# ECON900 Problem Set 1 Summary

**Guo Li**

For this problem set, I chose boardgamegeek.com as my web-scraping target.

At the first glance, I found that the website has a nice table for all the information that I need to collect, including names, ratings, voter number and prices. However, this table has more than 1000 pages and contains more than 100,0000 rows. More importantly, the structure of the html files is quite different from what I learned in class from coinmarketcap.com.

For example, the columns “Geek Rating”, “Avg Rating”, “Num Voters” are not clearly identified in the html files and have almost identical characteristics.

After some trial and error, I managed to parse all the required data. The data categories are: “Rank”, “Name”, “Geek_Rating”, “Avg_Rating”, “Votes” and “App_Price”. I also included a column “Parsed From” to track the original html file that each row of data comes from. Detailed data description can be found in the notes of the parsing file “boardgamegeek.com_parse.py”. There are a total of 106419 rows of data in the dataset, though many of them have a null value.

When I first examined the parsed csv file, everything looks normal. Data categories and entries are displayed clearly. Unfortunately, when I put the data in pandas dataframe, I found that all the data numbers shown in the csv file are marked as objects. I tried to convert these strings into floats using different methods, but none work. In the end, I had to create a big excel file to convert these numbers

and that's definitely not the most efficient way.

Due to this limitation for the data, I only performed a simple linear regression as my machine learning analysis. The data categories I used are "Votes", "Geek_Rating" and "Avg_Rating". Entries with "Votes"<30 are dropped because a valid Geek Rating requires as least 30 votes.

The question that I wanted to answer in this analysis is as follows: suppose a board game firm had a new game introduced and wanted the game to be listed as a top 100 game on boardgamegeek.com, what level of average user rating do they need to achieve (assume they can impact users rating with some marketing strategy)? What if their goal is to be list as top 50, top 25 or top 10?

To answer this, I just need to find out the linear relationship between "Geek_Rating" and "Avg_Rating". We can observe the target Geek_Rating for different rankings and use the linear regression model to predict the Avg_Rating that is necessary to achieve the goal.

After running the program "regression.py", I got the following results:

To achieve a top 100 ranking ("Geek_Rating"=7.447), it is predicted that the game needs at least 8.683 average user rating.

To achieve a top 50 ranking ("Geek_Rating"=7.651), it is predicted that the game needs at least 8.925 average user rating.

To achieve a top 25 ranking ("Geek_Rating"=7.889), it is predicted that the game needs at least 9.208 average user rating.

To achieve a top 10 ranking (“Geek_Rating”=8.072), it is predicted that the game needs at least 9.425 average user rating.

I’m hoping to solve the data string problem and then I should be able to perform more machine learning techniques using the complete data.