**IE 7374 Machine Learning in Engineering**

**Team 8 Lab 3 Report**

**Team Member: Yaodong Chen, Lin Zhu, Yuxiao Ma**

**Regression Analysis**

**Dataset description**

* board\_game.csv: The dataset contains 8000 board games and their associated review scores. The information include id, name, year-Published, minPlayers, maxPlayers, playingTime, minPlayTime, maxPlayTime, minAge, usersRate, averageRating, totalOwners, totalOwners, totalTraders, totalWanters, totalWishers, totalComments, totalWeights, averageWeights. Each row represents a single board game, and has descriptive statistics about the board game, as well as review information.
* Some column preview:
  + name -- name of the board game.
  + playing\_time -- the playing time (given by the manufacturer).
  + min\_playtime -- the minimum playing time (given by the manufacturer).
  + max\_playtime -- the maximum playing time (given by the manufacturer).
  + min\_age -- the minimum recommended age to play.
  + users\_rated -- the number of users who rated the game.
  + average\_rating -- the average rating given to the game by users. (0-10)
  + total\_weights -- Number of weights given by users. Weight is how “deep” involved a game is.
  + average\_weight -- the average of all the subjective weights (0-5).
* Our goal is to predict average\_rating through prediction models using all other columns.

**Major steps**

**Data Cleaning**

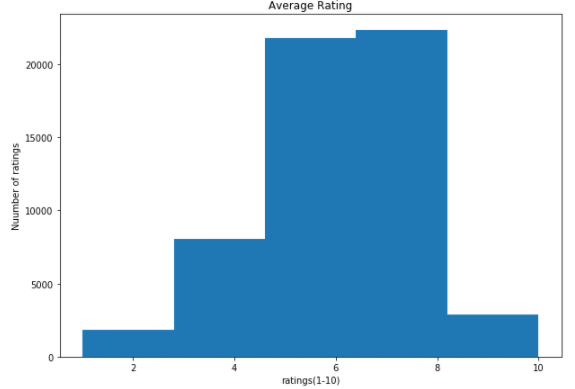
* Read board\_game.csv into a dataframe called board\_game using the Pandas library.
* Print out the first few rows to check data.
* Use dropna method to remove data with missing values.
* Remove rows where users rating is 0.

**Data exploration**

* Select an error metric to evaluate the performance of machine learning model.

We use root mean squared error to evaluate the performance of the prediction model because this is not a classification problem, the prediction will return continuous values.

* Create a histogram of the average\_rating column using the hist function.



From the histogram above, we observe that the majority of the rating is in the range 6-8 points. The mean of the average rating is also in this range.

* Calculate the standard deviation of the average\_rating column and print it out.

The standard deviation of the average rating is 1.578829934833296.

* Calculate the mean of the average\_rating column and print it out.

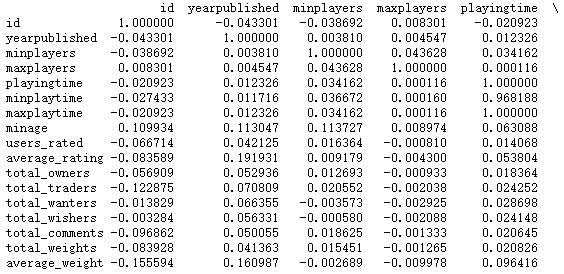
The mean of the average rating is 6.016112849333849.

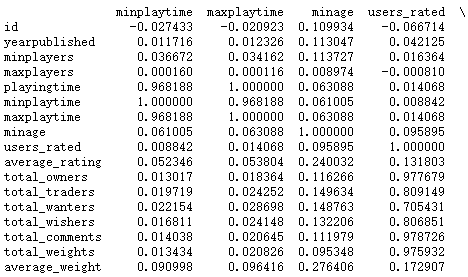
* Think about what error metric might make sense for this data and write a markdown cell with thoughts. List out the reason of picking this error.

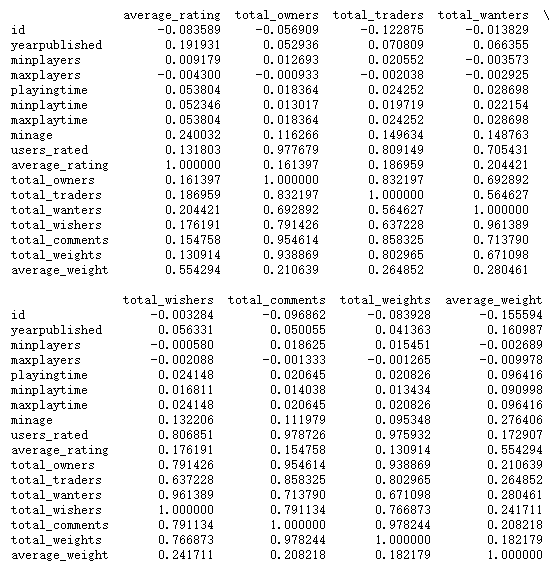
When we are dealing with classification problems, we usually use confusion matrix as our error matrix to represent the error and accuracy. But in this lab case, the y is not about classification. Instead, we will have continuous values as our returned value. We can use mean squared error to compute the error of continuous result intuitively.

**Data correlation and level of importance**

* Use the corr method on numeric\_columns to compute correlations between columns. Assign the result to correlations.

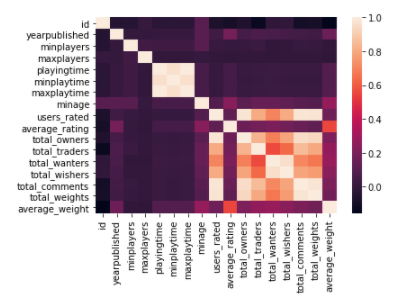






The correlation matrix above cannot be expressed intuitively under multi-features conditions. Therefore, we use heatmap to visualize our correlation matrix to get a better understanding of our data.

* Print out the average\_rating column of correlations.



* Do any of the correlations surprise you? Write up your thoughts in a markdown cell.

Playing\_time shows a strong correlation with min/max play time. Total\_owners shows strong correlation with traders/wanters/wishers/comments/weights. We are surprisingly to see that users\_rated shows a median correlation with total\_wanters and total\_wishers. People who don’t own the game actually rated the game. The rating could be biased.

* Figure out which columns, if any, you want to remove.

We will remove min/max players, playing time and min/max playing time since they do not show a very positive correlation with average\_rating feature.

* Make Insights through correlation plots, what do you observe from data?

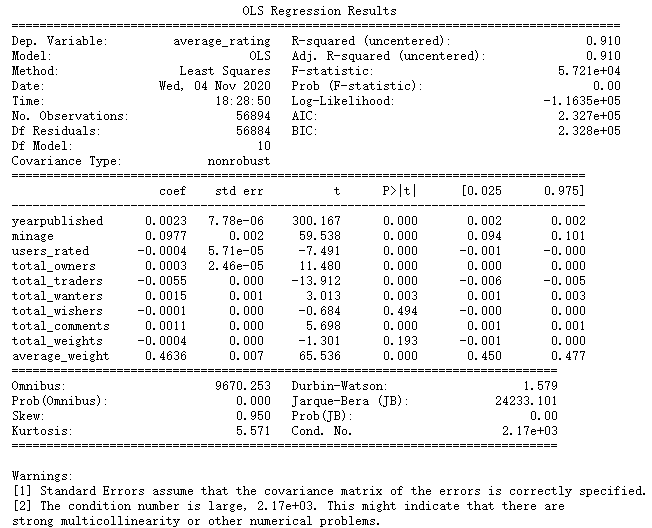
From the heatmap above, we get a better view of correlations between average ratings and other features. When the number gets closer to 1, it indicates a strong correlation. We observe that average rating has a mild correlation with average weight, a small correlation with minAge.

**K-fold cross validation**

* We performed 5-fold cross validation to our dataset and the RMSE value is [1.95467498 1.52620466 1.58238126 1.58153637 1.95859201] respectively. When we compare the RMSE value to our linear regression model, we conclude that linear regression performs better for model training.

**Ordinary Least Squares and Gradient Descent Method**

* We performed OLS model and here is the result:



* We also implemented gradient descent method to make prediction, the result returns:



* When we set the learning rate to 0.001 and tolerance to 0.5

the RMSE value returns 6.188072882411132

* When we set the learning rate to 0.05 and tolerance to 0.1

the RMSE value returns 6.187945535436197

* When we set the learning rate to 0.1 and tolerance to 0.01

the RMSE value returns 6.187945536177765

**Lasso Cross validation and Ridge Cross Validation**

* We perfomed LassoCV() for 11 regularization with default parameters and fit the model. The RMSE value is:

****

* We performed RidgeCV() from linear\_model in order to perform 12 regularization. The RMSE value is:

****

* By comparing the RMSE values, we find out that RidgeCV() performs better than LassoCV().

**Random Forest Regression and Neural Network**

* We implemented Random forest regressor and neural networks. Below is the result of their performance (RMSE value):





* The first is the result of random forest regressor, the second is the neural network. By comparing the values, we find out neural network has a higher accuracy and lower RMSE value. In this case, neural network performs better.