# CS 584-04: Machine Learning

AISHWARYA ANANTHARAM CWID: A20396732

Spring 2019 Assignment 1

## Question 1 (40 points)

Write a Python program to calculate the density estimator of a histogram. Use the field *x* in the NormalSample.csv file.

a) (5 points) According to Izenman (1991) method, what is the recommended bin-width for the histogram of x?
SOLUTION:

The recommended Bin width for histogram of x is: 0.3998667554864774

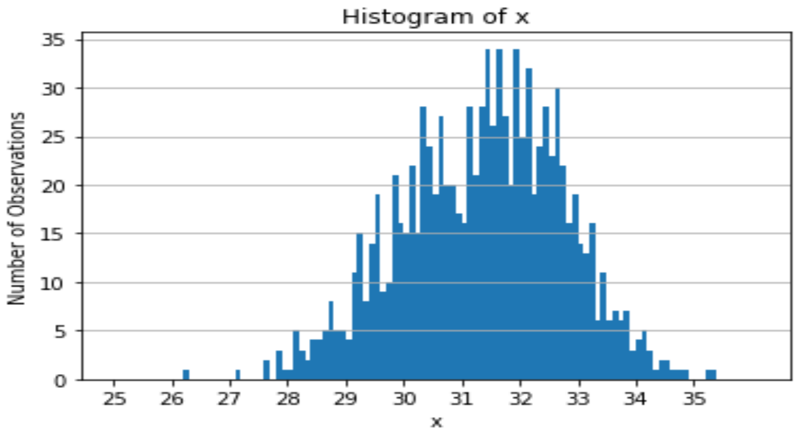
b) (5 points) What are the minimum and the maximum values of the field x?
SOLUTION:

Minimum value= 26.3
Maximum value = 35.4

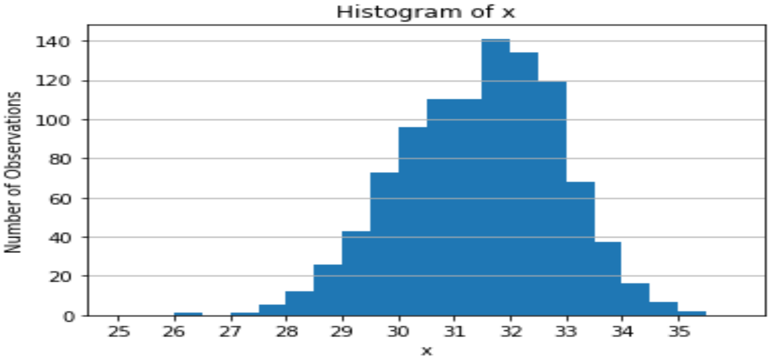
c) (5 points) Let a be the largest integer less than the minimum value of the field x, and b be the smallest integer greater than the maximum value of the field x. What are the values of a and b?
SOLUTION:

26.3
35.4
Value of a is= 26
Value of b is= 36

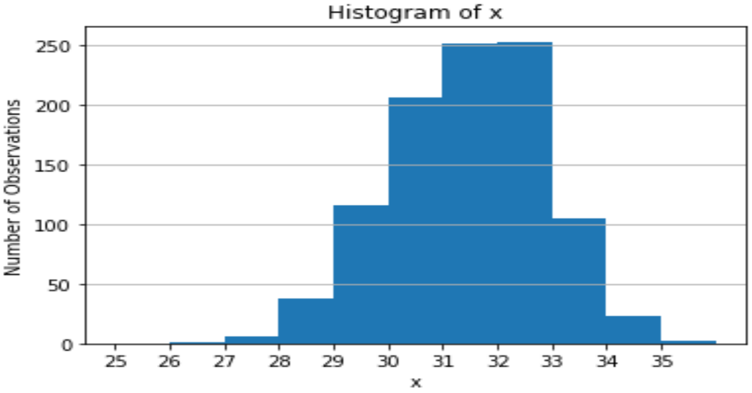
d) (5 points) Use h = 0.1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.



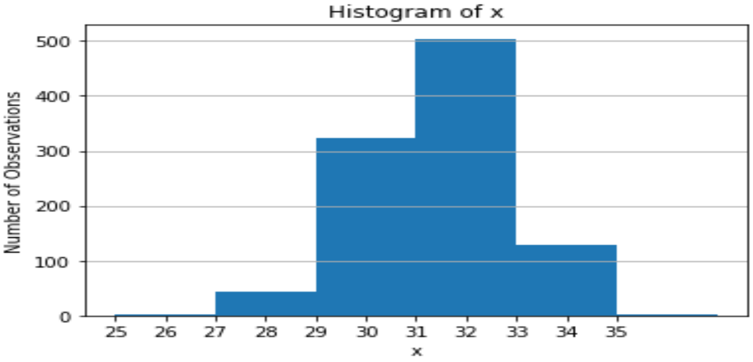
e) (5 points) Use h = 0.5, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.



f) (5 points) Use h = 1, minimum = a and maximum = b. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.



g) (5 points) Use h = 2, minimum = a and maximum = a. List the coordinates of the density estimator. Paste the histogram drawn using Python or your favorite graphing tools.



h) (5 points) Among the four histograms, which one, in your honest opinions, can best provide your insights into the shape and the spread of the distribution of the field x? Please state your arguments.

Histograms are used for pictorial representation of the data. Histograms help in obtaining normal distribution of the dataset.
Among the above four histograms, the first histogram with h=0.5 provides insights into the shape and the spread of the distribution of the field x. The histogram contains 11 bins from 25 to 35 with a bin size of 1. The x-axis contains the 'x' variable. And the y-axis contains the 'Number of observations. It gives more information about the data as compared to the other histograms.
Frequency of occurrence of the bin is calculated as:
Frequency = height of the bin * width of the bin

The height of the bin reflects the frequency therefore among the four histograms h=0.5 is best suited.

## Question 2 (20 points)

Use in the NormalSample.csv to generate box-plots for answering the following questions.

a) (5 points) What are the five-number summary of x? What are the values of the 1.5 IQR whiskers?
SOLUTION:
Five number summary:

```
Min: 26.300
Q1: 30.400
Median: 31.500
Q3: 32.400
Max: 35.400
```

```
Value of Lower whisker is: 27.4
 Value of max whisker is: 35.4
```

b) (5 points) What are the five-number summary of x for each category of the group? What are the values of the 1.5 IQR whiskers for each category of the group?

For group=0

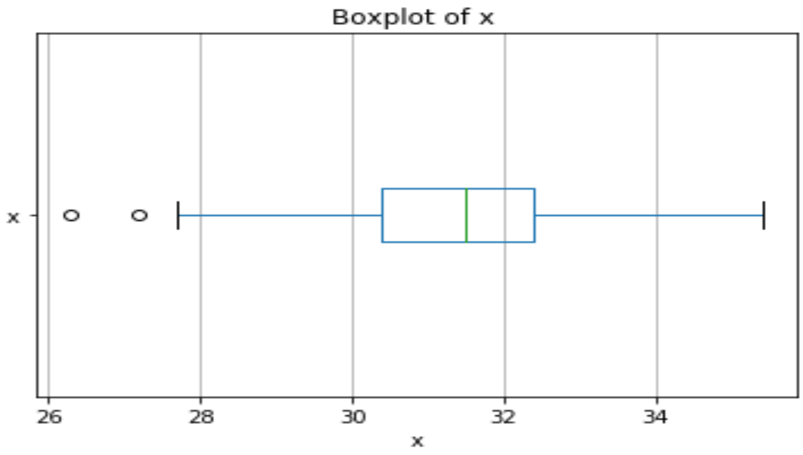
```
[315 rows x 3 columns]

median is:  30.0
mean is:  30.004126984126987
min_value 26.3
max_value 32.2
quartiles are [29.4] [30.] [30.6]
 Value of min whisker is: 0.25    27.4
Name: x, dtype: float64
```

```
q3 is 0.75    32.4
Name: x, dtype: float64
Value of max whisker is: 0.75    35.4
Name: x, dtype: float64
```

For group=1

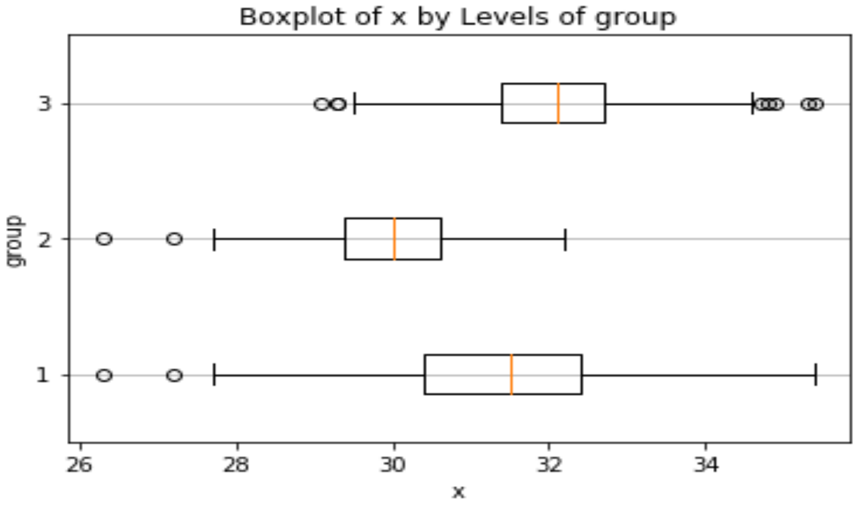
```
[686 rows x 3 columns]
median is:  32.1
mean is:  32.06224489795918
min_value 29.1
max_value 35.4
quartiles are [31.4] [32.1] [32.7]
Value of min whisker is: 0.25    27.4
Name: x, dtype: float64
q3 is 0.75    32.4
Name: x, dtype: float64
Value of max whisker is: 0.75    35.4
Name: x, dtype: float64
```

c) (5 points) Draw a boxplot of x (without the group) using the Python boxplot function. Can you tell if the Python's boxplot has displayed the 1.5 IQR whiskers correctly?



The min and max whiskers are 27.4 and 29.4. The min whisker is displayed correctly. But not the max whisker.

d) (5 points) Draw a graph where it contains the boxplot of x, the boxplot of x for each category of Group (i.e., three boxplots within the same graph frame). Use the 1.5 IQR whiskers, identify the outliers of x, if any, for the entire data and for each category of Group.
*Hint: Consider using the CONCAT function in the PANDA module to append observations*.



## Question 3 (40 points)

The data, FRAUD.csv, contains results of fraud investigations of 5,960 cases. The binary variable FRAUD indicates the result of a fraud investigation: 1 = Fraudulent, 0 = Otherwise. The other interval variables contain information about the cases.

1. TOTAL_SPEND: Total amount of claims in dollars
2. DOCTOR_VISITS: Number of visits to a doctor
3. NUM_CLAIMS: Number of claims made recently
4. MEMBER_DURATION: Membership duration in number of months
5. OPTOM_PRESC: Number of optical examinations
6. NUM_MEMBERS: Number of members covered

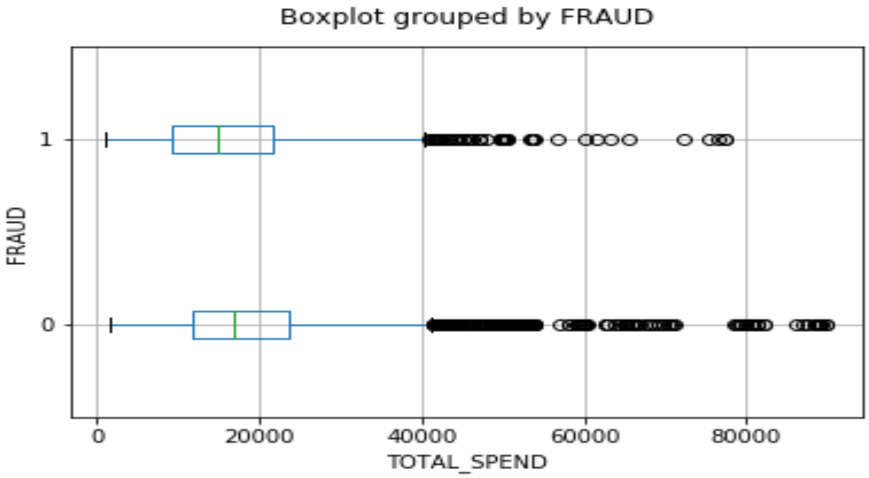
You are asked to use the Nearest Neighbors algorithm to predict the likelihood of fraud.

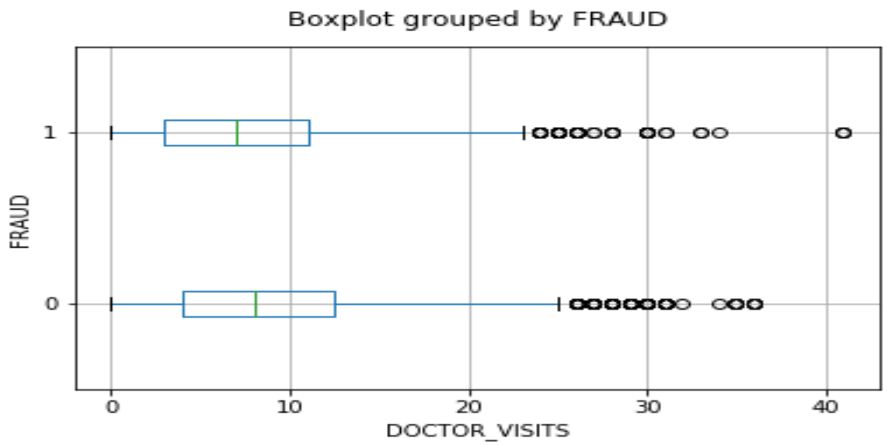
a) (5 points) What percent of investigations are found to be fraudulent? Please give your answer up to 4 decimal places.

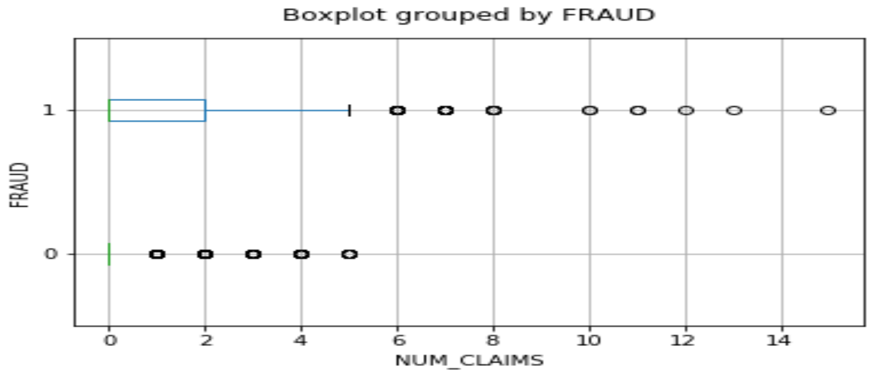
percentage of fraudulent investigations are: 19.949664

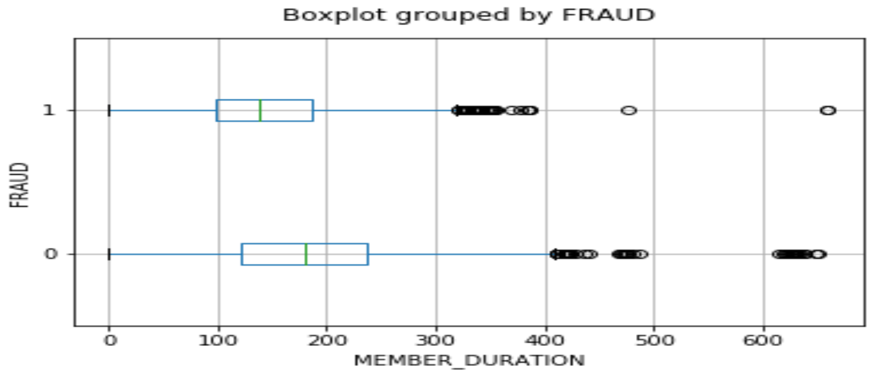
b) (5 points) Use the BOXPLOT function to produce horizontal box-plots. For each interval variable, one box-plot for the fraudulent observations, and another box-plot for the non-fraudulent observations. These two box-plots must appear in the same graph for each interval variable.

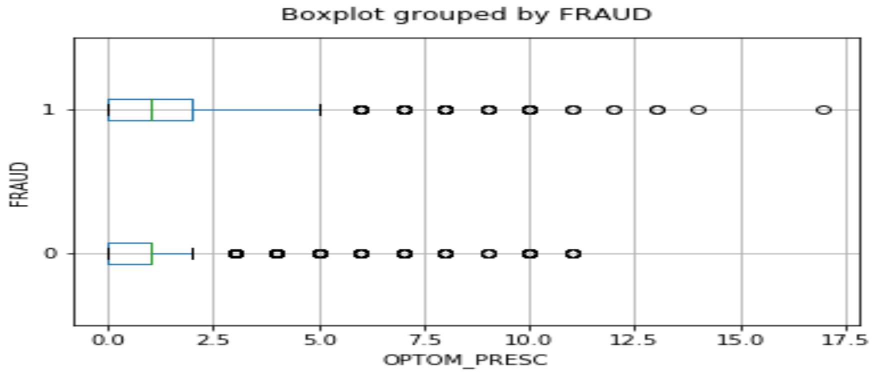
SOLUTION:

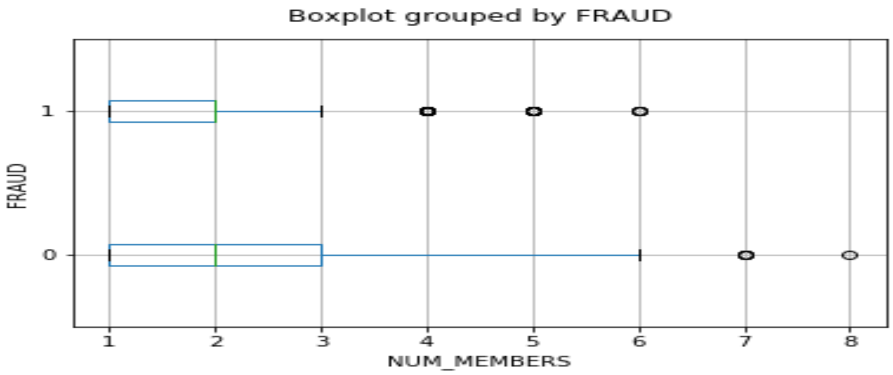



Boxplot grouped by FRAUD


Boxplot grouped by FRAUD


Boxplot grouped by FRAUD


Boxplot grouped by FRAUD



c) (10 points) Orthonormalize interval variables and use the resulting variables for the nearest neighbor analysis. Use only the dimensions whose corresponding eigenvalues are greater than one.

i. (5 points) How many dimensions are used?

Solution:

The orthonormalization of x=

```
[[-6.56324665e-04  9.39352141e-03  1.39590283e-02 -6.64664861e-03
   1.02081629e-02 -5.96859502e-03]
 [-7.75702220e-04  1.22658834e-02  5.16174400e-03  8.51930607e-04
   5.01932025e-03  2.09672310e-02]
 [-8.95075830e-04  1.50348109e-02 -1.71350853e-03 -7.38335310e-03
   1.97528525e-02 -7.64597676e-03]
 ...
 [-5.31896971e-02 -4.74021952e-02 -7.13245766e-03  2.75078514e-02
  -1.62580211e-02  7.18408819e-05]
 [-5.35474776e-02 -4.76625006e-02 -9.17125411e-03  2.76213381e-02
  -1.62154130e-02  1.80147801e-04]
 [-5.36071324e-02 -4.70861917e-02 -7.81347172e-03  2.93391341e-02
  -2.73884697e-02  2.21157680e-03]]
(5960, 6)
```

ii. (5 points) Please provide the transformation matrix? You must provide proof that the resulting variables are actually orthonormal.

Also Expect an Identity Matrix =

```
[[ 1.00000000e+00 -1.11022302e-16  9.67108338e-17 -7.63278329e-17
   1.99493200e-17 -7.91467586e-18]
 [-1.11022302e-16  1.00000000e+00  1.83447008e-16  2.25514052e-17
  -1.38777878e-17 -3.03576608e-18]
 [ 9.67108338e-17  1.83447008e-16  1.00000000e+00 -6.67868538e-17
  -7.91467586e-18  2.55465137e-17]
 [-7.63278329e-17  2.25514052e-17 -6.67868538e-17  1.00000000e+00
  -9.10729825e-17  1.63660318e-16]
 [ 1.99493200e-17 -1.38777878e-17 -7.91467586e-18 -9.10729825e-17
   1.00000000e+00  3.25748543e-16]
 [-7.91467586e-18 -3.03576608e-18  2.55465137e-17  1.63660318e-16
   3.25748543e-16  1.00000000e+00]]
```

```
t(fraud_x) * fraud_x =
 [[2812184770000    1040176400     42913200   20404919400     134771800
     220035900]
 [   1040176400       788159        23809      10264845        57654
        106717]
 [     42913200        23809         7922        448090         3459
          4765]
 [  20404919400     10264845       448090     232422585       1163391
       2121127]
 [    134771800        57654         3459       1163391         24460
         13581]
 [    220035900       106717         4765       2121127         13581
         29423]]
```

```
Eigenvalues of x =
 [6.84728061e+03 8.38798104e+03 1.80639631e+04 3.15839942e+05
 8.44539131e+07 2.81233324e+12]
```

```
Eigenvectors of x =
 [[-5.37750046e-06 -2.20900379e-05  3.62806809e-05 -1.36298664e-04
  -7.26453432e-03  9.99973603e-01]
 [ 6.05433402e-03 -2.69942162e-02  1.27528313e-02  9.99013423e-01
   3.23120126e-02  3.69879256e-04]
 [-9.82198935e-01  1.56454700e-01 -1.03312781e-01  1.14463687e-02
   1.62110700e-03  1.52596881e-05]
 [ 1.59310591e-04 -4.91894718e-03  3.11864824e-03 -3.25018102e-02
   9.99428355e-01  7.25592222e-03]
 [ 6.90939783e-02 -2.10615119e-01 -9.75101628e-01  6.26672294e-03
   2.19857585e-03  4.79234486e-05]
 [ 1.74569737e-01  9.64577791e-01 -1.95782843e-01  2.73038995e-02
   6.21788707e-03  7.82430481e-05]]
```

```
Transformation Matrix =
 [[-6.49862374e-08 -2.41194689e-07  2.69941036e-07 -2.42525871e-07
  -7.90492750e-07  5.96286732e-07]
 [ 7.31656633e-05 -2.94741983e-04  9.48855536e-05  1.77761538e-03
   3.51604254e-06  2.20559915e-10]
 [-1.18697179e-02  1.70828329e-03 -7.68683456e-04  2.03673350e-05
   1.76401304e-07  9.09938972e-12]
 [ 1.92524315e-06 -5.37085514e-05  2.32038406e-05 -5.78327741e-05
   1.08753133e-04  4.32672436e-09]
 [ 8.34989734e-04 -2.29964514e-03 -7.25509934e-03  1.11508242e-05
   2.39238772e-07  2.85768709e-11]
 [ 2.10964750e-03  1.05319439e-02 -1.45669326e-03  4.85837631e-05
   6.76601477e-07  4.66565230e-11]]


The Transformed x =
 [[ 5.96859502e-03  1.02081629e-02 -6.64664861e-03  1.39590283e-02
   9.39352141e-03  6.56324665e-04]
 [-2.09672310e-02  5.01932025e-03  8.51930607e-04  5.16174400e-03
   1.22658834e-02  7.75702220e-04]
 [ 7.64597676e-03  1.97528525e-02 -7.38335310e-03 -1.71350853e-03
   1.50348109e-02  8.95075830e-04]
 ...
 [-7.18408819e-05 -1.62580211e-02  2.75078514e-02 -7.13245766e-03
  -4.74021952e-02  5.31896971e-02]
 [-1.80147801e-04 -1.62154130e-02  2.76213381e-02 -9.17125411e-03
  -4.76625006e-02  5.35474776e-02]
 [-2.21157680e-03 -2.73884697e-02  2.93391341e-02 -7.81347172e-03
  -4.70861917e-02  5.36071324e-02]]


Expect an Identity Matrix =
 [[ 1.00000000e+00 -3.00432422e-16 -4.61219604e-16  5.45323877e-15
   1.20996962e-15 -1.28911638e-16]
 [-3.00432422e-16  1.00000000e+00 -6.44449771e-16 -2.76820667e-14
  -1.23512311e-15  7.78890841e-16]
 [-4.61219604e-16 -6.44449771e-16  1.00000000e+00  3.50891191e-15
   1.00613962e-16 -2.25514052e-16]
 [ 5.45323877e-15 -2.76820667e-14  3.50891191e-15  1.00000000e+00
   1.14860378e-14 -3.47812057e-15]
 [ 1.20996962e-15 -1.23512311e-15  1.00613962e-16  1.14860378e-14
   1.00000000e+00 -6.31439345e-16]
 [-1.28911638e-16  7.78890841e-16 -2.25514052e-16 -3.47812057e-15
  -6.31439345e-16  1.00000000e+00]]
```

d) (10 points) Use the NearestNeighbors module to execute the Nearest Neighbors algorithm using exactly five neighbors and the resulting variables you have chosen in c). The KNeighborsClassifier module has a score function.

i. (5 points) Run the score function, provide the function return value

Score value is: 0.8778523489932886

ii. (5 points) Explain the meaning of the score function return value.

Source: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.balanced_accuracy_score.html

The mean accuracy and the labels of the test data are returned by the source function. To help classifiers with the scoring metric that is suitable for each classifier, different scoring methods are returned. Different classification models return different metrics depending on their score method.

e) (5 points) For the observation which has these input variable values: TOTAL_SPEND = 7500, DOCTOR_VISITS = 15, NUM_CLAIMS = 3, MEMBER_DURATION = 127, OPTOM_PRESC = 2, and NUM_MEMBERS = 2, find its **five** neighbors. Please list their input variable values and the target values. *Reminder: transform the input observation using the results in c) before finding the neighbors*

Input variable names are:TOTAL_SPEND','DOCTOR_VISITS','NUM_CLAIMS','MEMBER_DURATION','OPTOM_PRESC','NUM_MEMBERS'

focal variables are: [[7500, 15, 3, 127, 2, 2]]

My Neighbors =

```
 [[2748 2173 2224  776   44]]
```

Transformation Matrix =

```
 [[-6.49862374e-08 -2.41194689e-07  2.69941036e-07 -2.42525871e-07
  -7.90492750e-07  5.96286732e-07]
 [ 7.31656633e-05 -2.94741983e-04  9.48855536e-05  1.77761538e-03
   3.51604254e-06  2.20559915e-10]
 [-1.18697179e-02  1.70828329e-03 -7.68683456e-04  2.03673350e-05
   1.76401304e-07  9.09938972e-12]
 [ 1.92524315e-06 -5.37085514e-05  2.32038406e-05 -5.78327741e-05
   1.08753133e-04  4.32672436e-09]
 [ 8.34989734e-04 -2.29964514e-03 -7.25509934e-03  1.11508242e-05
   2.39238772e-07  2.85768709e-11]
 [ 2.10964750e-03  1.05319439e-02 -1.45669326e-03  4.85837631e-05
   6.76601477e-07  4.66565230e-11]]
```

```
 My neighbors: [[ 588 2897 1199 1246  886]]
```

f) (5 points) Follow-up with e), what is the predicted probability of fraudulent (i.e., FRAUD = 1)? If your predicted probability is greater than or equal to your answer in a), then the observation will be classified as fraudulent. Otherwise, non-fraudulent. Based on this criterion, will this observation be misclassified?

```
nbrs.predict(focal * transf)

print_proba=nbrs.predict_proba(focal)

print("prediceted focal:",print_proba)
```

The predicted probability of focal with transf is [[0.8 0.2]]. In my opinion it is classified as Fraudulent. This data is not misclassified, since when the neighbors are considered the fraudulent cases are more.