A20474413
Jorge Gonzalez Lopez

Assignment 1

CS484 Spring 2021
February 05, 2021

Question 1

a)

- **Command Prompt Output**

```
count    1001.000000
mean       31.414585
std         1.397672
min        26.300000
25%        30.400000
50%        31.500000
75%        32.400000
max        35.400000
```

b)

- **Answer**
  0.40

- **Explanation**

  $bw = 2\,(Q3 - Q1)\,N^{\frac{-1}{3}}$

c)

- **Answer**
  1.00

- **Explanation**

```
def calcCD (Y, delta):
       maxY = np.max(Y)
       minY = np.min(Y)
       meanY = np.mean(Y)

       # Round the mean to integral multiples of delta
       middleY = delta * np.round(meanY / delta)
       # Determine the number of bins on both sides of the rounded mean
       nBinRight = np.ceil((maxY - middleY) / delta)
       nBinLeft = np.ceil((middleY - minY) / delta)
       lowY = middleY - nBinLeft * delta

       # Assign observations to bins starting from 0
       m = nBinLeft + nBinRight
       BIN_INDEX = 0;
       boundaryY = lowY
```

```
    # Assign observations to bins starting from 0
    m = nBinLeft + nBinRight
    BIN_INDEX = 0;
    boundaryY = lowY
    for iBin in np.arange(m):
        boundaryY = boundaryY + delta
        BIN_INDEX = np.where(Y > boundaryY, iBin+1, BIN_INDEX)

    # Count the number of observations in each bins
    uBin, binFreq = np.unique(BIN_INDEX, return_counts = True)

    # Calculate the average frequency
    meanBinFreq = np.sum(binFreq) / m
    ssDevBinFreq = np.sum((binFreq - meanBinFreq)**2) / m
    CDelta = (2.0 * meanBinFreq - ssDevBinFreq) / (delta * delta)
    return(m, middleY, lowY, CDelta)
```

d)

- **Command Prompt Output**

```
The midpoints are: [26.5 27.5 28.5 29.5 30.5 31.5 32.5 33.5 34.5 35.5]
```

- **Figure**

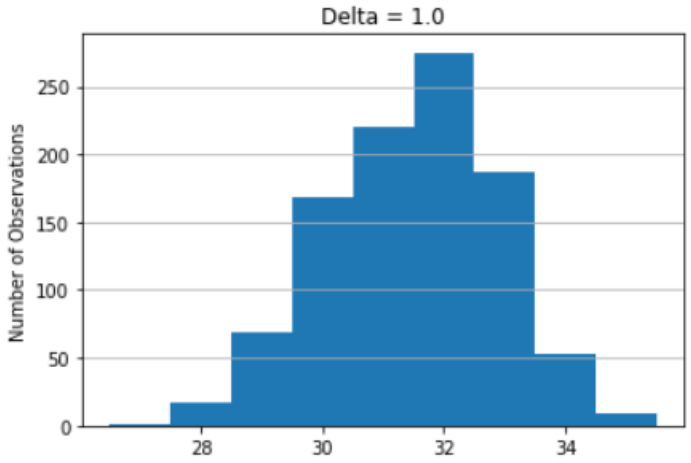

Figure 1: Vertical chart of the density estimator of the field x.

Question 2

a)

- **Command Prompt Output**

```
a) The five-number summary of x for each category of the group is:
group = 0 -> Min: 26.3, Q1: 29.4, Q2: 30.0, Q3: 30.6 and max: 32.2
group = 1 -> Min: 29.1, Q1: 31.4, Q2: 32.1, Q3: 32.7 and max: 35.4

And the values of the 1.5 IQR whiskers are:
group = 0 -> lower whisker = 27.6 and the upper whisker: 32.4
group = 1 -> lower whisker = 29.45 and the upper whisker: 34.65
```

- **Figure**

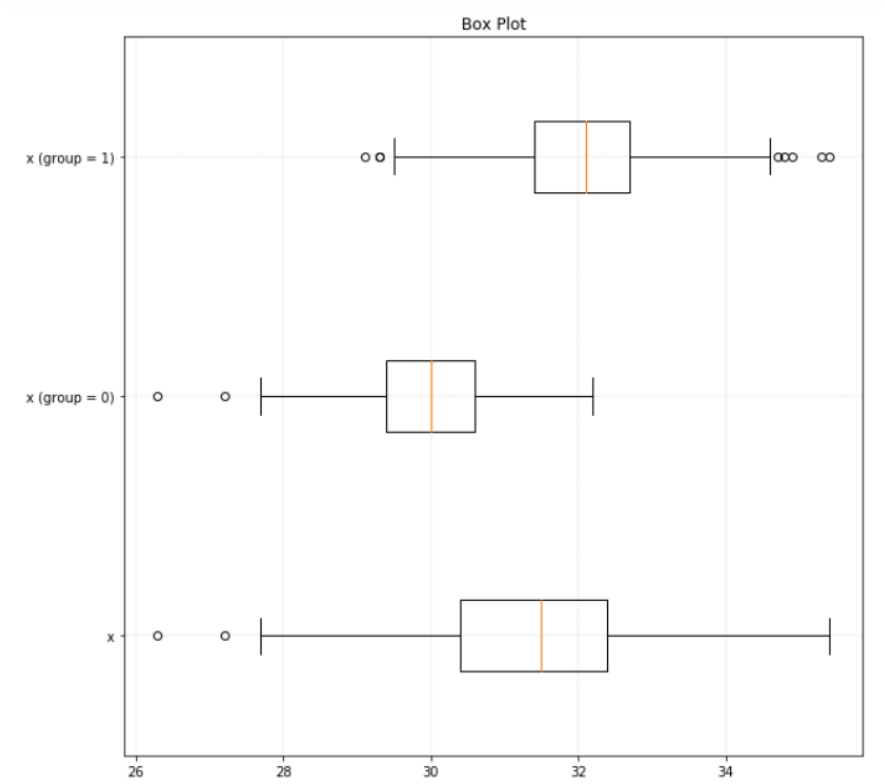

Figure 2: Horizontal overall boxplot of x and the boxplots of x for each category of group.

- **Explanation:**

The five-number summary are: the minimum, the first quartile (Q1), the median (Q2), the third quartile (Q3) and the maximum.
Then the Interquartile Range is: IQR = Q3 - Q1
And, finally, the lower whisker extends to Q1 – 1.5*IQR and the upper whisker to Q3 + 1.5*IQR.

b)

- **Command Prompt Output**

```
Outliers of x for the entire data:
70     27.2
295    26.3
Name: x, dtype: float64

Outliers of x for the group = 0:
70     27.2
295    26.3
Name: x, dtype: float64

Outliers of x for the group = 1:
30     35.3
107    29.3
297    35.4
812    34.9
846    34.7
907    34.8
938    29.3
975    29.1
Name: x, dtype: float64
```

- **Explanation:**

The outliers can be identified if their values are lower than the lower whisker or higher than the upper whisker.

```
data_g[(data_g < lower_whisker_g) | (data_g > upper_whisker_g)]
```

Question 3

a)

- **Answer**
  19.9497 %
- **Explanation:**

  It has been calculated by counting the number of frauds (= 1) of the total of investigations.

```
t = df['FRAUD'].value_counts(normalize = True)
np.round(t[1]*100, decimals=4)
```

b)

- **Answer**
  2
- **Command Prompt Output**

```
Number of Dimensions =  2
Eigenvalues of x greater than one =
 [6.84728061e+03 8.38798104e+03 1.80639631e+04 3.15839942e+05
 8.44539131e+07 2.81233324e+12]

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]]
```

- **Explanation:**

  The columns used are: ['TOTAL_SPEND', 'DOCTOR_VISITS', 'NUM_CLAIMS', 'MEMBER_DURATION', 'OPTOM_PRESC', 'NUM_MEMBERS'].

  The column ‘CASE_ID’ is subtracted because it does not provide any valuable information and the column ‘FRAUD’ is also subtracted as it corresponds to the target value.

Once the values (x) are transformed with the transformation matrix, if $x^T x$ provides an Identity matrix, its values are orthonormal:

```
Expect an Identity Matrix =
 [[ 1.  0. -0.  0.  0. -0.]
 [ 0.  1.  0. -0. -0.  0.]
 [-0.  0.  1. -0. -0. -0.]
 [ 0. -0. -0.  1.  0. -0.]
 [ 0. -0. -0.  0.  1. -0.]
 [-0.  0. -0. -0. -0.  1.]]
```

c)

- **Answer**

Returns the mean accuracy on the given test data and labels. In this case, 0.87785.

- **Command Prompt Input and Output**

```
Y = df['FRAUD']
X = transf_matrix

model = KNeighborsClassifier(n_neighbors=5, metric = 'euclidean')

res = model.fit(X,Y)
preds = res.predict(X)

print(res.score(X,Y))



 -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -   -

0.8778523489932886
```

- **Explanation:**

When computing the function score to the data, it provides de probability of correctly classifying the investigations as a fraud.

d)

- **Answer**

Its five neighbors are: [588 2897 1199 1246 886]

- **Command Prompt Output**

	CASE_ID	FRAUD	TOTAL_SPEND	DOCTOR_VISITS	NUM_CLAIMS	MEMBER_DURATION	OPTOM_PRESC	NUM_MEMBERS
588	589	1	7500	15	3	127	2	2
2897	2898	1	16000	18	3	146	3	2
1199	1200	1	10000	16	3	124	2	1
1246	1247	1	10200	13	3	119	2	3
886	887	1	8900	22	3	166	1	2

e)

- **Answer**

100 %

- **Explanation:**

The probability of classification as a fraud is of 100 % because it has the same values for every column that another investigation that is included in the data that has been trained.

Question 4

a)

- **Figure**

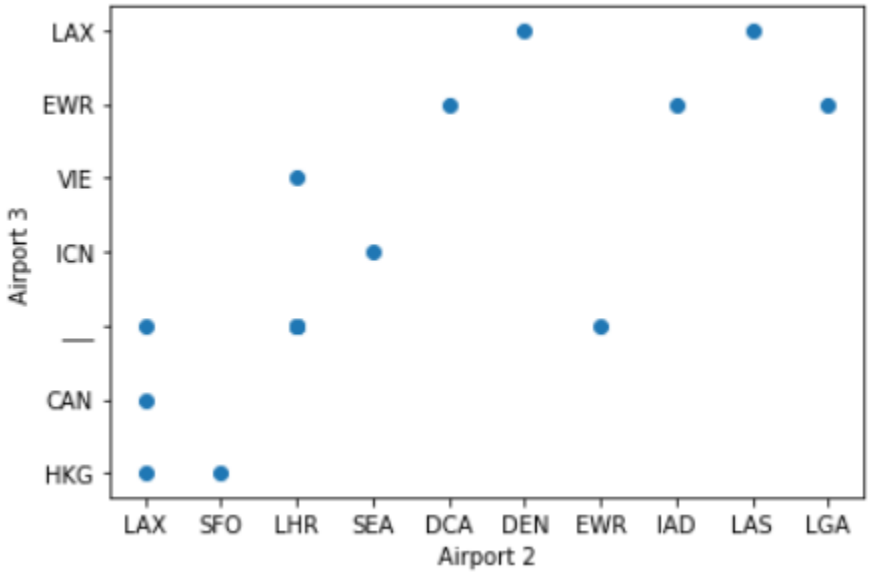

Figure 3: Scatterplot of Airport 3 (y-axis) versus Airport 2 (x-axis).

b)

- **Command Prompt Output**

```
LAX     5
___     5
LHR     4
EWR     4
HKG     2
SFO     1
VIE     1
IAD     1
LAS     1
LGA     1
DEN     1
SEA     1
CAN     1
DCA     1
ICN     1
```

c)

- **Answer**

The cosine distances are: [0.5 1. 0.5 1. 1. 0.5 1. 1. 1. 0.5 1. 1. 0.5 0.5 1.]

- **Explanation:**

The cosine distance is calculated as: $1 - < x_i, x_j > /(||x_i|| \cdot ||x_j||)$

The norm of all the vectors is $\sqrt{2}$ and the product of two norms is 2.

Therefore, if no airports (Airport 2 or 3) match between two vectors, the distance would be:

1 – 0/2 = 1

And if one airport matchs, the distance would be:

1 – 1/2 = 0.5

d)

- **Command Prompt Output**

	Flight	Carrier 1	Carrier 2	Airport 1	Airport 2	Airport 3	Airport 4
0	A	American	Cathay Pacific	ORD	LAX	HKG	PVG
2	C	American	China Southern	ORD	LAX	CAN	PVG
5	F	Delta	___	ORD	SEA	ICN	PVG
9	J	United	___	ORD	DEN	LAX	PVG
12	M	United	___	ORD	LAS	LAX	PVG
13	N	United	___	ORD	LAX	___	PVG

- **Explanation:**

All flights which include LAX or ICN as their Airport 2 or Airport 3.