Data Analysis Project 2 Report

We imputed the data with the mean. This means we will replace all missing values with the average of the corresponding column.

1.1) To find the most correlated users, we first need to calculate the Covariance Matrix of all users. The Covariance matrix entries are such that $(\Sigma)_{ij}$ is equal to the Covariance of User i with User j. Like all Covariance Matrices our diagonal is the Variance for each User i. We then just need to take the largest value in every column(or row) that is not on the diagonal.

1.2) The Users that are most correlated are User 896 and User 831. We can also see that there almost 301 users who's highest correlated user is 896

1.3) The Users 896 and Users 831 have a Correlation: 0.999542

1.4) The users 0,1,2,…,9 and their correlated users

User 0 and User 583
User 1 and User 831
User 2 and User 896
User 3 and User 364
User 4 and User 896
User 5 and User 99
User 6 and User 239
User 7 and User 896
User 8 and User 896
User 9 and User 1004

We first split the training data using sklearn.model selection module train_test_split

2.1) We model df_pers = function(df_rate) with Linear regression module from sklearn.linear_model. We also used the mean_squared_error module from

sklearn.metrics. Our Training error was .6097 and our Validation error was 3.7158. This was more than 6 times worse than on training. But that is to be expected because we used our training data to create and train our model.

2.2) We model df_pers = function(df_rate) with the Ridge regression module from sklearn.linear_model. We used the same mean_squared_error module from 2.1. Our results were

Train error for Ridges 0.001, 0.6097000210818234
Validation error for Ridges 0.001, 3.71422096385074

Train error for Ridges 0.01, 0.6097015012654998
Validation error for Ridges 0.01, 3.7012911041792154

Train error for Ridges 0.1, 0.6098311181080476
Validation error for Ridges 0.1, 3.583512539784067

Train error for Ridges 1, 0.6153896429448558
Validation error for Ridges 1, 2.943675235320678

We choose the Ridge with the best prediction/lowest MSE, which is 1

2.3) We model df_pers = function(df_rate) with the Lasso regression module from sklearn.linear_model. We used the same mean_squared_error module from 2.1. Our results were

Train error for Lassos 0.001, 0.6344647828183012
Validation error for Lassos 0.001, 2.4121469116113565

Train error for Lassos 0.01, 0.8917202015904482
Validation error for Lassos 0.01, 1.363186335078768

Train error for Lassos 0.1, 1.2004708808332134
Validation error for Lassos 0.1, 1.2493994505896162

Train error for Lassos 1, 1.217214495222191
Validation error for Lassos 1, 1.2593598026555604

We choose the Lasso with the best prediction/lowest MSE which is 0.1