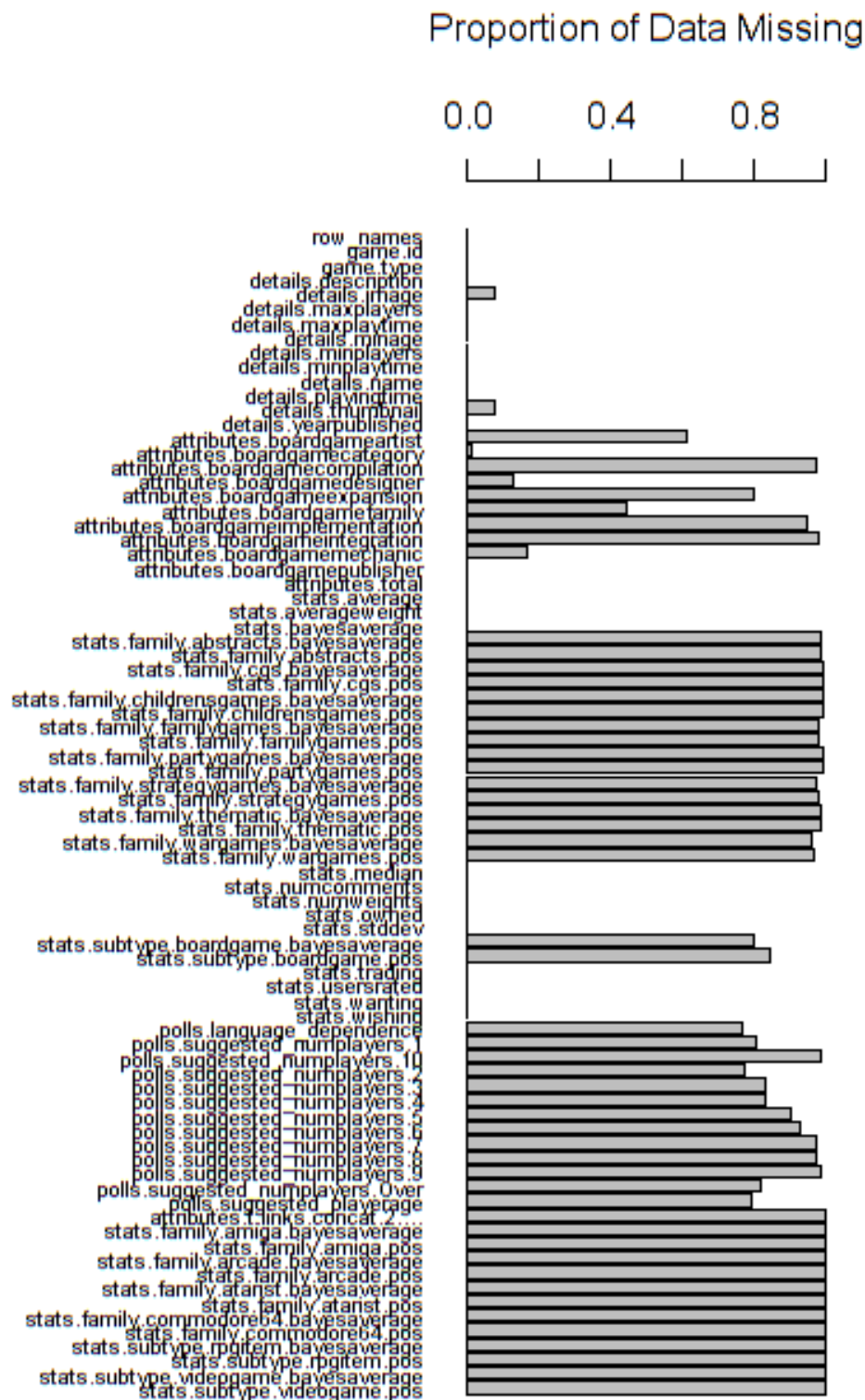


Figure 2: Missing Data by Feature

Intermediate Data and Pre-Processing

After the initial dimensional reduction, and some of the data verification discussed in the *Modelling and Evaluation* section, the data file was 4083 observations by 19 features, see Table 1. A few of these features were kept in with the intension of discarding them immediately after using them to help detect outliers (such as users.rated), while others (particularly polls.language dependence) were intended to be used as features to assess design decisions. Though not shown in Table 1, for clarity reasons, the dummy variables for mechanics and category were also created during this stage using `cSplit_e()` from the `splitstackshape` library (see figs 8 and 9) and for detailed descriptions of their meanings, consult BBG (BoardGameGeek, n.d.)

Index	Feature Name	Feature Description
1	game.id	Primary index, as assigned by BBG
2	details.name	The name of the boardgame
3	details.maxplayers	The maximum amount of people able to play a board game simultaneously
4	details.minage	The minimum recommended age for a boardgame
5	details.yearpublished	The original year in which a game was released / published
6	stats.average	The average rating assigned to a game by BBG users, later combined with a threshold value to produce our target classification
7	details.minplayers	The minimum number of players required to play a boardgame
8	details.minplaytime	The minimum amount of time (in minutes) required to complete a single uninterrupted playthrough
9	details.maxplaytime	The maximum amount of time (in minutes) required to complete a single uninterrupted playthrough
10	polls.language_dependance	The level of language dependence required to play a game at an acceptable level, as determined by majority vote by BBG users.
11	stats.subtype.boardgame.bayesaverage	The Bayesian average of the rating for each boardgame
12	stats.subtype.boardgame.pos	The ordinal ranking of each boardgame from best to worst, with 1 as the optimal value, as determined by BBG's algorithm
13	attributes.boardgameartist	The artist or artists responsible for producing the game's visual assets.
14	attributes.boardgamecategory	The BGG categories are a means with which to group games based on like subjects or similar characteristics.
15	attributes.boardgamedesigner	The designer or designers of the board game's rules.
16	attributes.boardgamemechanic	The ules and/or methods designed to allow the players to interact with the game state. Gameplay can be thought of as the function of its mechanics
17	stats.average	Accidental duplication, removed before model training
18	stats.averageweight	Quantified score representing the complexity of a game
19	stats.usersrated	The count of how many BBG users have rated a game

Final Data and Feature Engineering

After adding the dummy variables (see fig 8 and 9), the number of features climbed to 150. This would continue to increase, as, during EDA, five additional features were engineered. These were: 1) the total number of mechanics used in a given game [Mechanic Sum, AKA `mech_sum`], 2) `mech_sum`'s categorical counterpart: the total number of categories a given game fell into [Category Sum, AKA `cat_sum`], 3) The amount of variation in player-count which a board game could accommodate [`maxplayer` - `minplayer` AKA Player-Spread AKA `player_spread`], 4) the amount of variation in the time required to play a game [`(maxplaytime - minplaytime)` AKA Time-Spread AKA `time_spread`], and 5) a categorical variable which stores information of whether a game was rated above or below average [`successful`]. The former two were produced to try and capture how innovative the game design was,, while the latter two attempted to assess a game's versatility.

In order to combat the results of including a high number of features -- high computational time, and features which added noise without adding accuracy or precision, -- the number of categorical features was significantly reduced by an iterative importance-check / feature-removal carried out during implementation of the Random Forest model. Altogether, the finished dataframe was 2869 observations by 54 features, with the final features summarized in fig 3, and all features classified as either <int> or <dbl>., which are largely interchangeable.

[1] "details.maxplayers"	"details.minage"
[3] "details.yearpublished"	"stats.average"
[5] "details.minplayers"	"details.minplaytime"
[7] "details.maxplaytime"	"stats.averageweight"
[9] "time_spread"	"mechanic_Action...Movement.Programming"
[11] "mechanic_Area.Enclosure"	"mechanic_Area.Movement"
[13] "mechanic_Auction.Bidding"	"mechanic_Crayon.Rail.System"
[15] "mechanic_Deck...Pool.Building"	"mechanic_Paper.and.Pencil"
[17] "mechanic_Pick.up.and.Deliver"	"mechanic_Rock.Paper.Scissors"
[19] "mechanic_Role.Playing"	"mechanic_Roll...Spin.and.Move"
[21] "mechanic_Route.Network.Building"	"mechanic_Secret.Unit.Deployment"
[23] "mechanic_Time.Track"	"mechanic_Trick.taking"
[25] "mechanic_Variable.Phase.Order"	"category_Abstract.Strategy"
[27] "category_Arabian"	"category_Aviation...Flight"
[29] "category_Bluffing"	"category_Civil.War"
[31] "category_Collectible.Components"	"category_Comic.Book...Strip"
[33] "category_Deduction"	"category_Educational"
[35] "category_Exploration"	"category_Farming"
[37] "category_Game.System"	"category_Horror"
[39] "category_Medical"	"category_Music"
[41] "category_Mythology"	"category_Napoleonic"
[43] "category_Puzzle"	"category_Space.Exploration"
[45] "category_Trains"	"category_Transportation"
[47] "category_Travel"	"category_Trivia"
[49] "category_Vietnam.War"	"category_World.War.I"
[51] "category_World.War.II"	"category_Zombies"
[53] "cat_sum"	"mech_sum"

Figure 3: Final Features

Modeling and Evaluation

Verifying Data Quality

After deciding on which features would initially be included in the data file, the data quality of the dataframe was assessed using several methods. Firstly, `distinct_n()` was checked against `n()` using the summary function of `dplyr` for both `game.id` and `game.name` to check for duplicates, since they can both theoretically function as a primary index.. As the values matched, duplication was determined to be absent.

Next, non-NA-missing-values and outliers were scanned for by passing the dataframe through the summary function, and manually looking for unusual qualities. This was complimented by passing questionable features into the plot function. Zero weight games were obviously outliers (see Fig 4), likely from missing values, so observations with `[average.weight = 0]` were filtered out of the dataset. Games published before 1900 were also removed, as they contained a few classics like Chess and Backgammon, but with values as low as -3000, their inclusion was heavily throwing off visual EDA, and they weren't useful for answering the business question. Games with minimum and/or maximum payer counts of 0 were also removed, as there can't be zero people playing the boardgame, making 0 an invalid input. Board games that required less than 11 or more than 600 minutes to play were deemed outliers, and discarded. Finally, any

game with fewer than 20 user ratings (see fig 5) were removed, as low user ratings on BBG are indicative of unpublished games or rule variations on existing games, in the author’s experience.

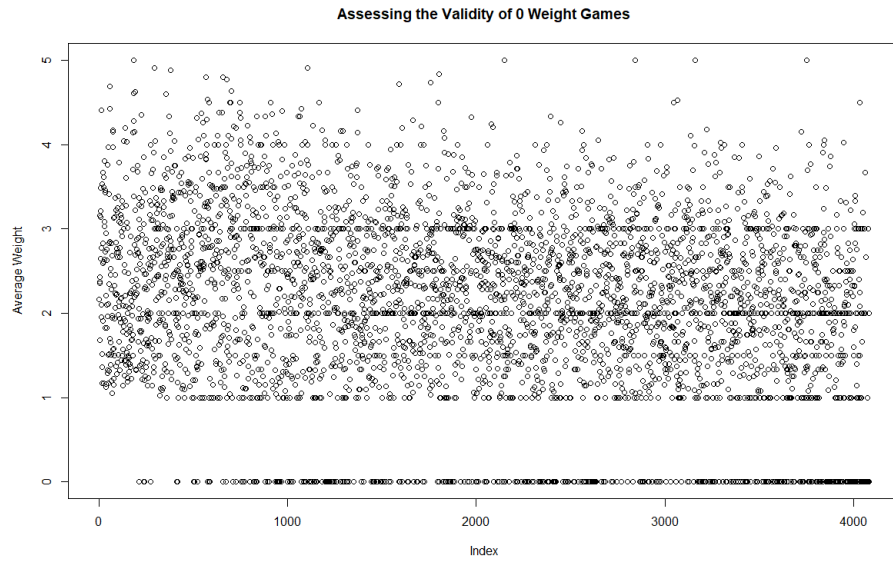


Figure 4: Assessing the Validity of Zero-Weight Games

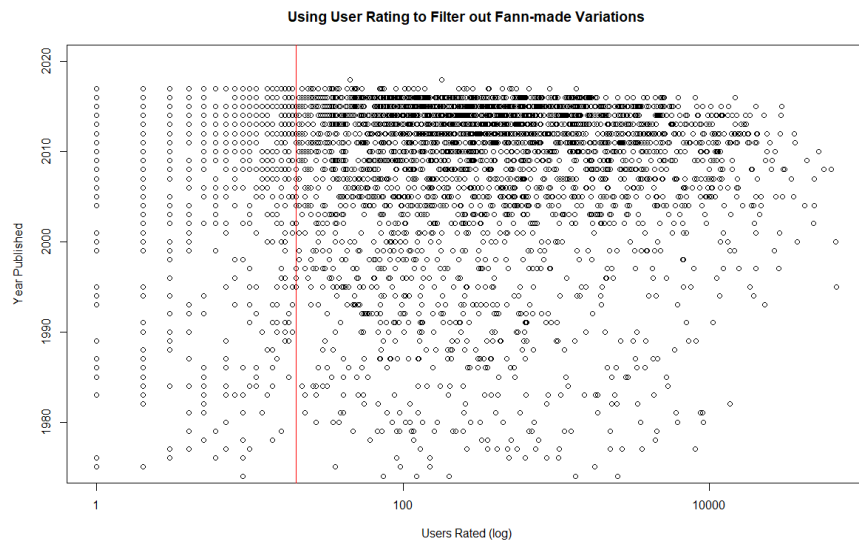


Figure 5: Using Number of User Ratings to Filter out Fan-Made Rule Variations

Once the quality of the remaining data had been verified, analysis continued on to the EDA phase, where the author familiarized themselves with the dataset using basic descriptive statistics, visualization, and assessed variable relationships. The data was augmented according to these observations, before finally proceeding to modelling.

Exploratory Data Analysis

In order to save the reader some time, the general rationale behind every visualization will be provided once, immediately below, instead of with each graph.

univariate: histograms were chosen, as they are more accurate representations of distribution than density plots.

bivariate plots, 1) if both variables were continuous, a scatterplot with an abline set on the linear regression was used in order to illustrate trend, 2) if one variable was continuous, and the other was discrete: a box plot with an added horizontal reference line set to the most important value for the continuous variable (usually the threshold value previously decided for average rating (6.9)) was used in order to see a feature's distribution relative to that threshold.

Correlation: If multidimensional, then corplot, otherwise, simple correlation stats.

Model effectiveness: Confusion matrix visualized via tile plot, with colouration dependant on what percentage of the test sample fell into each category classification. Percentile values were included and coloured red to increase readability. Statistic based performance indicators were included adjacent to the confusion matrix illustration and were scaled from baseline (0.5) to perfect (1.0) with the goal (0.8) highlighted via vertical line.

Basic Descriptive Statistics for Important Features

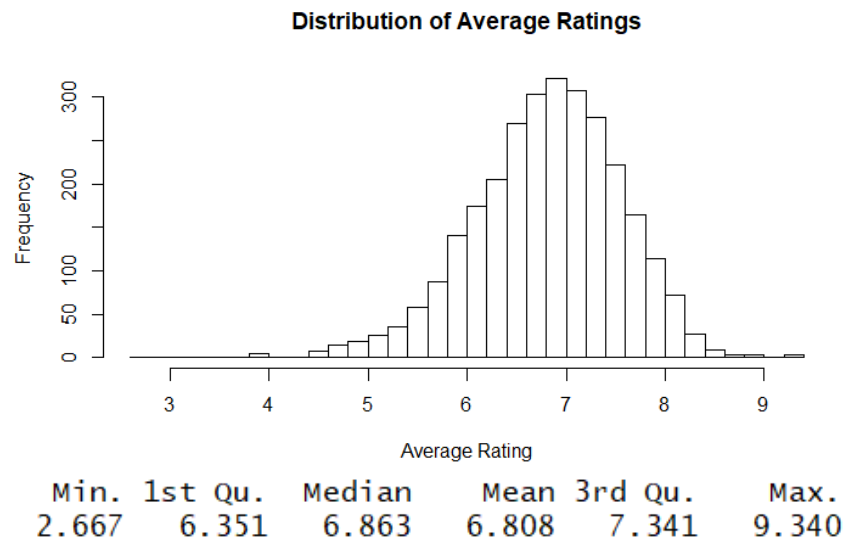


Figure 6: Univariate Distribution of Average Rating

As average rating is the parent input for the primary classification variable, it was examined first, via histogram specifically, in order to assess its distribution univariately. Overall, average rating was unimodal, following an extremely gaussian curve. it's left tail is slightly denser than its right tail, and it is centred, not surprisingly, around 6.8, which is its median and mean. Observations less than 6.9 will be considered below average, while those greater than 6.9 will be considered above average.

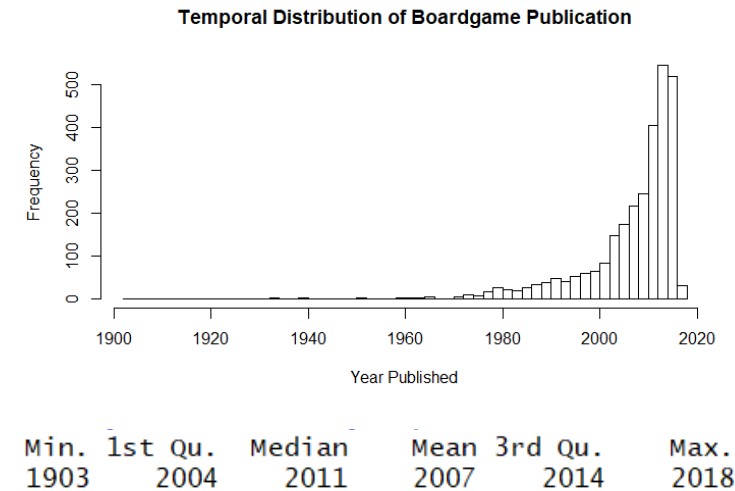


Figure 7: Increase of Boardgame Publication over Time

Observing the histogram of the publication year feature, it's quite clear that the majority of the games in this dataset are published in the late 2000's or later. This mirrors the expected explosion during the boardgame renaissance but may also reveal that board games from before the 2000's, especially those published before the 1980's, are underrepresented. The severe drop-off in frequency post-2016 is owed to the fact that this dataset was scrapped in 2016, making anything more recent than that estimated data.

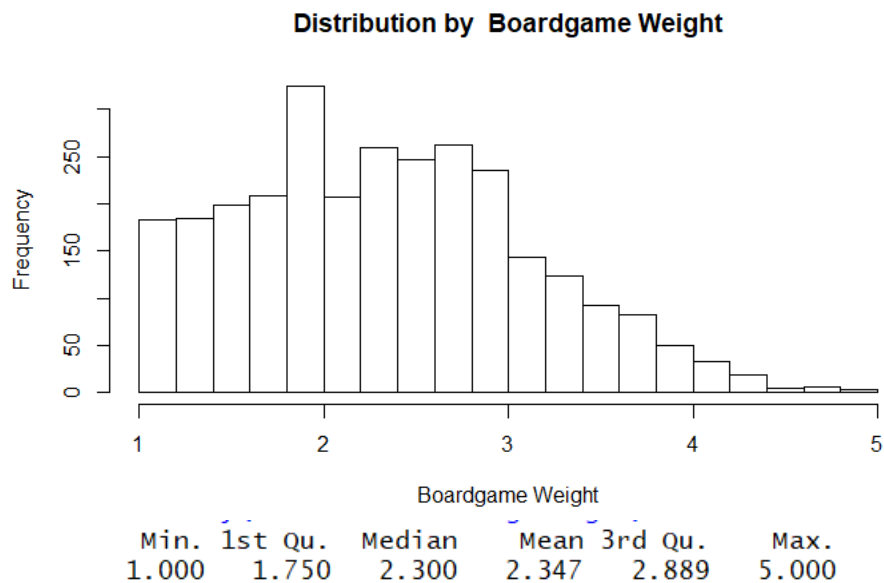


Figure 8: Distribution by Complexity

The majority of boardgames have a weight of between 1.75 and 2.9. There seems to be an unusually large uptick in frequency for games between 1.8 and 2.0. Frequency decreases quite steadily after a weight of 3.

Number of Boardgames by Mechanic

Dice Rolling	Hand Management	Variable Player Powers
989	871	691
Modular Board	Set Collection	Card Drafting
414	386	373
Hex-and-Counter	Area Control / Area Influence	Action Point Allowance System
341	325	296
Tile Placement	Co-operative Play	Simultaneous Action Selection
271	252	236
Deck / Pool Building	Simulation	Grid Movement
215	200	199
Area Movement	Partnerships	Worker Placement
189	189	185
Auction/Bidding	Point to Point Movement	Take That
162	151	144
Player Elimination	Role Playing	Route/Network Building
133	126	115
Variable Phase Order	Secret Unit Deployment	Campaign / Battle Card Driven
114	112	111
Press Your Luck	Pick-up and Deliver	Roll / Spin and Move
111	109	103
Trading	Memory Action / Movement Programming	Pattern Building
103	96	91
Voting	Storytelling	Stock Holding
78	63	57
Betting/Wagering	Commodity Speculation	Acting
45	38	36
Paper-and-Pencil	Pattern Recognition	Area Enclosure
34	34	25
Rock-Paper-Scissors	Chit-Pull System	Line Drawing
32	30	14
Trick-taking	Time Track	Crayon Rail System
22	19	7
Singing	Area-Impulse	7
9	7	

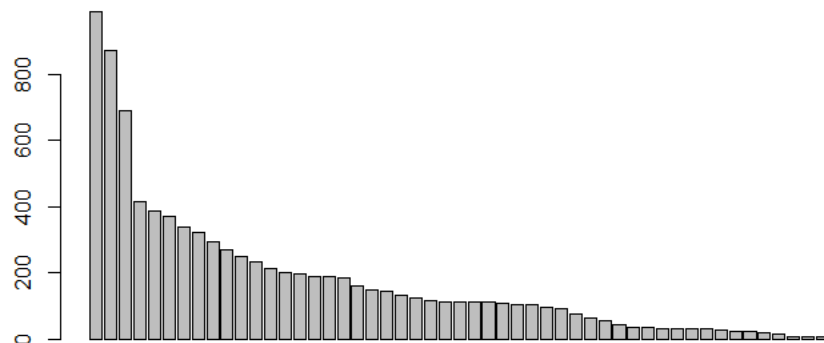


Figure 9: Count of Boardgames by Mechanics

There were 52 mechanics in the dataset. "Dice Rolling", "Hand Management", "Variable Player Powers", "Modular Board", "Set Collection", "Card Drafting", "Hex-and-Counter", "Area Control / Area Influence", "Action Point Allowance System", "Tile Placement" were the most popular. The decrease in frequency is much more gradual than seen in categories (see fig 8)

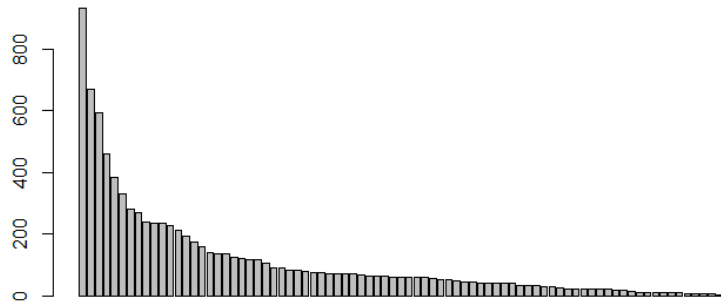
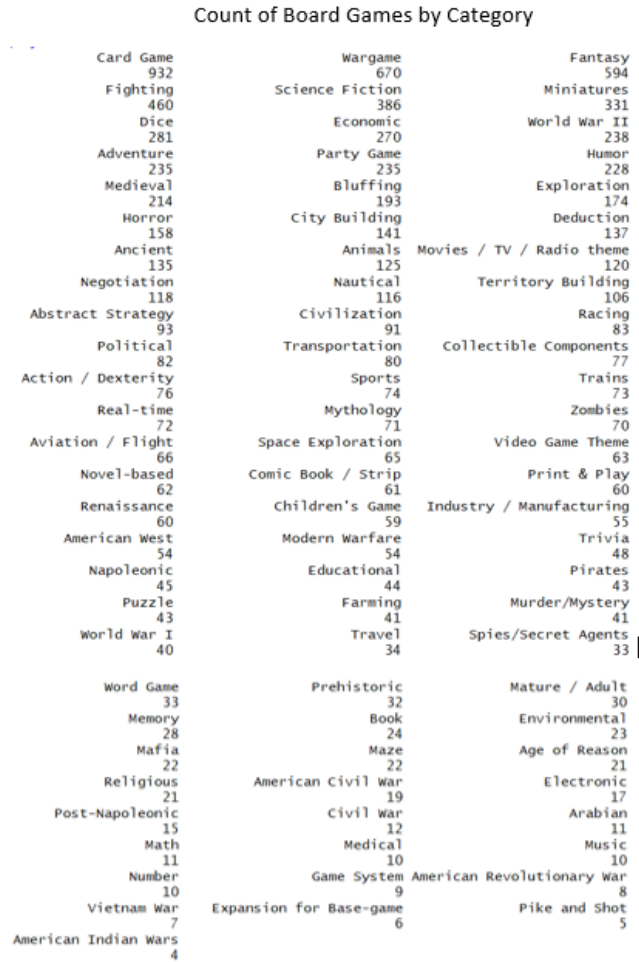


Figure 10: Count of Boardgames by Category

There were 94 categories of boardgame. The ten most popular categories were Card Game, Wargame, Fantasy, Fighting, Science Fiction, Miniatures, Dice, Economic, World War II, and Adventure (see fig 9) . Holding multiple categories was less common than holding multiple mechanics.

Relationships Between Other Features and Average Rating

As average rating forms the backbone of the classification models used in this paper, it was certainly worthwhile to explore its relationship with some of the dataset's other important features. Importance was determined by both correlations discussed later, and functions directed at the Random Forest model. The important features are: minimum age, average weight, number of mechanics, time spread, maximum players, and year of publication .

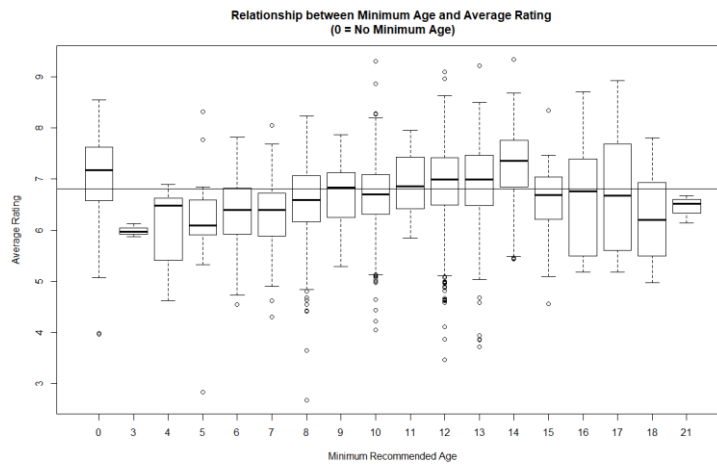


Figure 11: Age Recommendations and Reception

Boardgames without recommended ages seem to be preferred (see Fig 11), and board games with minimum ages under ten or over 15 seems to receive lower average ratings.

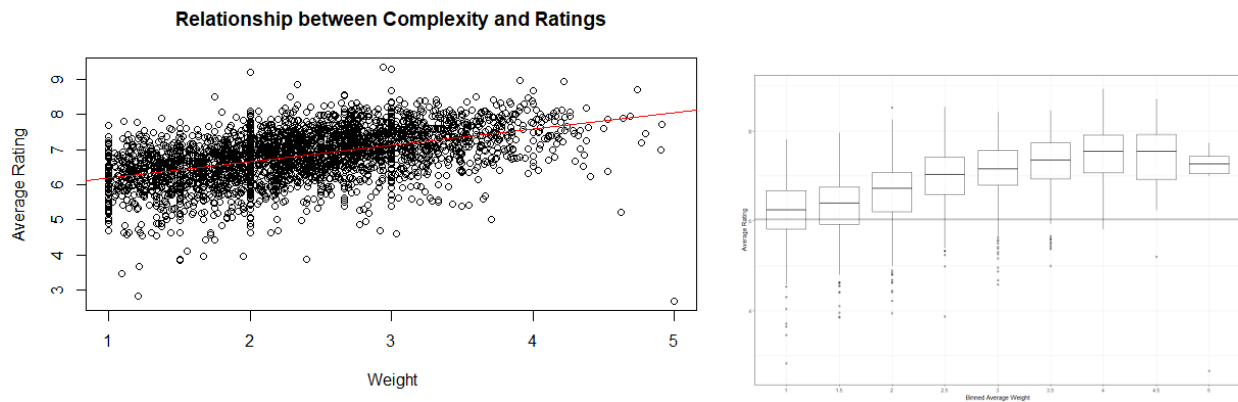


Figure 12: Relationship Between Complexity and Ratings

Generally, as the complexity of a game increases, so does its rating, though this relationship becomes difficult to prove after a weight of 4, as the number of data points becomes quite sparse.

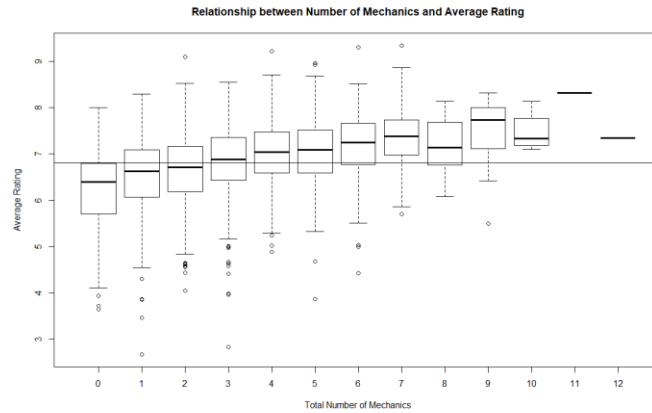


Figure 13: Relationship Between the Number of Mechanics and Average Rating

Backing up the previous figure, as the number of mechanics present in a game increases, so does its probability of being an above average game.

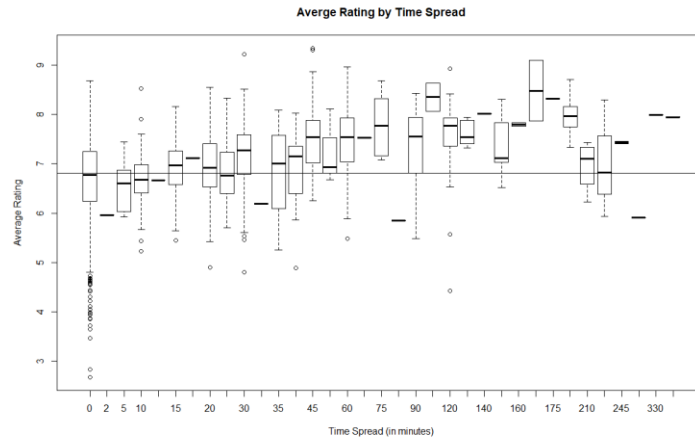


Figure 14 : Relationship between the Variability of Playtime and Average Rating

Generally, the more variable the time required to play a game the more popular.

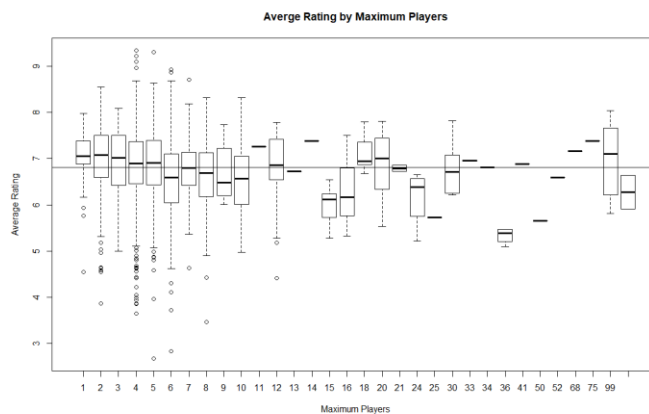


Figure 15: Relationship Between Maximum Player count and Average Ratings

Games with up to (but no more than) five players are more likely to be above-average.

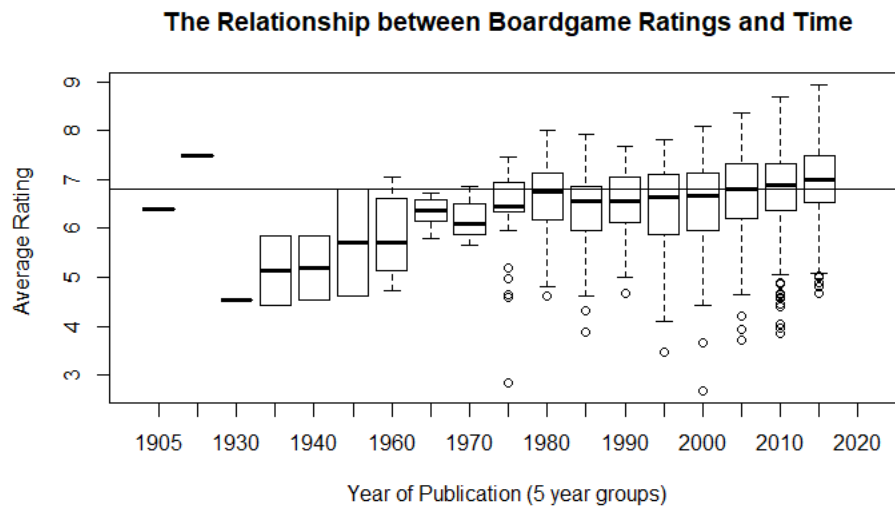


Figure 16: The Relationship Between Boardgame Ratings and Time

The mean scores for games increase over time. Games from before 1970 were (except for one) ill-regarded by BBG users, while games after 2010 were more likely than not to be above average.

Relationships Between Features

In order to assess the correlation of the non-categorical variables, they were placed in a separate data frame, and then plotted using corrplot. The only pairs which had worrying amounts of correlation were player spread to maximum number of players, and between maximum and minimum playtime. Both were expected relationships, since maximum playtime is inherently equal to or higher than minimum playtime, and the potential value for player spread scales linearly with the maximum number of players. The relationship between weight, average rating, and play time was also quite interesting, see fig 18,

As initially shown in fig 17, and then further explored in fig 18, complex games tend to have higher minimum and maximum times. Furthermore, as the play time increases Games with more complexity tend to have higher average scores than their similarly time contemporaries.

Model Creation

The data was subdivided randomly (seed (123)) into a 70/30 training/testing split using the caTools library. The proportion of the successful variable was quickly assessed and found to be representative of the expected value (50/50) for both the training and testing data. This data set was run through the first of three models (Random Forest) five times, with any features deemed less useful than a random number (via the importance function) removed.

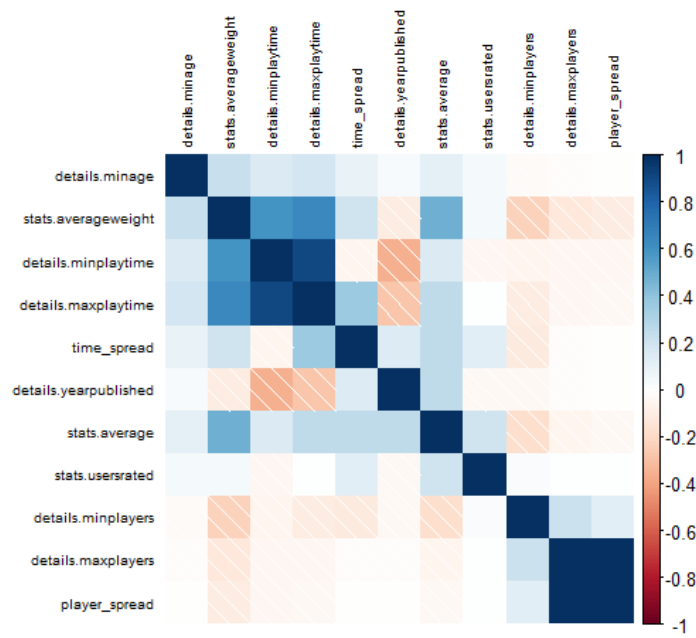


Figure 17: Correlation of Non-Categorical Features

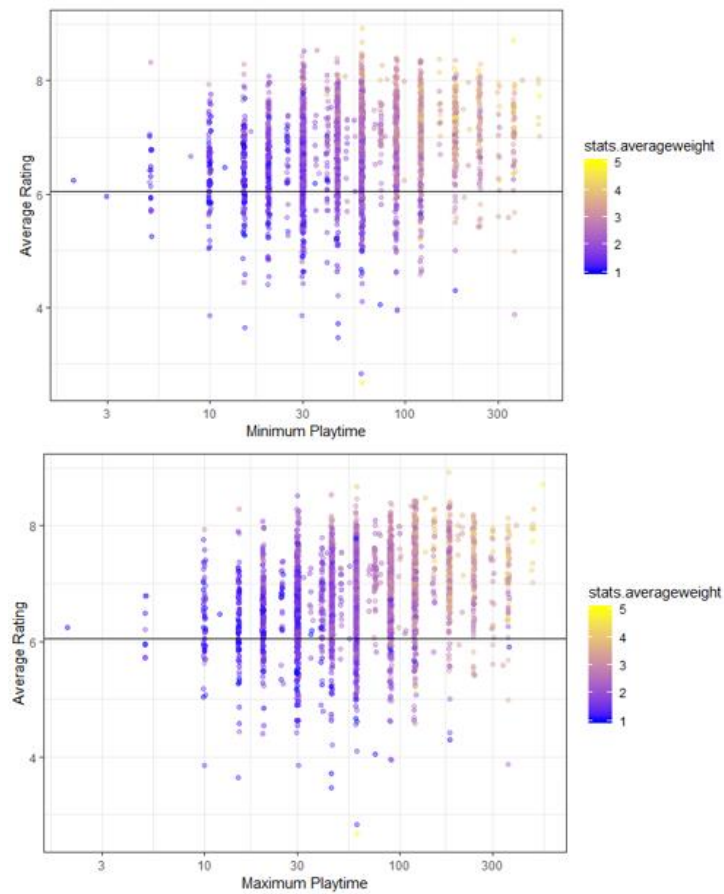


Figure 18: Relationship Between Weight, Playtime, and Average Ratings

This revised dataframe was then passed on to the other two model, which were SVM and Logistic Regression. The rational behind the use of each model, as well as its tuning, and performance are included below, starting with Random Forest. The metrics used to judge success were specificity (true negative rate), and sensitivity (true positive rate), as these represent the ability to correctly sort positives and correctly sort negatives (Witten and Grolemond, 2005). As a result, passing either makes the model useful for at least half the task (increasing the sorting speed by placing only actual successes at the top or removing only actual negatives, which decreases the chances of discarding potential hits before they've been reviewed)

Random Forest

Random Forest deals well non-linearity and complex relationships between the IV and the DVs (Witten and Frank, 2005), it's less easy to interpret than a regular Decision Tree, but this was more than made up for deficit with its ability to automatically assess the usefulness of features. This was used in five iterations for dimension reduction, by removing any features with a negative importance, and then rerunning the model. Unlike the other models, Random Forest was aiming for regression of average rating, and then converting that into a classification. As this relied on Regression, the RMSE was compared to the baseline model, and came out to .54 compared to the baseline .70 of the testing data.

Logistic Regression

As a GLM model, Logistic Regression manages to sort binary classifications via the predicted probability's relationship with a probability threshold value, making it well suited to this task, while also skirting many of the pitfalls of normal Linear Regression. Initially, the model was calibrated to respond positively at a probability of 0.5. the probability was increased by increments of one percent, but this did not improve accuracy.

SVM

SVM models are inherently well suited to classification of high-dimensional datasets, which this certainly qualifies as. They also allow for the inclusion of non-scaled data, and are robust against over tuning of the data, (Wikham and Grolemond, 2016)) which makes them quite ideal for this business problem. The main issue it had to deal with was computation time, or possibly misleading features (Witten and Frank, 2005), but these were alleviated earlier via the Random Forest Dimensional reduction.

Evaluation of the Models

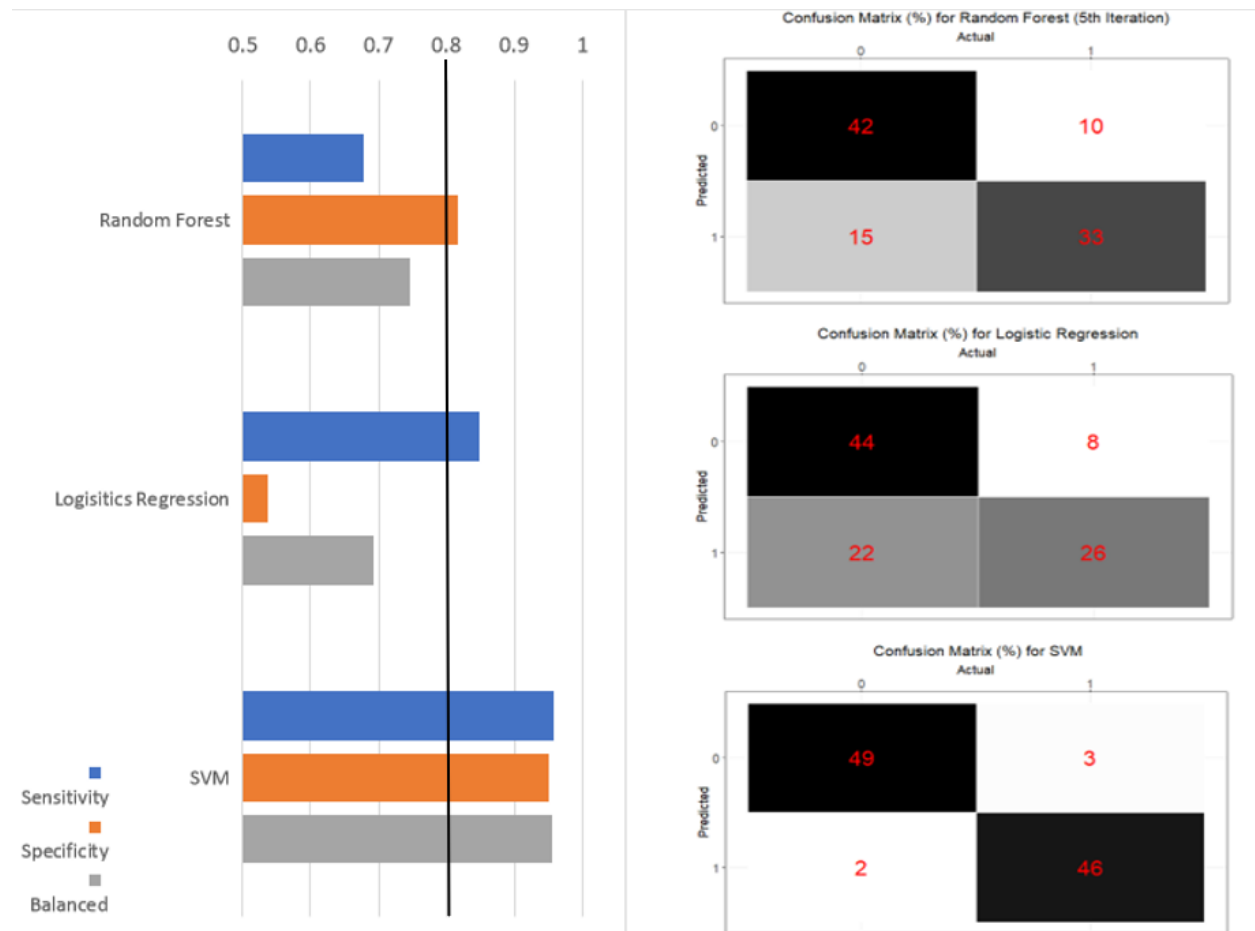


Figure 19: Confusion Matrices and Performance Metrics for each Type of Model

Random Forest and Logistic Regression were comparable in terms of their effectiveness, with Random Forest having a higher specificity, and Logistic regression having a higher sensitivity. The Random Forest model was slightly better overall than Logistic Regression, based on the balanced accuracy score. More SVM model was extremely successful and surpassed all the goals put forward in the analytics problem.

Deployment

Usefulness

The SVM model proved extremely capable of solving both the analytics problem and the business problem. The chances of misclassification are extremely low, as evidenced by the sensitivity, specificity and balanced accuracy scores, which all placed between 0.95 and .0 96. This was a notable overshoot of the target score (established as 0.8) and approached a perfect score of 1.0. Theoretically, CK could implement this to see a tremendous boost in their ability to compete, with limited negative consequences of doing so.

Deployment Strategy

The preferred method of gathering the data required to implement this model would be a self-reported by game designers upon electronic submission of their design, with glossary terms provided to prevent confusion, and possibly tables to record player counts and play times for at least 10 sessions, to ensure that the game has been play-tested and to make the screening process a little more covert to prevent willful gaming of the algorithm. Each self-report would be converted into an observation in a new dataframe, with the dataframe undergoing prediction via SVM when the queue of games to review is empty. Those that classify positively would be placed into the previously mentioned review queue, arranged by the date of submission, in order to implement FIFO. Data storage notwithstanding, those that classify negatively should be stored in case another model with a greater predictive power is developed.

Additional Data Requirements

While the model itself doesn't seem to need much more information, the character-count for the rulebook would be a nice addition, as would the average time required to play a game. Additional algorithms to optimize review order via potential profitability would be quite useful.

Upkeep Requirements

The upkeep required for this model is lax, since the average scores on BGG, which are the principle means of calibrating the model, experience latency between publication and accurately reflecting a game's reception. As a result, it's recommended that the analytic model be updated at the end of every financial quarter, though it could theoretically be updated every time the review queue was emptied, or nearly depleted. Increasing the update frequency past that would cease to offer any benefit.

Conclusion

Using a Random Forest model in order to perform dimensional reduction helped create a highly successful SVM model. Generally, the average rating of a boardgame appears to be quite dependant on the complexity and play time of the game.

References

- Baldassarre, G. (2017). *Board Games Dataset*. Retrieved from Kaggle :
<https://www.kaggle.com/gabrio/board-games-dataset>
- BoardGameGeek. (n.d.). Retrieved from BoardGameGeek: <https://boardgamegeek.com/>
- Browne, C. (2017). Back to Basics. *Puzzle and Game Design*, 3(2), 2-4. Retrieved from
<http://gapdjournal.com/issues/issue-3-2/gapd-3-2-00c-editorial.pdf>
- Martin, T. (2017, January 23). *How Board Games Became a Billion-Dollar Business*. Retrieved from New Statesman: <https://www.newstatesman.com/culture/games/2017/01/how-board-games-became-billion-dollar-business>
- Wickham, H., & Grommund, G. (2016). *R for Data Science: Import, Tidy, Transform, Visualize, and Model Data (1st Edition)*. O'Reiley.
- Witten, I., & Frank, E. (2005). *Data Mining: Prctical Machine Learning Tools and Techniques* (2nd ed.). Amsterdam: Morgan Kaufmann Publishers.
- Woods, S. (2012). *Eurogames: The Design, Culture and Play of Modern European Board Games*. London: MacFarland and Company, Inc, Publishers .