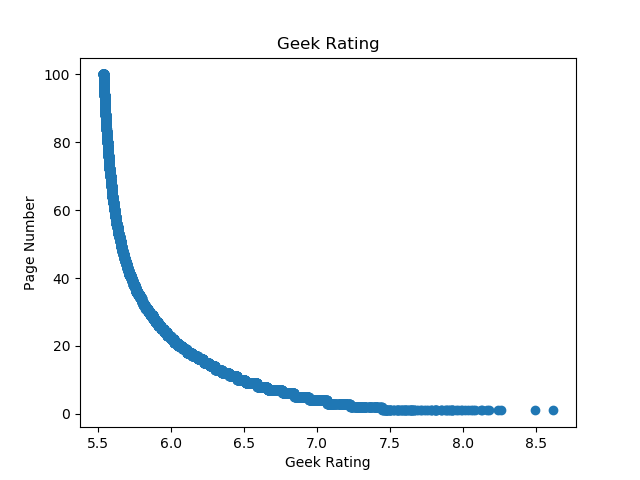
Problem Set 1 Writeup

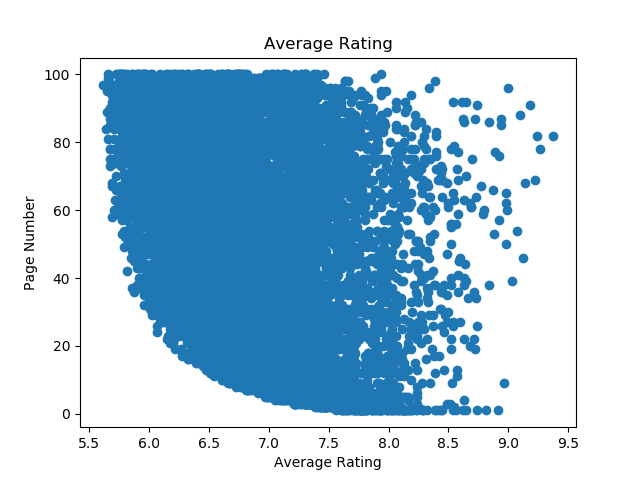
In part one, I scrapped data on the page number, title, geek rating, average rating, and number of votes from the first 100 pages of board game geek. This scrapping gave me 10,000 entries with 5 pieces of data: page number, title, geek rating, average rating, and number of voters. Below is the summary statistics.

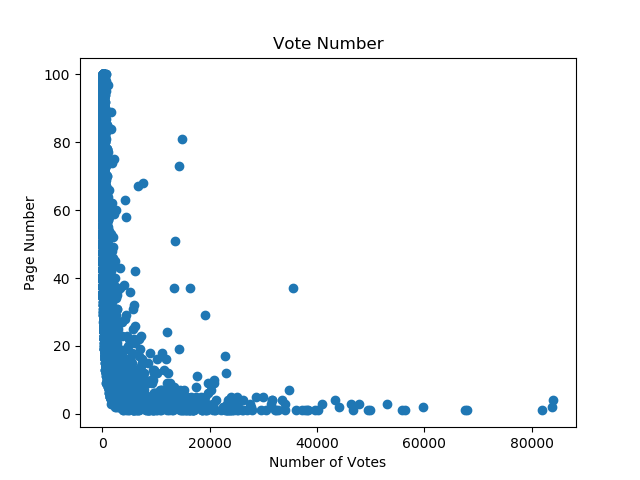
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | page\_number | geek\_rating | avg\_rating | vote\_number |
| count | 10000.000000 | 10000.000000 | 10000.000000 | 10000.000000 |
| mean | 50.500000 | 5.844962 | 6.861724 | 1211.097900 |
| std | 28.867513 | 0.432923 | 0.608268 | 3864.424472 |
| min | 1.000000 | 5.537000 | 5.620000 | 30.000000 |
| 25% | 25.750000 | 5.572000 | 6.400000 | 100.000000 |
| 50% | 50.500000 | 5.654000 | 6.800000 | 238.000000 |
| 75% | 75.250000 | 5.930000 | 7.260000 | 750.000000 |
| max | 100.000000 | 8.610000 | 9.370000 | 84078.000000 |

For the machine learning section of the project, I decided to use the geek rating, average rating, and number of votes to predict which page a board game with certain characteristics would belong to. I wanted to see how well my model would predict how much visibility a board game on board game geek would get based on these characteristics. I ran five supervised learning programs: LinearRegression , KNeighborsRegressor, KNeighborsClassifier, RandomForestRegressor, and RandomForestClassifier.

To begin, I ran a simple linear OLS, and created scatterplots to see how each dependent variable predicts the page number. I decided to use all three of the characteristics to sort the data since I had a small number of variables to start with. This is likely a problem for two reasons: the scatterplots show that geek rating was likely used to sort the data into pages (and so using just geek rating likely gives the best results) and the three variables display strong multicollinearity.





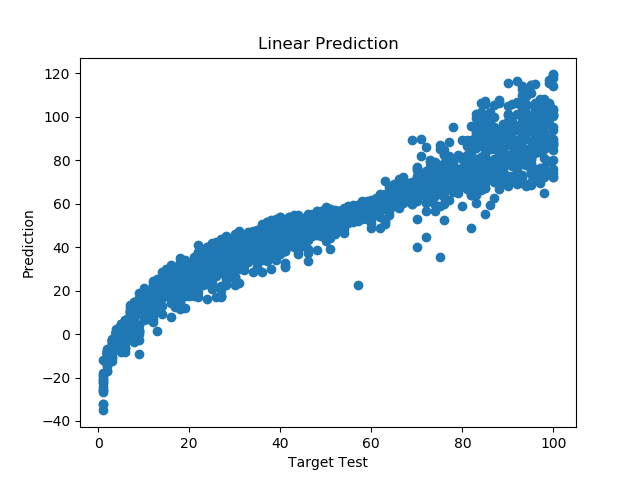


From the scatterplots I noticed that the data displayed a reciprocal function. Therefore, converted all of my explanatory variables by 1/x. I then used train\_test\_split to create to split my data into training and test data. I then compared the predicted data to the true values. For this I used R^2 and got a value of 0.904764910059101. 90.48% of the variation in the true value is predicted by the prediction from the test data.

I then ran k-NN Regression. I first used the GridSearchCV to identify the correct number of neighbors for the k-NN classifier. Using this program, I came up with the optimal number of neighbors as 1. Using the same process as for the linear model. I found a R^2 value of 0.8980265966372528. 89.8% of the variation in the true value is predicted by the prediction from the test data.

Next I ran Random Forest Regression and Random Forest Classification. I used an n\_estimators of 101 which is 1 more than the default level. From the Regression, I got an R^2 value of 0.9999084601134695. 99.99% of the variation in the true value is predicted by the prediction from the test data. Running the feature importance function, I got 'geek\_rating': 0.9999381887140169, 'avg\_rating': 3.177831256546148e-05, 'vote\_number': 3.003297341773011e-05. From the Random Forest Classifier, I got an accuracy score of 0.7555. The model accurately predicted the page number 75.55% of the time. Using the feature importance function, I got, 'geek\_rating': 0.5304790693002981, 'avg\_rating': 0.2129173919607326, 'vote\_number': 0.2566035387389695.

I think the best predictor is the Random Forest Regressor. For one it has the highest R^2 value of any of the models. This does not automatically mean it is the best predictor but it explains a significant amount of the variation. Looking at the scatterplot of the linear model prediction to true value we see that the errors are not random. The model tends to overpredict early and underpredict later:



I think the Random Forest Regressor is better than KNN regressor because of the higher R^2 and the cleaner prediction to target values (See scatter\_knn\_regression\_prediction.png and scatter\_random\_forest\_regression\_prediction.png). I prefer the Random Forest Regressor over the Classifier because of the feature importance. The R^2 and accuracy scores cannot be compared. However, the regressor puts a heavier importance on the geek rating. As mentioned earlier the geek rating is the variable actually used to sort the data so the fact that the regressor finds this more important suggests it is more accurate.

Overall, all of the measurements suggest that the geek rating, average rating, and vote number are very accurate in predicting what page number a specific board game will belong to. This suggests that if you did market research and could predict how board game geek would view your game you could predict its publicity on the website. This would allow you to compare how much visibility your game needs to what is expected to get. This could inform decisions on whether you need to do additional advertising for a board game and give an indication of whether the board game should be produced at all.