Name : Kewal Jani
NetId : kj2062

Q1:

Note: For all missing values in the data, use the average of the corresponding column so to fill in the missing data.

In this problem, under the most correlated, we consider the largest correlation in the absolute value.

To fill the missing data First I took the average of the entire column and substituted the nan value with average and I performed the same for all the 477 columns. I was a bit confused about the columns after 400. Whether it is a good idea to substitute average but later I thought that average would actually be a good option as it will help to get better corelated results. After filling the NAN values with average the questions were to compare the users. For that I took the data and made a new matrix of 1097 X 477. This suggest that there are 1097 users. I found correlation between all the users

By applying the r= np.corrcoef(a.T,rowvar=True) on the dataset.

Then I took the absolute value as there will be some -ve values for correlation matrix.

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

Ans : After getting the corelation matrix it was easy to find the most corelated user among all the users. For each column of matrix, I searched for 2nd highest value which states the pair which is most correlated to the user because the 1st value would be for the correlation with itself which is 1.

Conclusion: We obtained pair of correlated users for each from the corelated matrix. The decision for rating which r elates for every individaul.

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

After getting the corelation matrix it was easy to find the most corelated user among all the users. I iterated over the entire matrix and found the maximum value of corelation among all the users the value was 0.9994513305109192
And the index was [831, 896]

Which Indicates that there is maximum corelation between user 831 and 896 when index starts with zero.

The user 831 and 896 are most correlated that is they will make similar decisions regarding the ratings of movie.

1.3. What is the value of this highest correlation?

Ans :

In this question we have to find the pair which is most corelated. That is the value of the corelation which is maximum. The method to obtain the most correlated pair was the same which I performed in second question. From that I was able to get the most correlated pair [831,896]. **The user 831 and 896 are most correlated that is they will make similar decisions regarding the ratings of movie.**

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

Ans :

I ran the loop for 10 users that is for 1-10 columns and calculated the 2ndhighest value of the matrix which gives the most correlated par of those users.

The results I got were
for user 0 the correlated data of user is 583
for user 1 the correlated data of user is 831
for user 2 the correlated data of user is 896

Name : Kewal Jani
NetId : kj2062

for user 3 the correlated data of user is 896
for user 4 the correlated data of user is 896
for user 5 the correlated data of user is 99
for user 6 the correlated data of user is 239
for user 7 the correlated data of user is 850
for user 8 the correlated data of user is 896
for user 9 the correlated data of user is 1004

These indicated the user that resembles the most with the other user. The decision making of the user are related stated in above pair.

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?

Ans :

For this question we need to find the values of columns from 400 to 474 from the ratings of the users (the columns from 0-400) for that we need to train the model and then test it with the help of linear regression. I split the data using the library train_test_split and after that I trained the model using the linear_model library.

I applied the test data and found the mean_squared_error and scores of the error for the test and train were following:

Test error: 3.3076880927711865
Train error: 0.6150867921831857

This indicates that the test error is more compare to the train error. That is, it is comparatively less accurate and the model is little overfitted.

2.2. Model df_pers = function(df_rate) by using the ridge regression with hyperparamter 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?

Ans

In this question we were using the same splitted data and from that data we were performing the ridge regression to predict the values. For this method we are using different values of the alpha. And getting error values for each testing and testing we get as follow.

[1e-8,1e-5,0,0.1, 1,10]
Test error: 3.307688081259402 Train error: 0.6150867921831857
Test error: 3.3076765810775535 Train error: 0.6150867921846376
Test error: 3.3076880927711914 Train error: 0.6150867921831856
Test error: 3.2028584784624163 Train error: 0.6152128006695894
Test error: 2.6909825915207817 Train error: 0.6206515636479406
Test error: 1.808604743686749 Train error: 0.6747823869719883

From the results we can see that the optimum value **for alpha is 10** and it goes decreasing as we iterate over the values of the alpha.

The value of the alpha at 0 is same as the one we obtained in linear regression which states that there won't be any penalty term in the ridge regression.

Name : Kewal Jani
NetId : kj2062

2.3. Model df_pers = function(df_rate) by using the lasso regression with hyperparamter 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?

Ans :

performing the same experiment but with lasso regression it comparatively took more time to iterate over small values. I took the same train and test split and then I applied the values of alpha given in the question. The error results for the value of alpha were following.

[1e-3, 1e-2, 1e-1, 1]
Test error: 2.247141334722522 Train error: 0.6391702839805774
Test error: 1.3044716836968897 Train error: 0.9037309593017302
Test error: 1.2246191465005256 Train error: 1.2166821052511474
Test error: 1.2330458071231467 Train error: 1.2342340059607373

The best value from the above result is for the alpha **value 1e-1**.