Data analysis Project 2

Yoon Tae Park(yp2201@nyu.edu)

Q1:

1.1. For every user in the given data, find its most correlated user.

Answer)

User 0: User 118 with correlation coefficient of 0.5643247218460281,
User 1: User 831with correlation coefficient of 0.831628039345204,
User 2: User 896 with correlation coefficient of 0.9441217594224397,
…
User 1094: User 896 with correlation coefficient of 0.7417160890513692,
User 1095: User 896 with correlation coefficient of 0.8425494495934484,
User 1096: User 710 with correlation coefficient of 0.6112931872103409
(Full lists are presented on the Jupiter notebook: 2021_fall_dsga1001_proj02_yp2201.ipynb)

Explanation)

- Before doing the analysis, I've filled null values with mean value of the corresponding column, as mentioned in the instruction. To find the most correlated user for each user, I've calculated correlation coefficient for each user's list of movie ratings to rest of other user's list of movie ratings, one by one.
- Also, I've sliced the dataset by only movie ratings, since columns other than movie ratings are asking characteristics of users with different value range.
- By comparing every other correlation coefficient, I was able to find the most correlated user for each user. Iterating this logic for every other user, I got the results as above.
- It is interesting to see that there is a same user to have the most correlation coefficient for different users. For example, user 2, 1092, 1094, 1096 have the most correlation coefficient value with user 896, which is the same user.

1.2. What is the pair of the most correlated users in the data?

Answer)

(User 896, User 831) is the pair of the most correlated users in the data

Explanation)

- Looking at user 896 and 831, and user 896 being the most correlated users for several other users, I've searched ratings for user 896 and 831.
- User 896 has no ratings at all previously, so ratings were filled with the mean ratings for each movie.
- User 831 has only two movie ratings, and all other null movie ratings are converted into mean ratings.
- This explains the most correlated pair of users as (User 896 and User 831), since user 896 and user 831 have mostly mean ratings for each movie and those ratings should be same.

1.3. What is the value of this highest correlation?

Answer)

The value of the highest correlation is 0. 9987890924779805

Explanation)

This value is from user pair (User 896, User 831) and since both users have only two different movie ratings, correlation of 0.9987890924779805 makes sense.

1.4. For users 0, 1, 2, \dots, 9, print their most correlated users

Answer)

Most correlated user for 0th user is: 118th user
Most correlated user for 1th user is: 831th user
Most correlated user for 2th user is: 896th user
Most correlated user for 3th user is: 19th user
Most correlated user for 4th user is: 784th user
Most correlated user for 5th user is: 990th user
Most correlated user for 6th user is: 1071th user
Most correlated user for 7th user is: 1074th user
Most correlated user for 8th user is: 821th user
Most correlated user for 9th user is: 1004th user

Explanation)

Printing out first 10 rows of the result in problem 1.1

Data analysis Project 2

Yoon Tae Park(yp2201@nyu.edu)

Q2:

2.1. Model df_pers = function(df_rate) by using the linear regression.
What are the errors on: (i) the training part; (ii) the testing part?

Answer)

Train error: 0.6136136807843432, Test error: 3.7512602611461263

Explanation)

- To get above results, set features as movie ratings(df_rate), and target values as ratings other than movie ratings but not including gender identity, only child, and movies enjoyed alone(df_pers). Split dataset into training and testing as the ratio 0.80: 0.20, and for consist result, set random state = 0
- Create logistic regression function and train by using fit function. Then, calculate train error and test error by using mean squared error.
- Train error comes from comparing actual train result to predicted train result, and Test error comes from comparing actual test result to predicted test result

2.2. Model df_pers = function(df_rate) by using the ridge regression with hyper parameter values alpha from [0.0, 1e-8, 1e-5, 0.1, 1, 10]. For every of the previous values for alpha, what are the errors on: (i) the training part; (ii) the testing part? What is a best choice for alpha?

Answer)

Alpha: 0.0, train error: 0.6136136807843432, test error: 3.7512602611461956
Alpha: 1e-08, train error: 0.6136136807843432, test error: 3.7512602466154603
Alpha: 1e-05, train error: 0.6136136807858574, test error: 3.751245730543043
Alpha: 0.1, train error: 0.6137444088945111, test error: 3.6189809955290633
Alpha: 1, train error: 0.6192994848065657, test error: 2.9761184218057446
Alpha: 10, train error: 0.6721924040742049, test error: 1.9460619232622307
Best choice for alpha is 10, since it will decrease the test error at most (test error: 1.9460619232622307)
In terms of minimizing train error, alpha 0.0 is the best choice (train error: 0.6136136807843432)
In terms of minimizing test error, alpha 10 is the best choice (test error is 1.9460619232622307)

Explanation)

- To get results, create Ridge regression function and use different alphas (0.0, 1e-8, 1e-5, 0.1, 1, 10) as a hyperparameter. Then, train by using fit function and calculate train error and test error by using mean squared error.
- Minimizing the test error should be the objective of the modeling since we are trying to predict the test result from test features which are not trained by the model. Minimizing the train error is also important but may result in overfitting since it will reflect all the noises and outliers that train dataset might have.
- Ridge regression makes the model in a way of minimizing test error but increasing train error. It gives a penalty for overfitting by shrinking all regression coefficients towards zero. This makes the model more generalized(regularized) compared to the linear regression. Ridge moves in a way of decreasing variance and increasing bias.

2.3. Model df_pers = function(df_rate) by using the lasso regression with hyperparameter values alpha from [1e-3, 1e-2, 1e-1, 1]. For every of the previous values for alpha, what are the errors on: (i) the training part; (ii) the testing part? What is a best choice for alpha?

Answer)

Alpha: 0.001, train error: 0.6383958246258742, test error: 2.4416434629270967
Alpha: 0.01, train error: 0.8972701600069916, test error: 1.371722792459132
Alpha: 0.1, train error: 1.2085190626256925, test error: 1.2573981863303583
Alpha: 1, train error: 1.2254655564827854, test error: 1.2678753797127884
Best choice for alpha: 0.1, since it will decrease the test error at most (test error: 1.2573981863303583)
In terms of minimizing train error, alpha 0.001 is the best choice (train error is 0.6383958246258742)
In terms of minimizing test error, alpha 0.1 is the best choice (test error is 1.2573981863303583)

Explanation)

- To get results, create Lasso regression function and use different alphas (1e-3, 1e-2, 1e-1, 1) as hyperparameter. Then, train by using fit function and calculate train error and test error by using mean squared error.
- Minimizing the test error should be the objective of the modeling since we are trying to predict the test result from test features which are not trained by the model. Minimizing the train error is also important but may result in overfitting since it will reflect all the noises and outliers that train dataset might have.
- Lasso regression makes the model in a way of minimizing test error but increasing train error. It gives a penalty for overfitting by shrinking all regression coefficients to zero (actual zero). This generalizes(regularizes) the model more drastically, compared to ridge regression. Lasso also moves in a way of decreasing variance and increasing bias.