Name: Daniel Adigun

Econ: 9000

Data collection description

Data was scrapped from https://boardgamegeek.com/browse/boardgame, Name, Rank, Average Rating, Geek Rating, Votes details were extracted for 5000 distinct games.

Descriptive statistics

	Rank	Arating	Grating	votes
count	5000.000000	5000.000000	5000.000000	5000.000000
mean	2500.500000	7.006868	6.112275	2283.710000
std	1443.520003	0.557323	0.482809	5279.456839
min	1.000000	5.800000	5.656000	66.000000
25%	1250.750000	6.590000	5.750000	345.000000
50%	2500.500000	6.950000	5.932000	731.500000
75%	3750.250000	7.370000	6.325000	1847.250000
max	5000.000000	9.170000	8.611000	84569.000000

ORDER IN WHICH DATA SCRAPPED

1. Load and run board_request.py, this file gets you the html for the first 50 pages of the boardgamegeek website and stores it in a folder name html_files
2. Load and run ratings_parse.py, this file scrapes the geek ratings, average ratings and the votes for all 5000 game into a csv file called boardgame_ratings_data.csv located in the parsed_results folder.
3. load and run board_info_parse_and_merge.py, this file scrapes the Rank and Name of all the games, stores them in a csv file called boardgame_info_data.csv, the code then has a extension that merges boardgame_ratings_data.csv and boardgame_info_data.csv together to create the complete dataset called boardgame_dat.csv. there is an additional extension of this board_info_parse_and_merge.py that then drops the Name column so we ccan have a dataset that containes only numbers which we would use for our analysis, the name of this final dataset is called boardgame_data.csv

DATA ANALYSIS USING supervised learning models A series of analysis were performed on the ddtaset testing different machine learning models and evaluating the efficicncy of each of these models

1. we started with the basic OLS linear regression with the file board_ols.py and predicted a range of values with this model

2. We then used Kneighbors Classifier board_knn.py to analyse the data and predict the same range of values using the same input.

3. Then we run a pair of unsupervised learning models and compare the results and accuracy of each of the model on our data set, the models used are file board_decision.py and board_randomforest.py

Analysis

This project utilized data obtained from the readme file on the repositories. with the data scrapped we ran a set of supervised and unsupervised models in other in to predict the dependent variable (Rank) using data about r2_square and accuracy test to see which gives the desired results and to what degree of certainty or accuracy.

after running both the OLS and KNN models we precited the rank by inputting these numbers X = [

[7.0,15367],

[7.7,50235],

[6.5,7842],

]

and printing out the predicted rank based on each of the parameters included in the prediction model.

1. for OLS we got prediction values of [461.33105654 23.95626287 1602.42956393] for a game with the details given in X.

2. for the knn model using the same set of X, we got predicted values of [42 36 161] for the same exact values of X defined above.

I tried calculating the accuracy values of each model to determine our good our prediction was but I go zero for both models.

Also ran supervised learning algorithms(decision tree and random forest) and obtained accuracy scores for both models, values gotten were 0.009 and 0.004 respectively, this doesn't say much about the usefulness of the models run.