**Springboard capstone project 1**

**Board Game Geek Blog**

Board Game Geek (BGG) is the biggest board game forum for hobbyist and contains a database of reviews, images, and ratings. People or companies may join BGG for free and join in conversations about different games, post questions, or rate their favorite games. There is a lot of data on the site about each game.

**Problem**

With hundreds of new games coming out each year it is hard to not have a game get lost in all the noise and fanfare. How can a fan find games that they will enjoy but haven’t received the hype and publicity that other games have gotten? Can you predict a game’s popularity by the mechanics, theme, and player count?

**Data**

The data that will be used is from Kaggle. ([https://www.kaggle.com/mrpantherson/board-](https://www.kaggle.com/mrpantherson/board-game-data)

[game-data). It was collected from BBG in March of 2017.](https://www.kaggle.com/mrpantherson/board-game-data)

The board game data has 20 columns and 5000 rows. Each column is a characteristic of a board game and each row is a unique board game. There seems to be three types of columns:

1. Qualitative: mechanic, category, designer

2. Quantitative: min\_players, max\_ player, avg\_time, min\_time, max\_time, year, avg\_rating, geek\_rating, num\_votes, age, owned, weight

3. unique rank, bgg\_url, game\_id, names, image\_url

The columns and what data they contain are:

* rank: each game is ranked by popularity from 1 to 4999.
* bgg\_url: url of game on BBG
* game\_id: each game is given a unique id.
* names: name of board game
* min\_players: minimum number of players
* max\_player: maximum number of players
* avg\_time: average time the game takes to play
* min\_time: minimum time to play game
* max\_time: maximum time to play game
* year: year board game was released
* avg\_rating: mean of all board game user ratings
* geek\_rating: BGG rating based on avg\_rating but altered with secret BBG Stats
* num\_votes: number of votes by users
* image\_url: url to picture of board game cover
* age: recommended starting age
* mechanic: list of board game mechanics
* owned: number of users who stated they own game
* category: list of themes or categories of the board game
* designer: list of designers
* weight: average rating of how hard a game is to understand given by users

One problem with the data is some categories are user given such as weight, and rating. Others will be on the board game box but could be entered wrong. These are things that must be taken into account when exploring the data. It isn’t full of hard facts but we must assume most categories are correct.

The following columns were removed because the data would not help a model and were unique:

1. bgg\_url - url to the board game on BBG.

2. image\_url - url to a picture of the board game on BBG.

3. rank - coresponds to geek\_rating

For columns min\_players, max\_players, avg\_time, min\_time, max\_time having a 0 as a value doesn’t make sense. A game has to have at least one player and time has to be greater than 0 to be played. They are only 127 games that have these missing values so they will be removed.

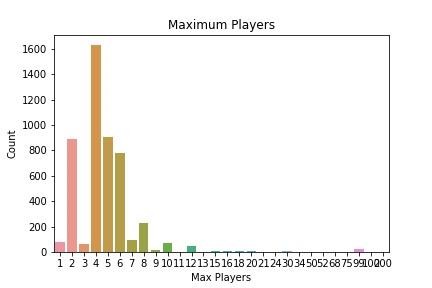


Figure 1

Most games have a maximum player count of 4 with 2, 5, and 6 being the next most common. In the figure 1 there seems to be a a significant amount of games at 99 players. There are only 120 games that have a max of 11 players or more and most times you will not have more than 10 people at a board game party. If I assume that I will also remove any board games that have a minimum player count of more than 10 players. I will remove those.

Some games will last longer than there is time in a day according to the play time listed. There are 1440 minutes in a 24 hour day. These games are probably campaign style games that can be left set up and come back to. Most people can't or won't devote that much time to them.

I will only be looking at games that are less than 8 hours to play. That is a full day of gaming.

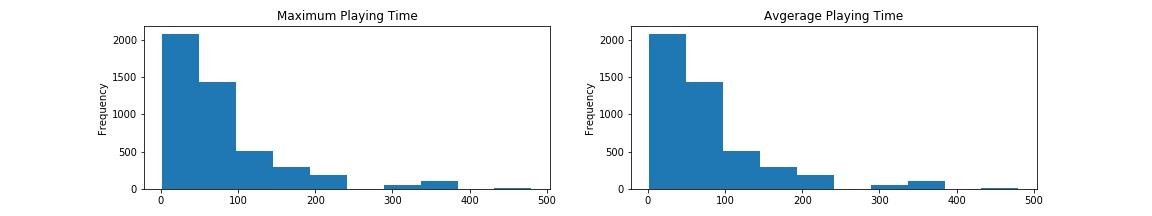


Figure 2

Looking at average time vs. max time in figure 2 we see that they are very similar. If they are it would be redundant to have both average time and max time. We will keep max time.

When we look at the data in the year column 97.5% of board games were released after 1974. I

will be using that has the cut off for my analysis.

**Exploratory Analysis**

There are two different ratings in the data. One is avg\_rating thatis the average of all ratings from registered BGG users that the game has recieved, calculated by adding up all individual ratings.

The second is geek\_rating that the rank column is based on. It is based on the avg\_rating but the number is altered by BGG using an algorithm and beysian statistics to prevent games with relatively few votes climbing up to the top of the BGG ranting.

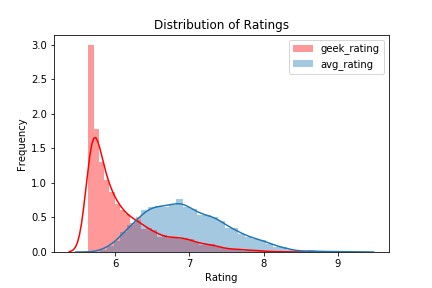


Figure 3

The distribution of the two ratings we see in figure 3 that geek\_rating is skewed towards the

left. This makes sence since the geek\_rating is a ranking system that BGG uses. The avg\_rating has much more of a normalized distribution.

An important part of any board game is how many people can play. There are 2 different columns that describes this: min\_players and max\_players. I added a new column by combining the min and max players to form a player range. The most popular player ranges is 2-4 players with 2-2 player games as the surprise. I don't play a lot of 2-2 player games but they are popular to make.

The average rating of 1-1 player games are the best rated but also have the fewest board games with 50 or more counts. 2-4 player games which have the greatest amount have a wide range of ratings

Time is the ultimate currency for a board game. Are we going to spend all night to play one board game or are we going to play lots of shorter games to maximize the amount of exposure.

Play time ranges for most games are less than 2 hours. Also a lot of the games have the same min\_time and max\_time.

The weight column has an average rating by users on how complex a game is. It is a scale from

0-5. We need to keep in mind that BGG is user driven data and that the data might be skewed towards people who are involved in hobby. The hobbyist might not think a game is complex and rate is lower. But we will take the weight as a good measure. The weight distribution seems to be a normal distribution skewed a little to the left.

Looking at the relationship between age and weight we see that there are a lot of games that have an age of 0 but have a wide range of weight. This is weird. Babies will not understand a game with complexity over 4. I think these are missing values. We will replace all 0 in the age column with the mean of the age column

The owned column has the same shape as num\_votes. There is probably a strong correlation between the two. This makes sense because people vote on the games they own.

We can see that there are 52 different Mechanics. We combined all mechanics of board games of 250 or less into a column of other\_mechanic

96% designers haven't made more than 10 games. It seems that the designer category will have very little influence on predicting the rating of a game and will be dropped from the data frame.

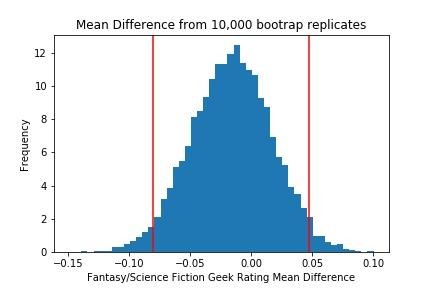
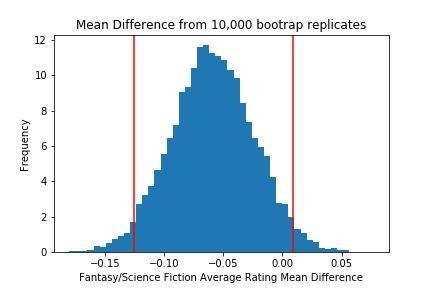
There are a lot of different themes of board games. Two of the most popular themes are Science

Fiction and Fantasy. The question that I have is if there is a statistical difference between the two themes average score or geek score. We see that the avg\_rating and geek\_rating between Fantasy and SciFi board games are very close. Using bootstrap replicates we will see if there is a statistical difference between the two.

The null hypothesis is that there is no difference between the two themes of board games while hypothesis claims there is a difference.

H0 = fantasy ratings = scifi ratings OR fantasy ratings - scifi ratings = 0

Ha = fantasy ratings != scifi ratings



The above histograms is the mean difference in bootstrap deviation of 10,000 samples. The red lines mark the 95% confidence interval. We see that the 95% confidence interval does contain 0 on both avg\_rating and geek\_rating. This means we can accept the null hypothesis that the avg\_rating and geek\_rating of the themes Fantasy and SciFi have no statistical difference.

**Model**

Now that the data is cleaned and organized regressor I will be building a predictive model with sklearn algorithms. I will be trying to predict the avg\_score of each game using regressors. The avg\_score of a game is the average of all ratings given by users on a perticular game. It hasn't been modified by board game geek like geek\_score has.

I will scale the data using the normalization function. Scaling creates non-dimensional features so that those features with larger units do not have an undue influence on the classifier as would be the case if the classifier uses some sort of distance measurement as a similarity metric.

I selected Linear Regression, RandomForestRegressor, DecisionTreeRegressor, and KNeighborsRegressor. I will be using RandomizedSearchCV and GridSearchCV to tune the algorithms and find the best algorithm to use.

I will be looking at Mean Absolute Error as the score parameter. I will look at the standard deviation of all folds to see if there is much variance between the folds scoring. I will also look at the average performance of each model.

I have split the data into 4 different feature sets to see which one is the best for my model.

features\_1 = ['min\_players', 'max\_players', 'min\_time', 'max\_time', 'year',

'num\_votes', 'age', 'owned', 'weight',

'Action Point Allowance System', 'Area Control / Area Influence',

'Area Movement', 'Auction/Bidding', 'Card Drafting',

'Co-operative Play', 'Deck / Pool Building', 'Dice Rolling',

'Grid Movement', 'Hand Management', 'Hex-and-Counter', 'Modular Board',

'Point to Point Movement', 'Route/Network Building', 'Set Collection',

'Simulation', 'Simultaneous Action Selection', 'Tile Placement',

'Variable Player Powers', 'Worker Placement', 'other\_mechanic',

'Abstract Strategy', 'Adventure', 'Ancient', 'Animals', 'Bluffing',

'Card Game', 'City Building', 'Deduction', 'Dice', 'Economic',

'Exploration', 'Fantasy', 'Fighting', 'Medieval', 'Miniatures',

'Party Game', 'Science Fiction', 'combined\_war', 'other\_category']

features\_2 = ['weight', 'year', 'num\_votes', 'owned', 'max\_players',

'combined\_war']

features\_3 = ['weight', 'year', 'num\_votes', 'owned', 'max\_players',

'combined\_war', 'max\_time', 'age', 'min\_time',

'Deck / Pool Building', 'Miniatures']

features\_4 = ['weight', 'year', 'num\_votes', 'owned', 'max\_players',

'combined\_war', 'max\_time', 'age', 'min\_time',

'Deck / Pool Building', 'Miniatures', 'min\_players',

'other\_mechanic', 'other\_category', 'Dice Rolling',

'Simulation']

I have split the data into train and test areas and created a kfold object to have the data split into 10 folds. After running all the different models with Random Grid Search CV or Grid Search CV we see the results in the following table:

| **name** | **training\_score** | **validation\_score** |
| --- | --- | --- |
| **0** | lr\_features\_1 | 0.300 (0.001) | 0.305 (0.012) |
| **1** | lr\_features\_2 | 0.317 (0.002) | 0.318 (0.016) |
| **2** | lr\_features\_3 | 0.310 (0.002) | 0.311 (0.015) |
| **3** | lr\_features\_4 | 0.307 (0.002) | 0.309 (0.015) |
| **4** | dtr\_features\_1 | 0.252 (0.001) | 0.298 (0.011) |
| **5** | dtr\_features\_2 | 0.247 (0.001) | 0.296 (0.009) |
| **6** | dtr\_features\_3 | 0.250 (0.001) | 0.295 (0.010) |
| **7** | dtr\_features\_4 | 0.250 (0.001) | 0.297 (0.008) |
| **8** | rfr\_features\_1 | 0.154 (0.000) | 0.260 (0.006) |
| **9** | rfr\_features\_2 | 0.178 (0.001) | 0.267 (0.008) |
| **10** | rfr\_features\_3 | 0.163 (0.000) | 0.262 (0.007) |
| **11** | knn\_features\_1 | -0.000 (0.000) | 0.342 (0.009) |
| **12** | knn\_features\_2 | -0.000 (0.000) | 0.289 (0.009) |
| **13** | knn\_features\_3 | -0.000 (0.000) | 0.293 (0.011) |
| **14** | knn\_features\_4 | -0.000 (0.000) | 0.315 (0.006) |

We see that Random Forest Regressor with Features 1 was the best Mean Square Error and the best parameters to use are:

{'clf\_\_n\_estimators': 195,

'clf\_\_min\_samples\_split': 11,

'clf\_\_min\_samples\_leaf': 2,

'clf\_\_max\_features': 'auto',

'clf\_\_max\_depth': 25}

Now we are ready to predict the average rating for games using the best estimator we found. By doing so we predict the test data and a Mean Absolute Error of 0.498 was found. This is higher then any of the training scores but still low so we can say the model was a good predictor of average score.