Assignment – 1

1. **Data Scraping**

I have chosen to scrape 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. There is td and class in which the price data is present. Then there was a loop of 3 divs to get to the price value. When I was trying to read the data in soup all it showed was tags from <div> to <div> but couldn’t read the price data. So, I couldn’t scrape the price data.

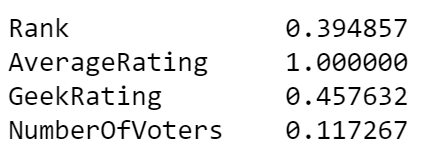
With the rest of the variables, I appended all the collected data into a csv file. All the data is submitted in the csv named “boardgamegeek\_maneesh.csv”. In this dataset we can see there are 106592 data points. Although, the boardgamegeek.com has 106600 games, after 17000 games there is no rank available 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. This is named boardgamegeek\_dataset.

Removing the rank from this scraping I have scraped a second data set which has all 1066 pages with name boardgamegeek\_maneesh. All the learning was done on this file. In this data I had many cells which have N/A as data value. I removed all these data points. The final data size is settled down at 22,926.

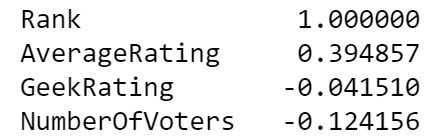
1. **Machine Learning**

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.

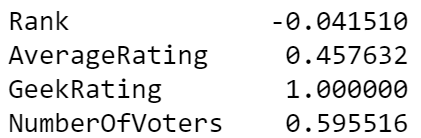
First, I checked the correlations for the variables with all other variables. I checked the correlation for AverageRating, GeekRating and GameRank. The results are given below.



Correaltion og AverageRating with other variables



Correaltion of Rank with other varibles



Correlation of GeekRating with other variables

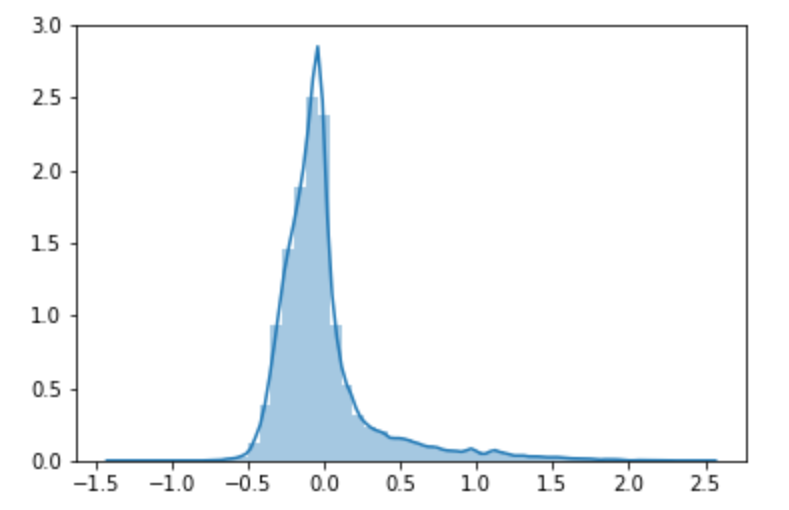
The difference between geek and average rating is given by the explanation below taken from kaggle.com:

“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.”

The average rating seemed a weird metric as the data has a negative correlation with number of voters. As the voters increase the rating is decreasing. But GeekRating has a positive correlation with number of voters. So, I finalized that GeekRating can be a better metric to run the supervised/unsupervised learning on.

Under the **Supervised Learning,**

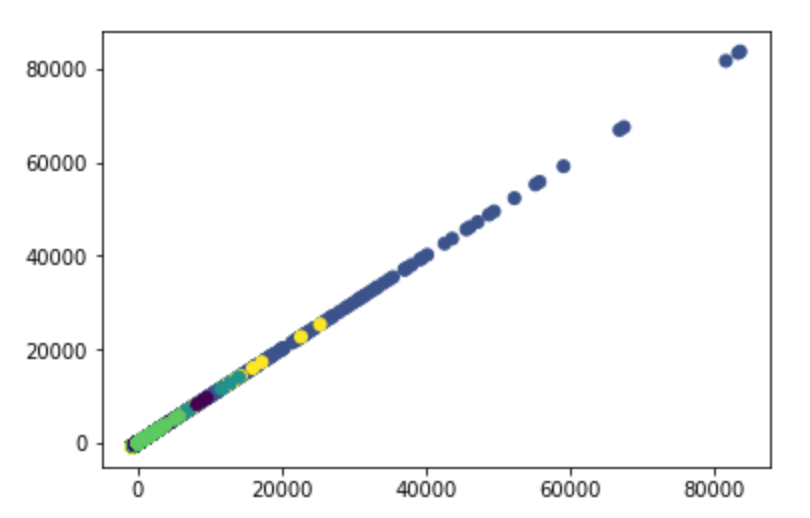
The total data I have split the data into two parts. One is test data and other is training data. 30% of the data is made test data and the rest 70% is training data. With this I have run a Linear Regression model fitting the training data. Finally, to justify the procedure is right I, ran a distribution plot of error terms, which turned out to be perfectly normal. This shows the prediction is right in a sense.



Also, the mean squared error value I found was 0.006, which is pretty close to zero, justifying the procedure conducted.

In the **unsupervised learning,**

I have conducted the normal k-means clustering with Geek-Rating as my target variable. After running the clustering with respect to eek-Rating correlation with all the other variables, the output is as below. The k-mean has 5 clusters within the data.



On the whole the exercise looked at relating Geek\_rating with other variables of interest. The handicap I had was being unable to scrape the price data. If I could have got the price more interesting questions could have been studied like price vs ratings, price vs number of voters.