دانشگاه شیراز

Shiraz University

Fuzzy Logic Homework2

Covid Detection

Instructed by Dr.Tahayori

Reza Tahmasebi

saeed77t@gmail.com, Stu ID: 40160957

Table of Contents

1 Covid detection based on relation matrix

This project aims to create a relationship matrix based on our dataset. And to predict our test data labels with the composition of the relation matrix and test data.

2 Data preprocess

First, the column *SESSO* has been changed; the Female became 0, and the male became 1.

The second problem was the missing values that existed in the data set. I used the *KNNImputer* method from *the sklearn* library to fill the blank values with the four nearest neighbors.

After that, I split the data to test data and train data (80% for training and 20% for testing). I used the *TestAndTrain* method from *PreprocessData* class to break my data.

3 Method to create relation matrix

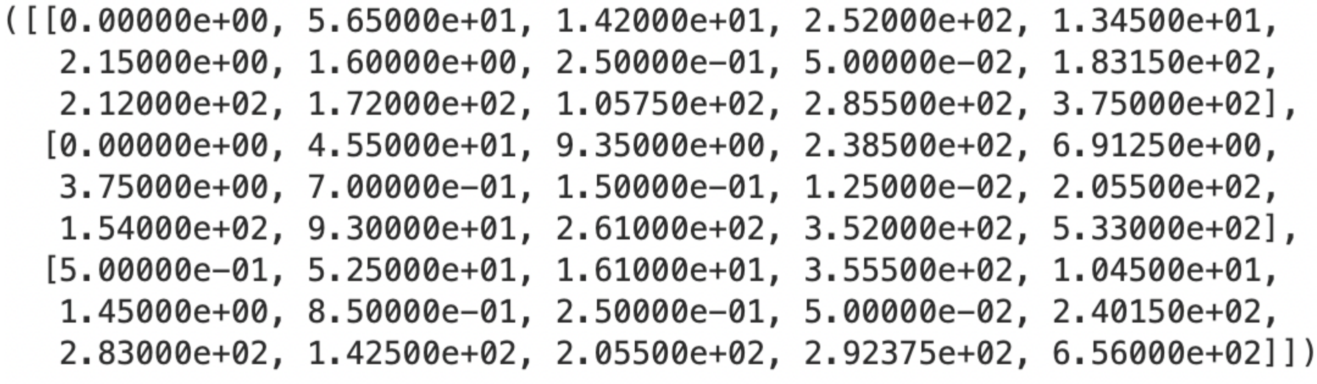
I implemented five methods to create my relation matrix, and each way gives different accuracies.

3.1 Method one numeric range

In this method, I split the training data into three parts based on the labels; training data became three parts because we had three labels.

After that, I determined a numeric range for each class of three parts. We had 15 features, and because of that, each matrix shape became 15x1.

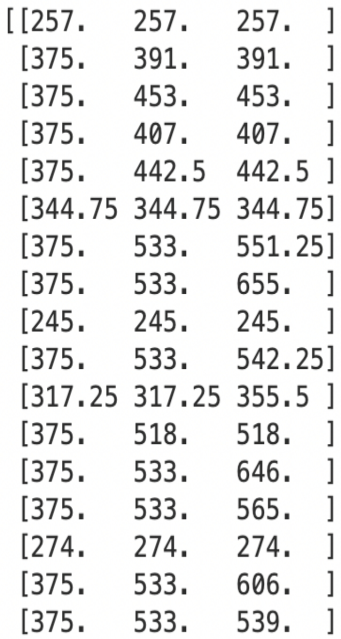
The relation matrix was created with those three matrixes; the relation matrix shape was 3x15.



```
([[0.00000e+00, 5.65000e+01, 1.42000e+01, 2.52000e+02, 1.34500e+01,
   2.15000e+00, 1.60000e+00, 2.50000e-01, 5.00000e-02, 1.83150e+02,
   2.12000e+02, 1.72000e+02, 1.05750e+02, 2.85500e+02, 3.75000e+02],
  [0.00000e+00, 4.55000e+01, 9.35000e+00, 2.38500e+02, 6.91250e+00,
   3.75000e+00, 7.00000e-01, 1.50000e-01, 1.25000e-02, 2.05500e+02,
   1.54000e+02, 9.30000e+01, 2.61000e+02, 3.52000e+02, 5.33000e+02],
  [5.00000e-01, 5.25000e+01, 1.61000e+01, 3.55500e+02, 1.04500e+01,
   1.45000e+00, 8.50000e-01, 2.50000e-01, 5.00000e-02, 2.40150e+02,
   2.83000e+02, 1.42500e+02, 2.05500e+02, 2.92375e+02, 6.56000e+02]])
```

the relation matrix of method one

I used the *composition* method to decompose the relation matrix with the test dataset to create the Similarity matrix.

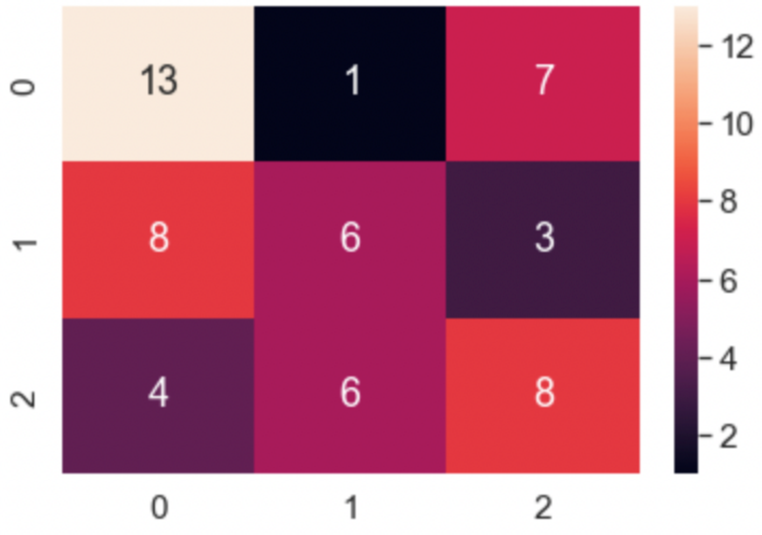


```
[[257.   257.   257.  ]
 [375.   391.   391.  ]
 [375.   453.   453.  ]
 [375.   407.   407.  ]
 [375.   442.5  442.5 ]
 [344.75 344.75 344.75]
 [375.   533.   551.25]
 [375.   533.   655.  ]
 [245.   245.   245.  ]
 [375.   533.   542.25]
 [317.25 317.25 355.5 ]
 [375.   518.   518.  ]
 [375.   533.   646.  ]
 [375.   533.   565.  ]
 [274.   274.   274.  ]
 [375.   533.   606.  ]
 [375.   533.   539.  ]
```

Example of method one Similarity matrix

After all those actions above, I ran the *argmax* method on each row of the similarity matrix to predict the labels.

The average accuracy of this method for the three classes was about 48.2%.

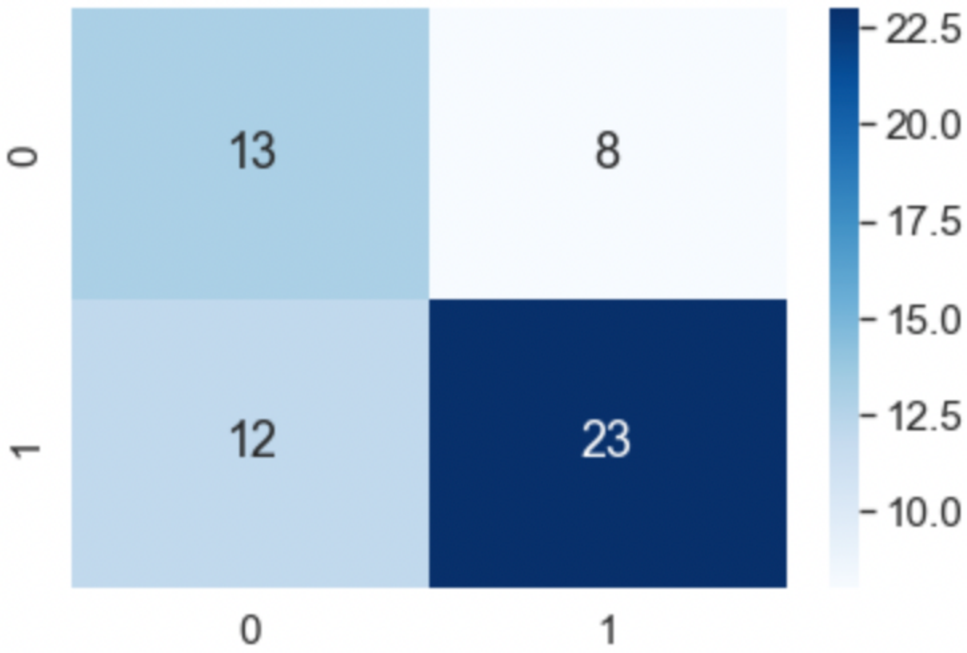


The confusion matrix of method 1 for three classes

Also, I took classes 1 and 2 as one, so class 0 is not sick, and class 1 is sick. And then, I applied method one to it. The results are as follows.

Average accuracy: 64.2%

And the confusion matrix is :



The confusion matrix of method 1 for two classes

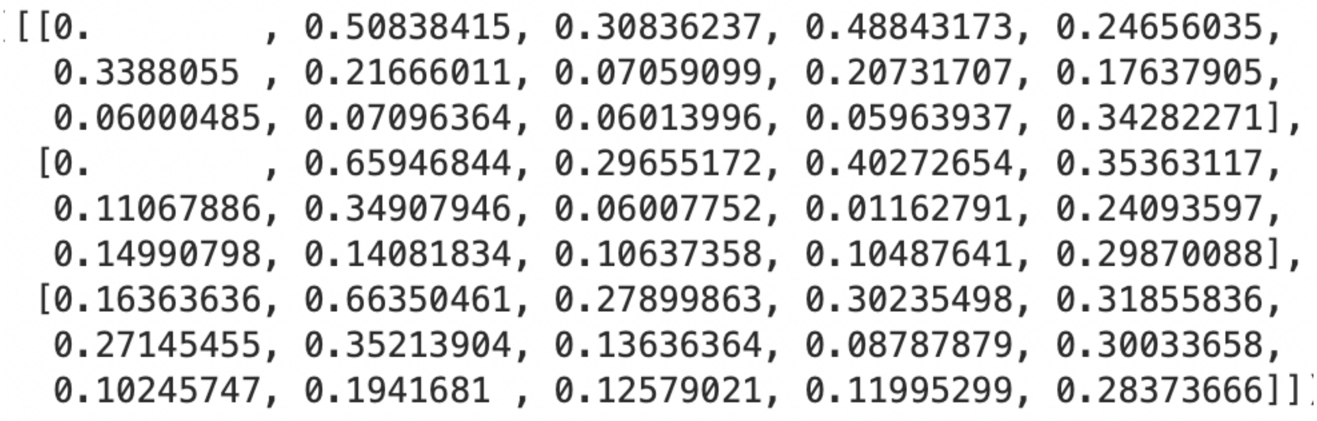
3.2 Method two the average of each feature

In this method, I split the training data into three parts based on the labels; training data became three parts because we had three labels.

After that, I calculate the average of each part for all features. We had 15 features, and because of that, each matrix shape became 15x1.

The data has been normalized in this method.

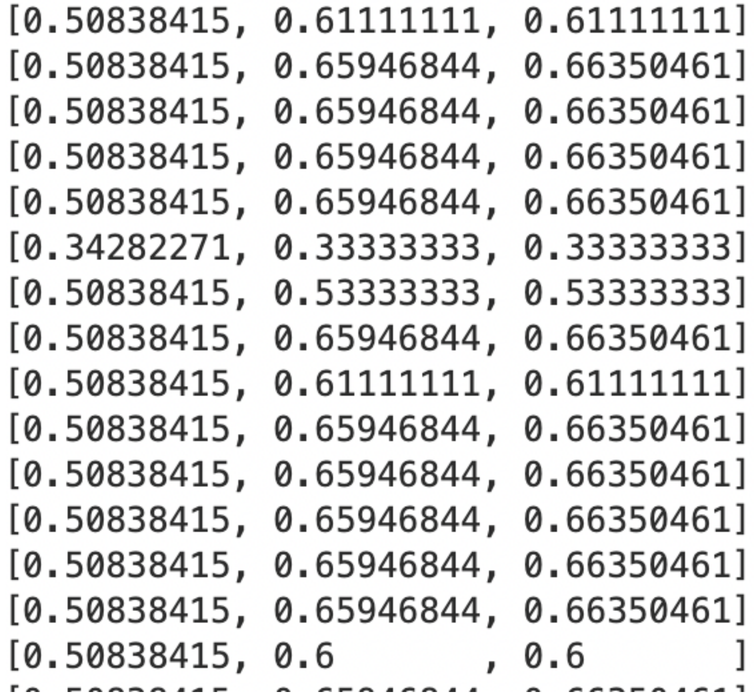
I put the matrixes together to create the relation matrix. The shape of the relation matrix is 3x15.



```
[[0.        , 0.50838415, 0.30836237, 0.48843173, 0.24656035,
  0.3388055 , 0.21666011, 0.07059099, 0.20731707, 0.17637905,
  0.06000485, 0.07096364, 0.06013996, 0.05963937, 0.34282271],
 [0.        , 0.65946844, 0.29655172, 0.40272654, 0.35363117,
  0.11067886, 0.34907946, 0.06007752, 0.01162791, 0.24093597,
  0.14990798, 0.14081834, 0.10637358, 0.10487641, 0.29870088],
 [0.16363636, 0.66350461, 0.27899863, 0.30235498, 0.31855836,
  0.27145455, 0.35213904, 0.13636364, 0.08787879, 0.30033658,
  0.10245747, 0.1941681 , 0.12579021, 0.11995299, 0.28373666]]
```

the relation matrix of method two

After creating the relation matrix, I used the *composition* method to decompose the relation matrix with the test dataset to create the Similarity matrix.

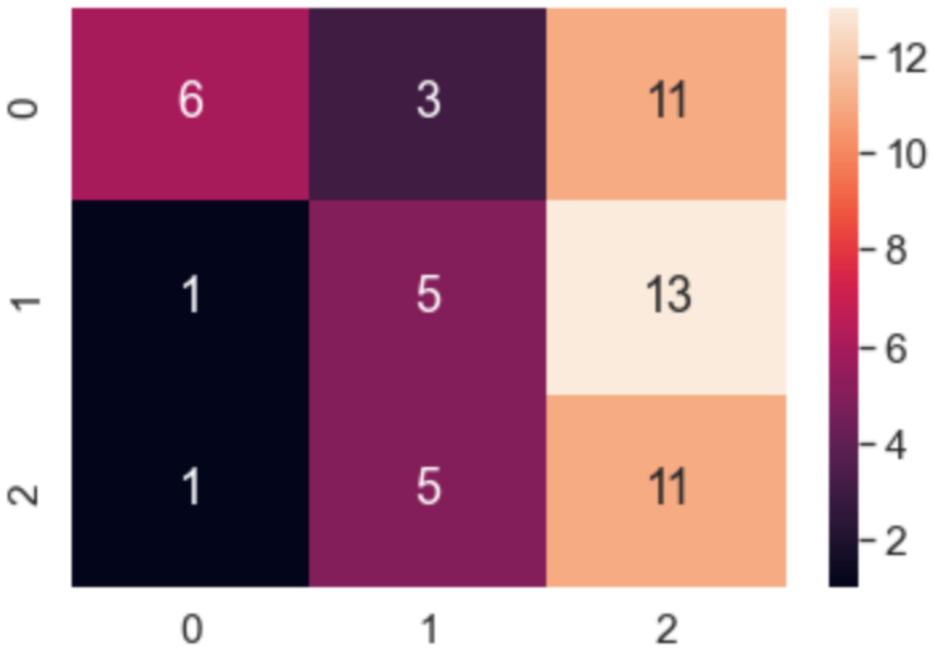


```
[0.50838415, 0.61111111, 0.61111111]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.34282271, 0.33333333, 0.33333333]
[0.50838415, 0.53333333, 0.53333333]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.61111111, 0.61111111]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.65946844, 0.66350461]
[0.50838415, 0.6       , 0.6       ]
```

Example of method two Similarity matrix

Like method one, the *argmax* method ran on a similarity matrix to predict the test data.

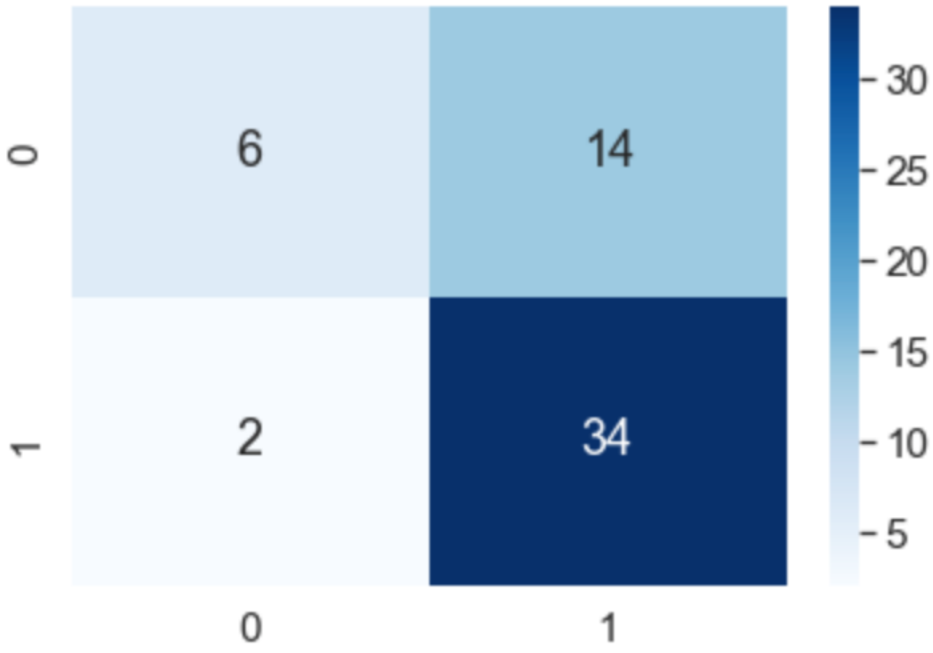
The accuracy of this model for three classes is 39.3%.



	0	1	2
0	6	3	11
1	1	5	13
2	1	5	11

The confusion matrix of method two for three classes

I changed the three classes problem to two class problem and ran method two for it; the accuracy of method two for two classes problem is 71.42%, and the confusion matrix is :



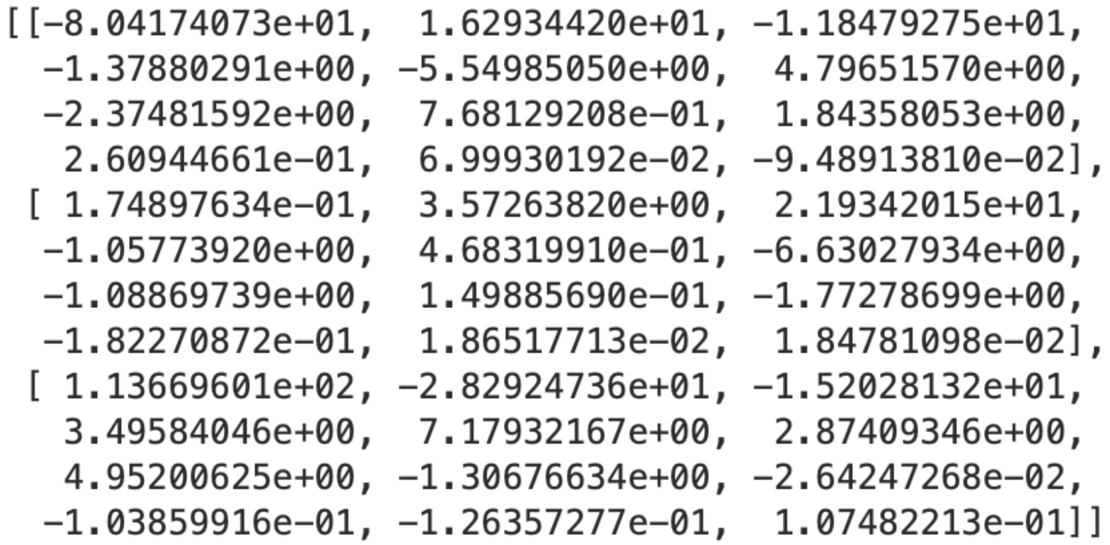
	0	1
0	6	14
1	2	34

The confusion matrix of method two for two classes

3.3 Method three PCA feature extraction

In this method, I used the *PCA* method on the dataset, and then I split the data into three parts based on the class labels.

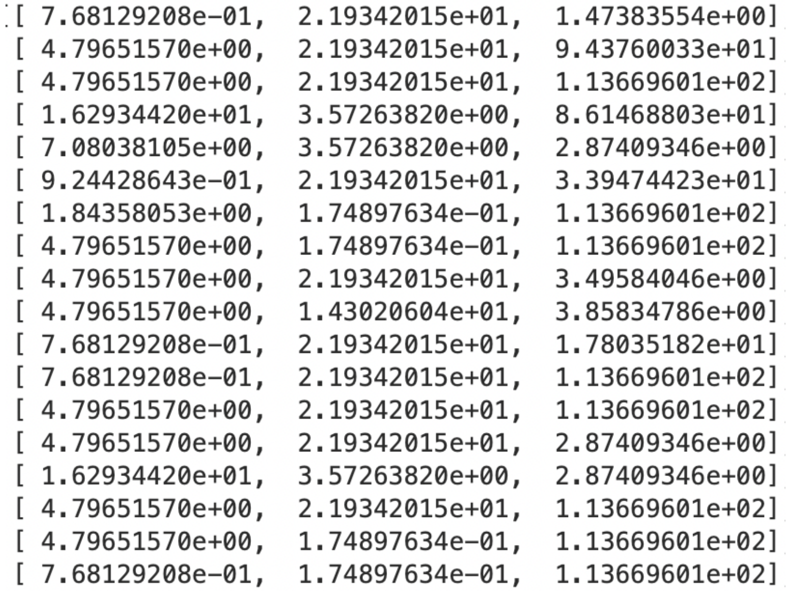
After that, I calculate the average of those three parts for each feature and put them beside each other to create the Relation matrix.



```
[[-8.04174073e+01,  1.62934420e+01, -1.18479275e+01,
  -1.37880291e+00, -5.54985050e+00,  4.79651570e+00,
  -2.37481592e+00,  7.68129208e-01,  1.84358053e+00,
   2.60944661e-01,  6.99930192e-02, -9.48913810e-02],
 [ 1.74897634e-01,  3.57263820e+00,  2.19342015e+01,
  -1.05773920e+00,  4.68319910e-01, -6.63027934e+00,
  -1.08869739e+00,  1.49885690e-01, -1.77278699e+00,
  -1.82270872e-01,  1.86517713e-02,  1.84781098e-02],
 [ 1.13669601e+02, -2.82924736e+01, -1.52028132e+01,
   3.49584046e+00,  7.17932167e+00,  2.87409346e+00,
   4.95200625e+00, -1.30676634e+00, -2.64247268e-02,
  -1.03859916e-01, -1.26357277e-01,  1.07482213e-01]]
```

The relation matrix of method three

After creating the relation matrix, I used the *composition* method to decompose the relation matrix with the test dataset to create the Similarity matrix.



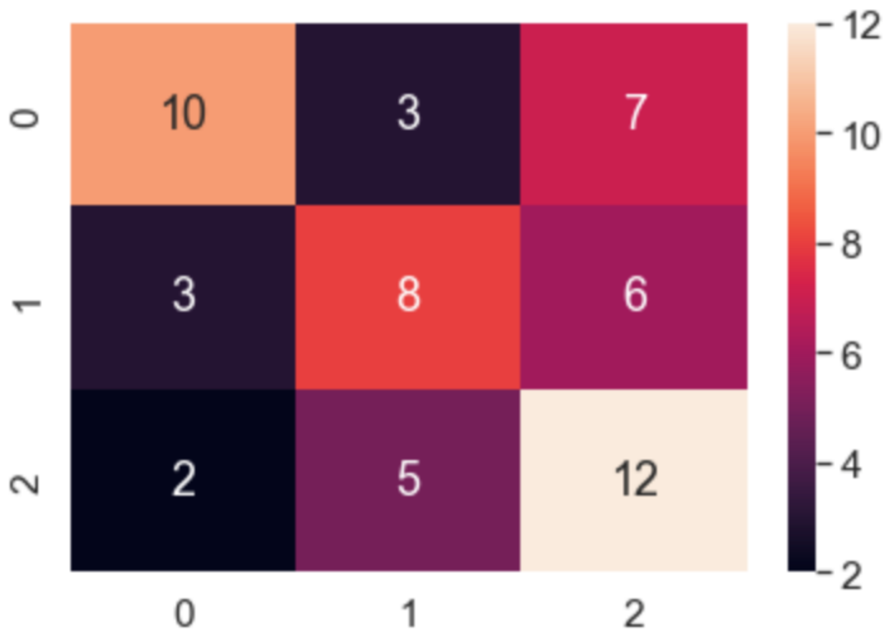
```
[ 7.68129208e-01,  2.19342015e+01,  1.47383554e+00]
[ 4.79651570e+00,  2.19342015e+01,  9.43760033e+01]
[ 4.79651570e+00,  2.19342015e+01,  1.13669601e+02]
[ 1.62934420e+01,  3.57263820e+00,  8.61468803e+01]
[ 7.08038105e+00,  3.57263820e+00,  2.87409346e+00]
[ 9.24428643e-01,  2.19342015e+01,  3.39474423e+01]
[ 1.84358053e+00,  1.74897634e-01,  1.13669601e+02]
[ 4.79651570e+00,  1.74897634e-01,  1.13669601e+02]
[ 4.79651570e+00,  2.19342015e+01,  3.49584046e+00]
[ 4.79651570e+00,  1.43020604e+01,  3.85834786e+00]
[ 7.68129208e-01,  2.19342015e+01,  1.78035182e+01]
[ 7.68129208e-01,  2.19342015e+01,  1.13669601e+02]
[ 4.79651570e+00,  2.19342015e+01,  1.13669601e+02]
[ 4.79651570e+00,  2.19342015e+01,  2.87409346e+00]
[ 1.62934420e+01,  3.57263820e+00,  2.87409346e+00]
[ 4.79651570e+00,  2.19342015e+01,  1.13669601e+02]
[ 4.79651570e+00,  1.74897634e-01,  1.13669601e+02]
[ 7.68129208e-01,  1.74897634e-01,  1.13669601e+02]
```

Similarity matrix of method three

And I used the argmax method to choose in what class our data belongs.

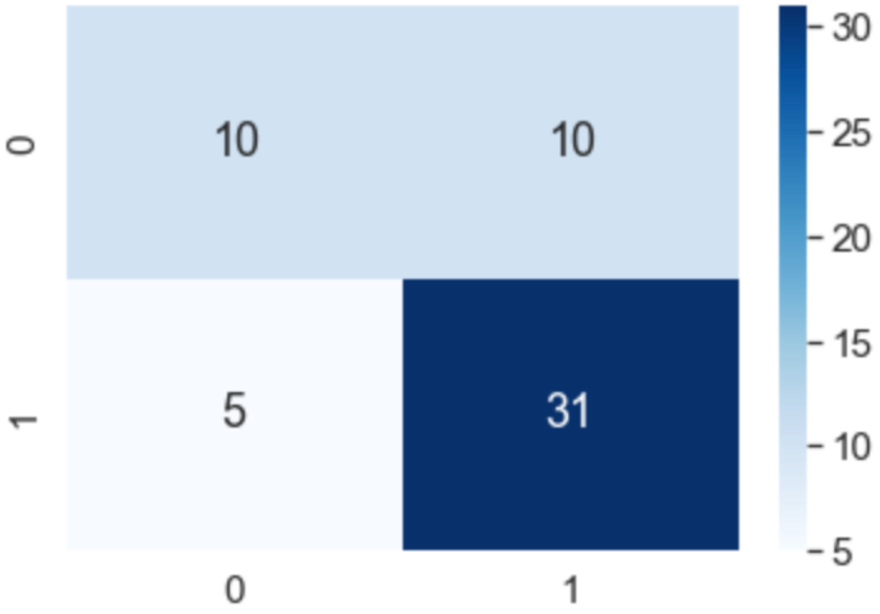
The accuracy of this method for a three-class problem is 53.5%.

The confusion matrix is :



The confusion matrix of three class problems for method three

This method has been running for two-class problems, and the accuracy of this method is:73.21%

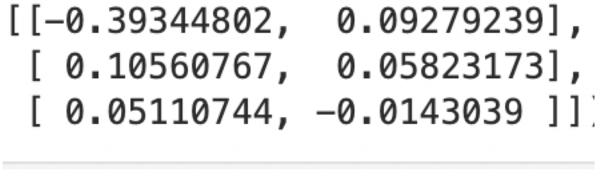


The confusion matrix of two class problems for method three

Note: PCA is a technique for unsupervised data, so PCA leads to overfitting.!

3.4 Method four LDA feature selection

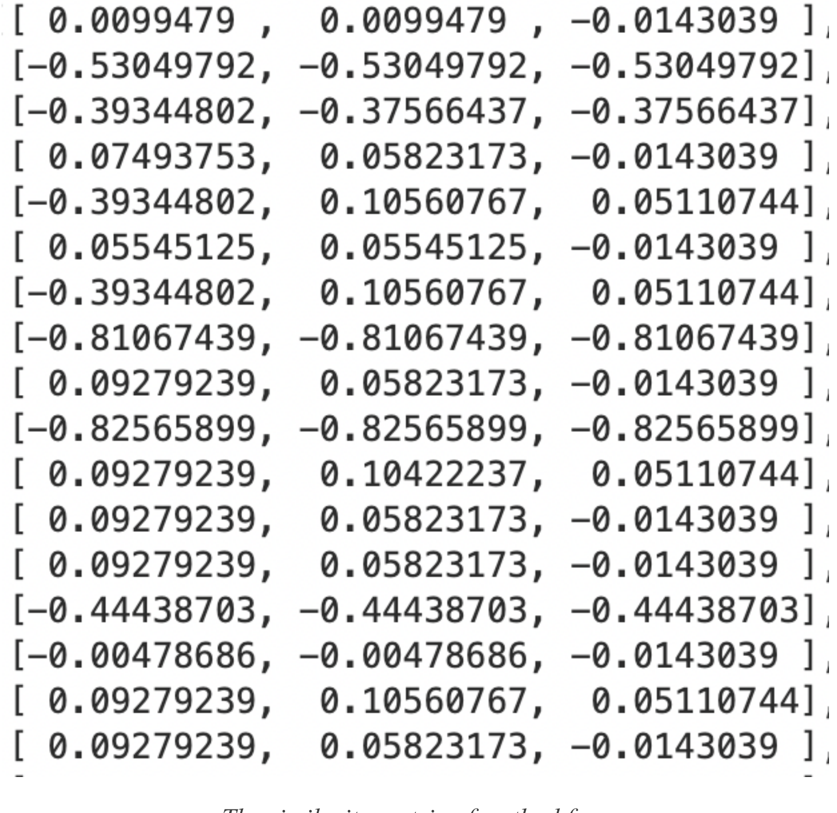
In this method, I used LDA to reduce the number of dimensions, and then like the other methods, I calculated the relation matrix with the average of each feature.



```
[[-0.39344802,  0.09279239],
 [ 0.10560767,  0.05823173],
 [ 0.05110744, -0.0143039 ]]
```

The relation matrix of method four

I used the *composition* method to decompose the relation matrix with the test dataset to create the Similarity matrix.



```
[ 0.0099479 ,  0.0099479 , -0.0143039 ],
[-0.53049792, -0.53049792, -0.53049792],
[-0.39344802, -0.37566437, -0.37566437],
[ 0.07493753,  0.05823173, -0.0143039 ],
[-0.39344802,  0.10560767,  0.05110744],
[ 0.05545125,  0.05545125, -0.0143039 ],
[-0.39344802,  0.10560767,  0.05110744],
[-0.81067439, -0.81067439, -0.81067439],
[ 0.09279239,  0.05823173, -0.0143039 ],
[-0.82565899, -0.82565899, -0.82565899],
[ 0.09279239,  0.10422237,  0.05110744],
[ 0.09279239,  0.05823173, -0.0143039 ],
[ 0.09279239,  0.05823173, -0.0143039 ],
[-0.44438703, -0.44438703, -0.44438703],
[-0.00478686, -0.00478686, -0.0143039 ],
[ 0.09279239,  0.10560767,  0.05110744],
[ 0.09279239,  0.05823173, -0.0143039 ],
```

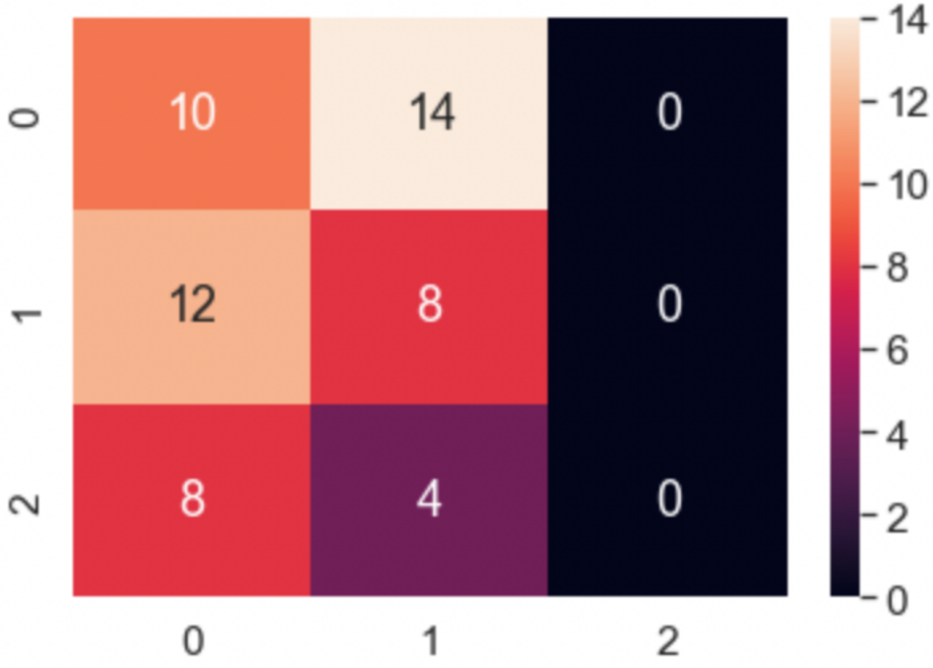
The similarity matrix of method four

And we predict the test data labels with a similarity matrix.

The results are as follows:

Accuracy: 32.1%

The confusion matrix :



The confusion matrix of method four, three classes

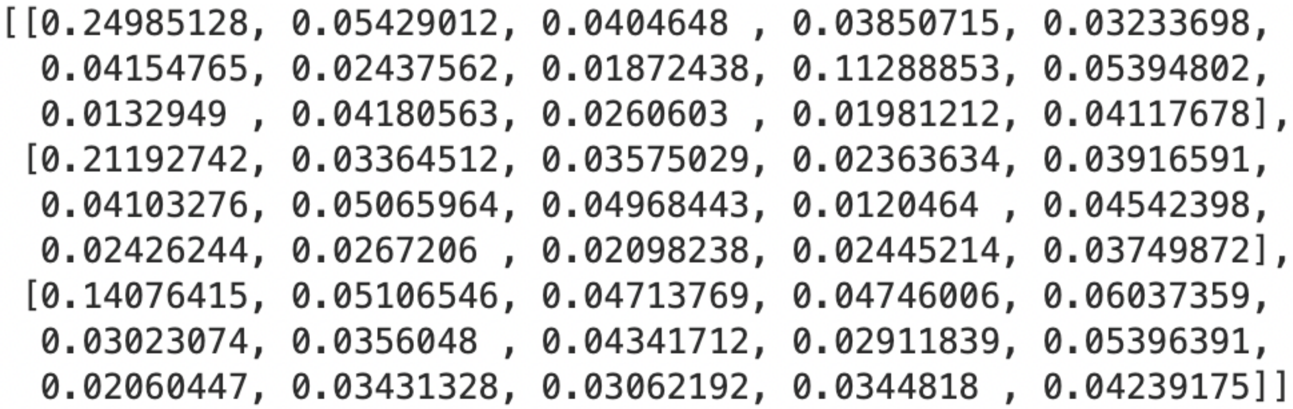
3.5 Method five, the variance of each feature

In this method, I split the training data into three parts based on the labels; training data became three parts because we had three labels.

After that, I calculate the variance of each part for all features. We had 15 features, and because of that, each matrix shape became 15x1.

The data has been normalized in this method.

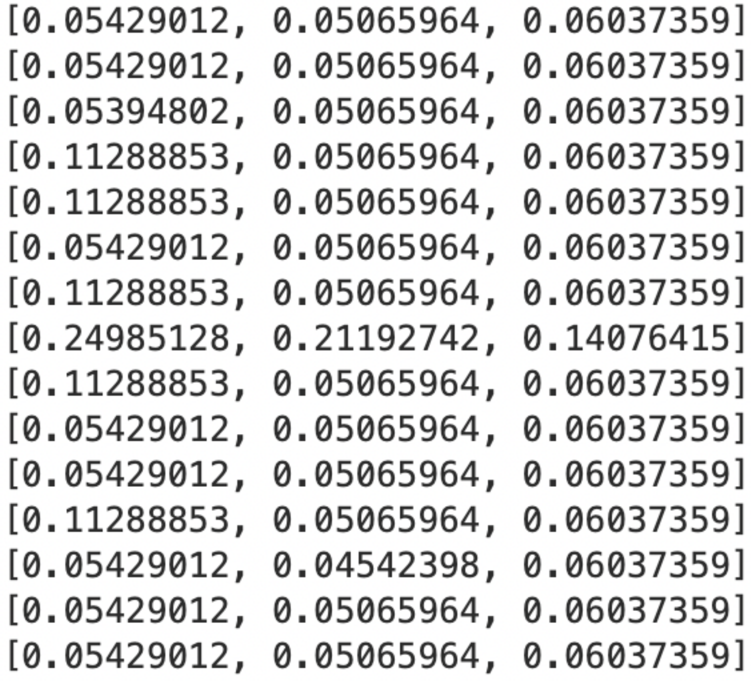
I put the matrixes together to create the relation matrix. The shape of the relation matrix is 3x15.



```
[[0.24985128, 0.05429012, 0.0404648 , 0.03850715, 0.03233698,
  0.04154765, 0.02437562, 0.01872438, 0.11288853, 0.05394802,
  0.0132949 , 0.04180563, 0.0260603 , 0.01981212, 0.04117678],
 [0.21192742, 0.03364512, 0.03575029, 0.02363634, 0.03916591,
  0.04103276, 0.05065964, 0.04968443, 0.0120464 , 0.04542398,
  0.02426244, 0.0267206 , 0.02098238, 0.02445214, 0.03749872],
 [0.14076415, 0.05106546, 0.04713769, 0.04746006, 0.06037359,
  0.03023074, 0.0356048 , 0.04341712, 0.02911839, 0.05396391,
  0.02060447, 0.03431328, 0.03062192, 0.0344818 , 0.04239175]]
```

The Relation matrix of method five

Like the other methods, I used the *composition* method to decompose the relation matrix with the test dataset to create the Similarity matrix.



```
[0.05429012, 0.05065964, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
[0.05394802, 0.05065964, 0.06037359]
[0.11288853, 0.05065964, 0.06037359]
[0.11288853, 0.05065964, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
[0.11288853, 0.05065964, 0.06037359]
[0.24985128, 0.21192742, 0.14076415]
[0.11288853, 0.05065964, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
[0.11288853, 0.05065964, 0.06037359]
[0.05429012, 0.04542398, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
[0.05429012, 0.05065964, 0.06037359]
```

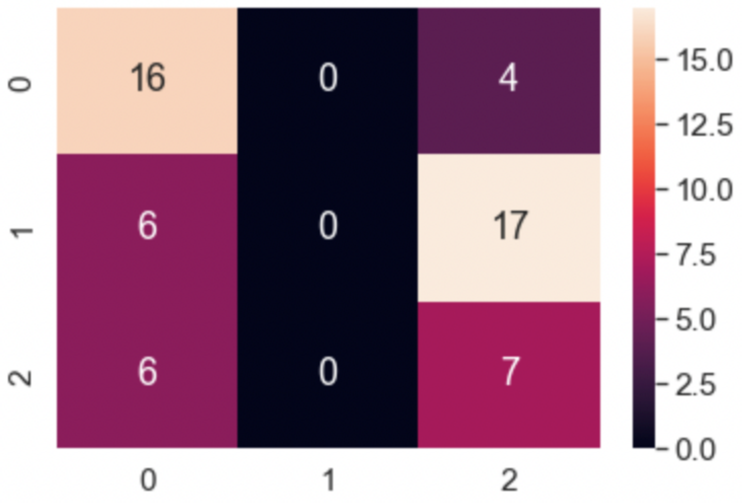
An example of a similarity matrix in method five

After that, the argmax method became handy, and like the other methods, I used it to determine the classes of the test data set.

The results for the three class method are as follow,

Accuracy: 41.07 5

The confusion matrix :

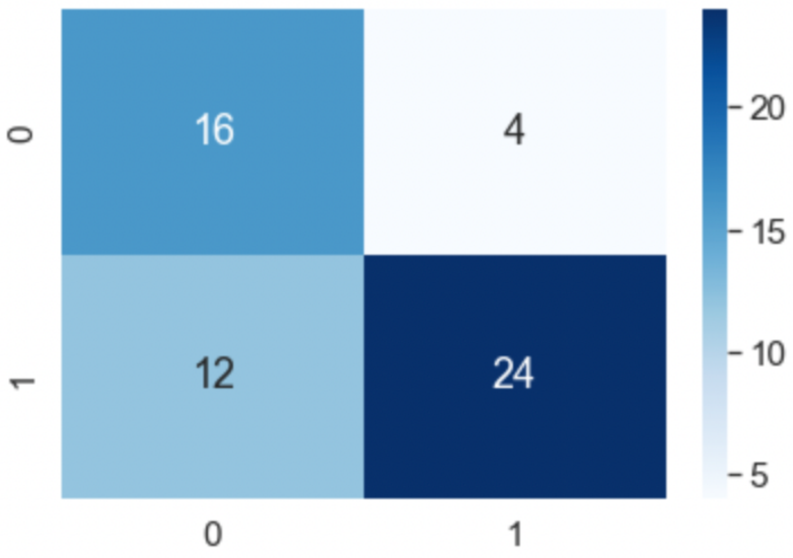


The confusion matrix of three class problems with method five

This method was also applied to two class problem, and the results are as follow :

Accuracy: 71.42%

The confusion matrix is :



The confusion matrix of two class problems with method five