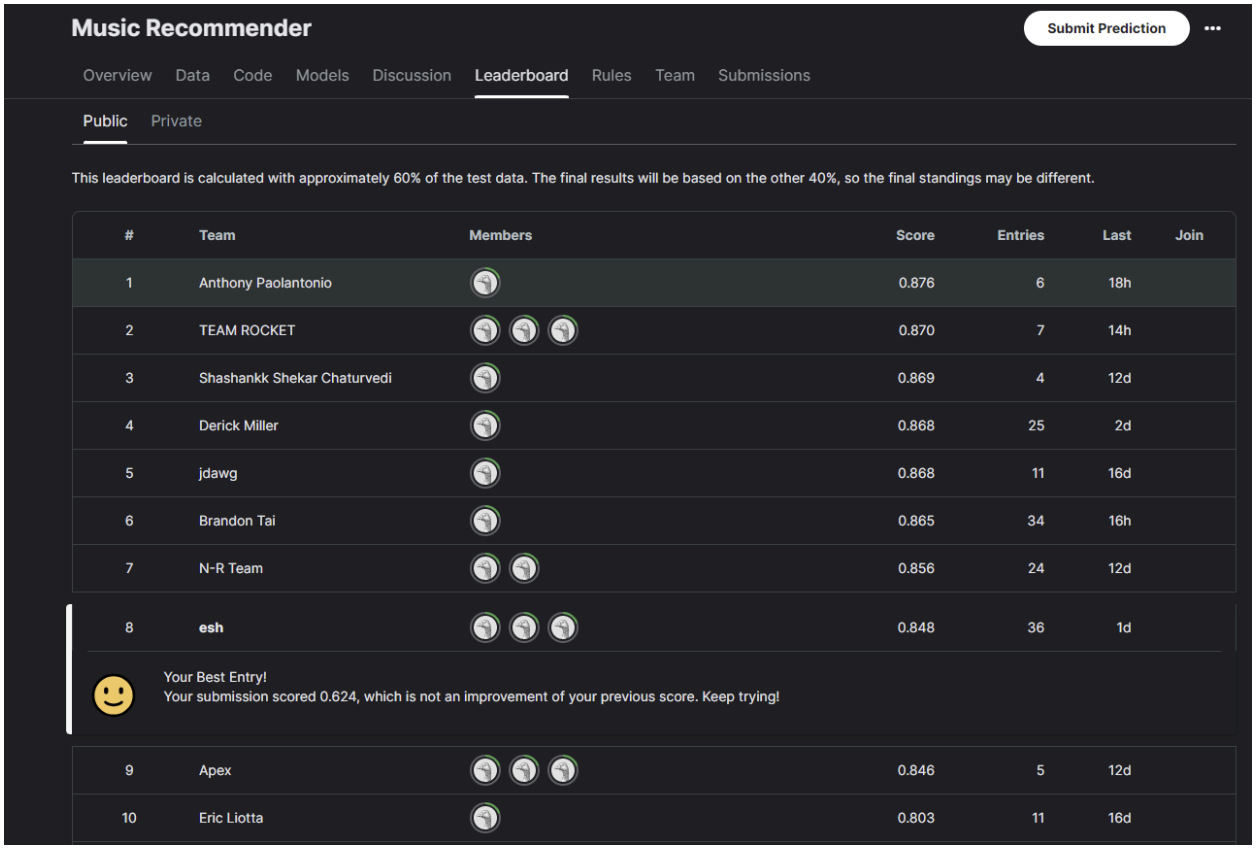
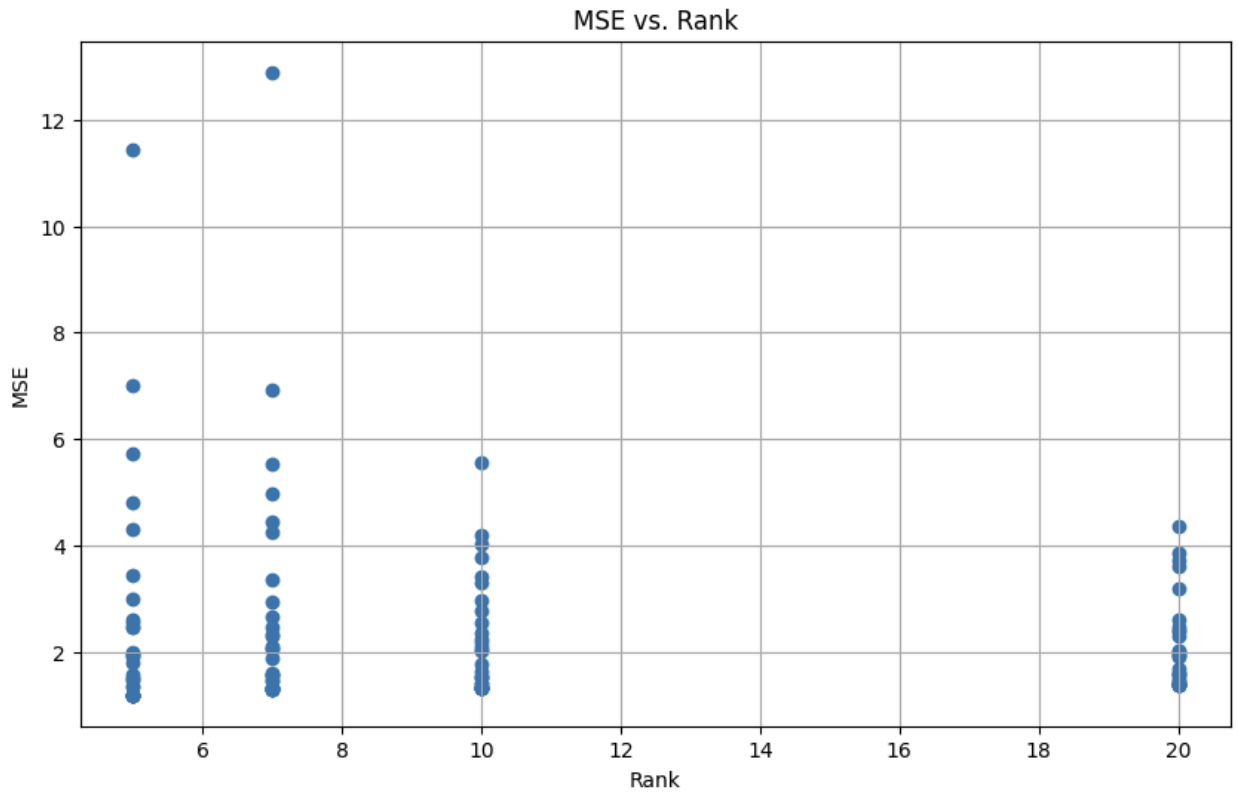
Team Esh: Chris Muro, Andrew Greensweight, Marc DiGeronimo

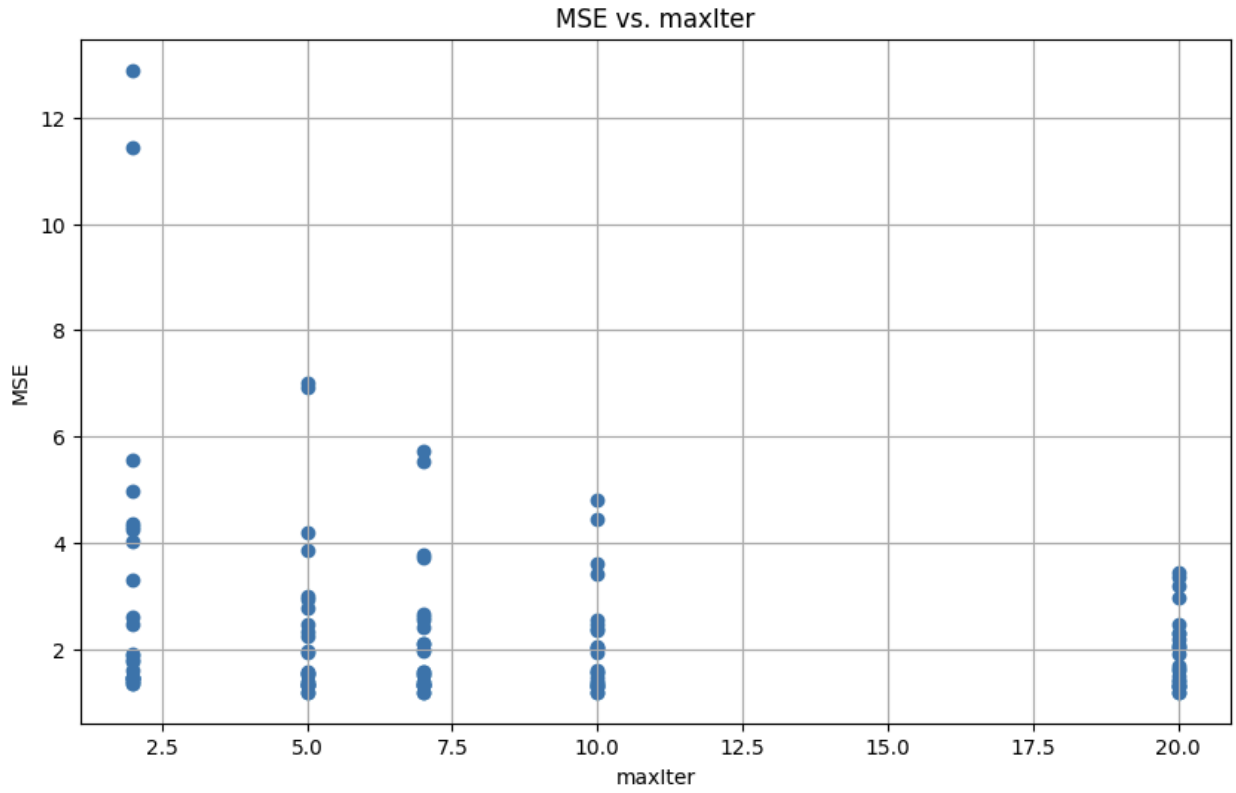


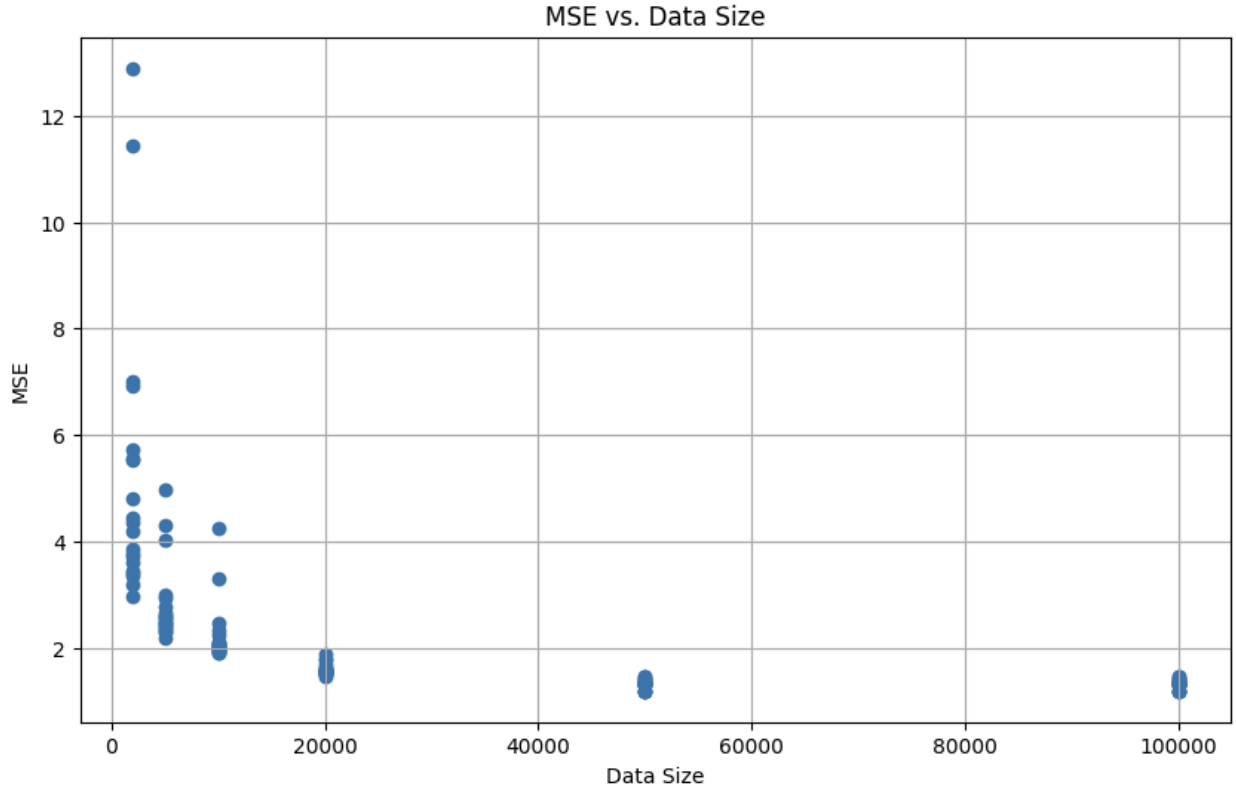
Current Score = .848 Total Submissions = 36

Part 2: For part 2 of the lab, re_u.data file was used. It contains 3 columns of data being the same as the trainItem.data and testItem.data files in the previous part which are in order from left to right, userID, itemID and the rating. First the environment must be configured properly on google colab. This is done by installing the proper programs and seting the enviornment variables and uploading the file. Also re_u.data needs to be split into the train and test set. To do this the dataset was split using np.array_split() function. For 1. maxIter is set to 20 and the rank sizes of 5,7,10 and 20 are tested. With Constant maxIter 20 and rank = 5 the mse was 1.1893121240129518. Rank 7 mse = 1.2858819425771069, Rank = 10, mse = 1.322679774938557, Rank =20 mse = 1.402295299515129. Therefore for these observations we can conclude that at same maxIter and data_size values, as rank increases, mse also increases. This would lead us to using lower rank number for possible models. For 2. rank is set to equal a constant 10 as maxIter takes on the possible values of 2,5,7,10. For maxIter of 2, mse = 1.3911330043443335. MaxIter = 5, mse = 1.315439824961866, MaxIter = 7, mse = 1.3192493847018245. MaxIter = 10, mse = 1.3202044450326698. The results are very interesting. It is assumed that the more iterations the more accurate the model and the less the mse. That is true for this data only to a point however. The lowest mse was at MaxIter 5 and then the mse gradually increased. This may be explained by overfitting as the more iterations

the more closley the model will resemble the traning set and loose precision on the test set. This would lead us to choose a lower MaxIter value such as 5 but not too low as 2. However this does not seem to have a drastic impact overall on the mse. For 3, rank is fixed to constant 10, and maxIter is set to constant 20, Here different sizes of data are taken for the size of the train and test set, the values being 2000, 5000, 10000, 20000, 50000 and 100000. For data_size = 2000, mse = 2.9512606448263443. Data_size = 5000, mse = 2.1864230858059064, Data_size = 10000, mse = 2.0322522978298445, Data_size = 20000, mse = 1.6178100700769171. Data_size = 50000, mse = 1.322679774938557. Data_size = 100000, mse = 1.322679774938557. From the results of our data, we can conclude that for the dataset increase in data size is inversely proportional to mse. This would lead us to choose the highest possible data size for our model. Which is consistent with our prediction as more data means more trends and patterns can be identified and averaged out and outliers become less impactful. From the parameters tested in this HW, we conclude that the data_size will change MSE value most significantly with MaxIter having the least significant impact for our data and models. There are more parameter sets tested with their corresponding mse in the provided .ipynb file for part 2. Highest MSE = 12.89519086944308 occurs with rank = 7 maxIter = 2 and data_size = 2000. Lowest MSE = 1.1884060045633205 occurs with rank = 5, maxIter = 10 and data_size = 100000.




MSE vs. maxIter
MSE
maxIter
2
4
6
8
10
12
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


MSE vs. Data Size
MSE
Data Size
2
4
6
8
10
12
0
20000
40000
60000
80000
100000