In the movie ratings dataset, we were asked to explore two topics:

1) Correlations between users' movie ratings
2) Finding an optimal linear regression model with the movie ratings as the independent variable and personality questions as the dependent variable

Below I will explain my methods and conclusions for each part:

Topic 1:

First, I imported the already cleaned dataset of movie ratings. However, there were many missing values, which poses a problem because we need a fully filled matrix in order to find the correlation matrix. It makes the most sense to fill the missing values with the corresponding column mean, although this will inflate the correlations between users with many missing values.

Our main goal was to find the most correlated user of each user. To do this, after I obtained the correlation matrix, I found the absolute value of each correlation since the highest correlation refers to the magnitude of the correlation. I then set the diagonals, which were equal to 1 since each user is 100% correlated with itself, equal to 0 to make it easy to find where the maximum correlation occurs for each row other than itself.

(Q1.1) The DataFrame showing each user's most correlated user is posted below:

	0	1	2	3	4	5	6	7	8	9	...	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096
0	118	831	896	19	784	990	1071	1074	821	1004	...	1048	818	352	896	896	896	784	896	896	710

Note: In the csv file, rows 2-1098 correspond to the users. Here, user 0 is row 2, and so on until user 1096 is row 1098.

(Q1.2 and 1.3) The maximum correlation was 0.998789 between users 831 and 896 – however analysis of the dataset shows that the correlation is so high because these two users had almost no data, so their correlation was inflated when the missing values were replaced with the column mean.

(Q1.4) The most correlated user for the first 10 users is also shown above.

Topic 2:

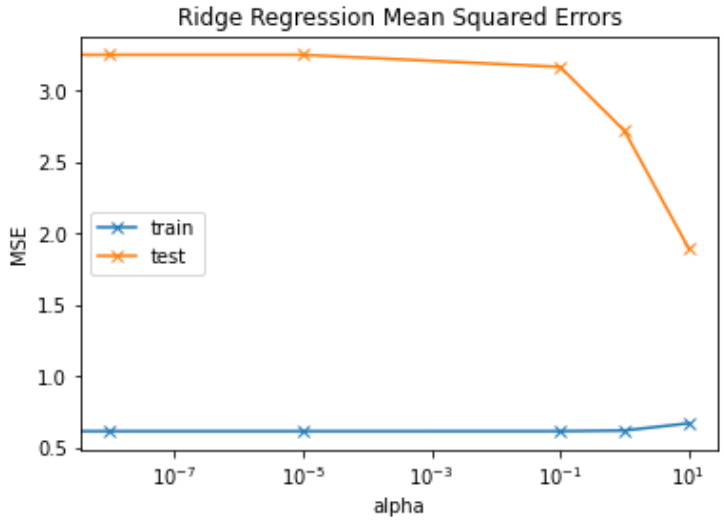
I performed linear regression to model df_pers as a function of df_rate, where df_rate are the users' movie ratings (Columns 1 – 400) and df_pers are the users' personality question ratings (Columns 401 – 474).

(Q2.1) To split the data into training and testing parts in an 80:20 ratio, I used sklearn's train_test_split module (with random_state = 42 for a repeatable outcome) to produce 4 sets of data: X_train, X_test, y_train, and y_test.

I then trained three different models: Multiple Linear Regression, Ridge Regression, and Lasso Regression. For all three models, I calculated the training error as the mean square error between y_train and the X_train prediction (which I call y_pred_train), and I calculated the testing error as the mean square error between y_test and the X_test prediction (which I call y_pred_test).

(Q2.2) For MLR, the MSE on the training part is 0.61276. The MSE on the testing part is 3.25096.

(Q2.3) For Ridge Regression, we were tasked to find errors for 6 different values of the hyperparameter alpha from [0.0, 1e-8, 1e-5, 0.1, 1, 10]. When alpha = 0, the MSE of the training and testing parts are the same as the errors of MLR (Q2.2), since Ridge Regression when alpha = 0 is equivalent to MLR. The training and testing errors for the other alphas on log scale are plotted below:



Since the lowest MSE on the testing part occurs when alpha = 10, the optimal choice of alpha among the ones given is alpha = 10. We are interested in minimizing the test error because it implies that the model has the best bias-variance tradeoff – it is neither underfitting nor overfitting.

(Q2.4) For Lasso Regression, we were tasked to find errors for 4 different values of hyperparameter alpha from [1e-3, 1e-2, 1e-1, 1]. The training and testing errors for the alphas on log scale are plotted below:

Since the lowest MSE on the testing part occurs when alpha = 0.1, that is our optimal choice of alpha among the ones given using the same logic as (Q2.3).

Overall, Lasso Regression with alpha = 0.1 resulted in the lowest MSE on the testing part, implying that among all our models, this one has the best bias/variance tradeoff and should be our model of choice.