Assignment – 1

I have scrapped the data from the Boardgamegeek.com website. For this, first I used the url and initiated it with beautiful soup. There I found all the data viz.., Name, Rank, Geek Rating, Average rating and Number of voters. Price data was not scrapped, as on inspection I found that the data of the price is not being read by the soup. I appended all the collected data into a csv file. All the data is submitted in the csv named “boardgamegeek\_dataset.csv”. In this dataset we can see there are 17000 data points. Although, the boardgamegeek.com has 106600 games, after 17000 games there is no rank available for the further games. This led to a missing data which led to the website not allowing us to scrape more data. So, this led me to stop scraping at the 170th page which has a total of 17000 data points. Removing the rank from this scraping I have scraped a second data set which has all 1066 pages with name boardgamegeek\_dataset3. But I did the machine learning on the first dataset named baordgamegeek\_dataset with the 17000 data points.

I have first uploaded this data to read. First, I read the columns. IN these columns I have found that the data of a column is not needed. So, I eliminated the column which is not needed by using the “drop” command. Then I plotted the data on a histogram. I have average rating for this. Then I employed the kmeans model to cluster the data. This led the data to be clustered into 5 different groups as shown in the figure below.

Then I conducted a principal component analysis to increase the speed of the algorithm. Then I plotted the data with average rating on Y-axis. Then I checked the correlation of average rating with all other data. The values are all as expected. Then I looked at the relation between Geek rating and average rating. According to the kaggle.com user explanation the difference between geek and average rating is explained as

“The User Rating of a game is the Average Rating rounded down to 1 decimal point. The Average Rating for a game is the average of all ratings from registered BGG users that the game has received, calculated by adding up all individual ratings and dividing by the number of ratings. The BGG Rating is based on the Average Rating, but the number is altered. BoardGameGeek's ranking charts are ordered using the BGG Rating. To prevent games with relatively few votes climbing to the top of the BGG Ranks, artificial "dummy" votes are added to the User Ratings. These votes are currently thought to be 100 votes equal to the mid range of the voting scale: 5.5, but the actual algorithm is kept secret to avoid manipulation. The effect of adding these dummy votes is to pull BGG Ratings toward the mid range. Games with a large number of votes see their BGG Rating alter very little from their Average Rating, but games with relatively few user ratings will see their BGG Rating move considerably toward 5.5. This is known as "Bayesian averaging" and a quick search of both BGG and/or the Web will reveal much discussion on the topic. In effect, usually the games with many votes will Rank higher than those games with the same Average Rating but fewer votes.”

Then in the total data I have split the data into two parts. One is test data and other is training data. 25% of the data is made test data and the rest 75% is training data. With this I have run a Linear Regression model fitting the training data. Finally, to justify the procedure is right I, ran a distribution plot of error terms, which turned out to be perfectly normal. This shows the prediction is right in a sense.