INFS 4203 / 7203 Data Mining

Tutorial 6: Anomaly Detection + Assignment 2 Q&A

Content

- 1. Density-based: LOF Calculation (Step-by-step)
- 2. Assignment 2 Q&A

Density-based Technique
Calculate the Local Outlier Factor (LOF)

- Given four points: $P_1(1,0)$, $P_2(2,0)$, $P_3(1,1)$, $P_4(2,2.5)$. Calculate the Local Outlier Factor (LOF) for each point and find the top-1 outliers. Use a **k** value of 2 and Euclidean Distance as the distance function.

LOF Calculation

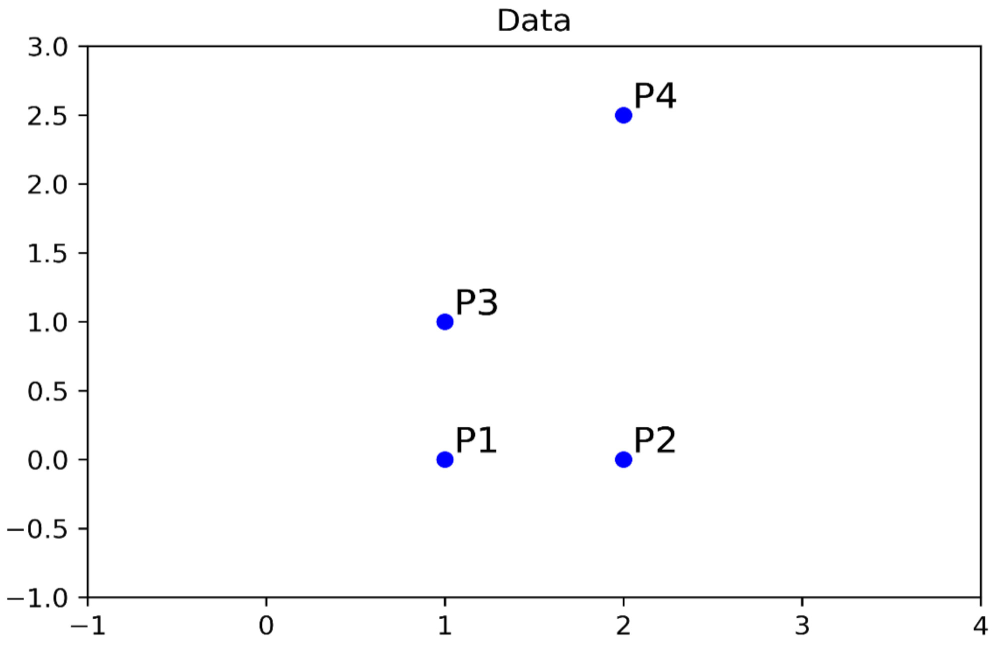
Background Knowledge:

- Density
- Average Relative Density (ard)

Steps of LOF calculation:

- Distance Matrix and k-nearest neighbourhood
- Reachability Distance (reachdist)
- Local Reachability Density (lrd)
- Average Relative lrd (arlrd)
- Anomaly Score

Distance Matrix and k-nearest neighbourhood



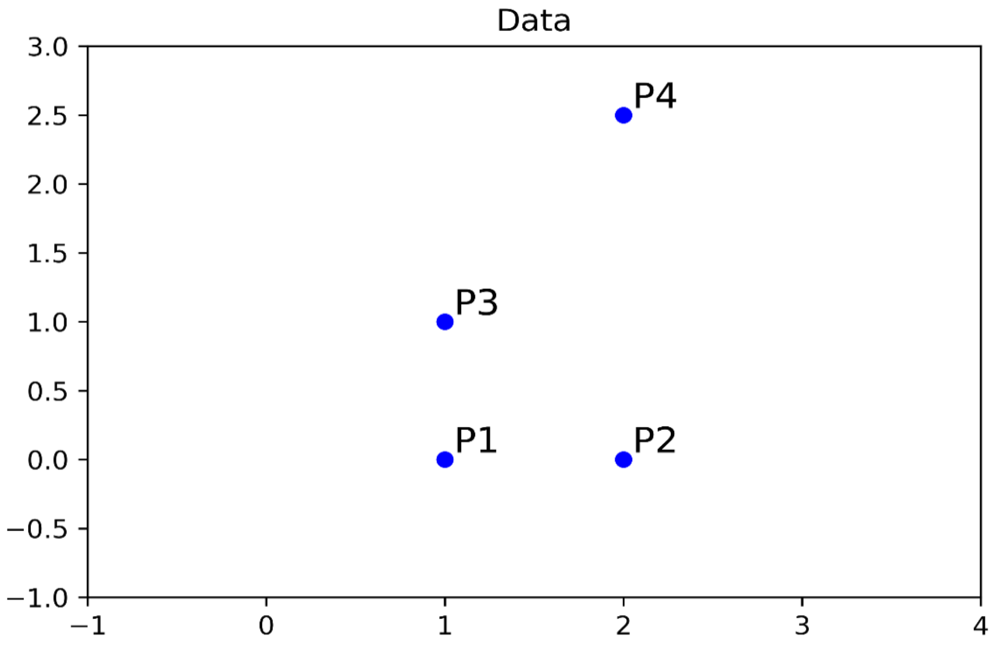
Distance Matrix (Euclidean)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

$if\ k = 2:$

k-nearest neighbourhood?

Distance Matrix and k-nearest neighbourhood



Distance Matrix (Euclidean)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$ $N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$ $N\{P_4, 2) = \{P_2, P_3\}$

Background: Density Calculation

Distance Matrix ($dist$)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

$if\ k = 2$:

$N\{P_1, 2) = \{P_2, P_3\}$
$N\{P_2, 2) = \{P_1, P_3\}$
$N\{P_3, 2) = \{P_1, P_2\}$
$N\{P_4, 2) = \{P_2, P_3\}$

$$density(\boldsymbol{x}, k) = \left(\frac{\sum_{\boldsymbol{y} \in N(\boldsymbol{x},k)} dist(\boldsymbol{x}, \boldsymbol{y})}{|N(\boldsymbol{x}, k)|} \right)^{-1}$$

	P_1	P_2	P_3	P_4
$density$	?	?	?	?

Background: Density Calculation

Distance Matrix ($dist$)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

$if\ k = 2$:

$N\{P_1, 2) = \{P_2, P_3\}$
$N\{P_2, 2) = \{P_1, P_3\}$
$N\{P_3, 2) = \{P_1, P_2\}$
$N\{P_4, 2) = \{P_2, P_3\}$

$$density(\boldsymbol{x}, k) = \left(\frac{\sum_{\boldsymbol{y} \in N(\boldsymbol{x}, k)} dist(\boldsymbol{x}, \boldsymbol{y})}{|N(\boldsymbol{x}, k)|} \right)^{-1}$$

	P_1	P_2	P_3	P_4
$density$	1	0.8290	0.8290	0.4650

$$density(P_1, 2) = \left(\frac{dist(P_1, P_2) + dist(P_1, P_3)}{|N(P_1, 2)|} \right)^{-1} = \left(\frac{1+1}{2} \right)^{-1} = 1$$

Background: Average Relative Density (ard)

	P_1	P_2	P_3	P_4
$density$	1	0.8290	0.8290	0.4650

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$

$N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$

$N\{P_4, 2) = \{P_2, P_3\}$

$$ard(\boldsymbol{x}, k) = \frac{density(\boldsymbol{x}, k)}{\sum_{\boldsymbol{y} \in N(\boldsymbol{x},k)} density(\boldsymbol{y}, \boldsymbol{k}) / |N(\boldsymbol{x}, k)|}$$

The average density in the *k-nearest neighbourhood*

	P_1	P_2	P_3	P_4
ard	?	?	?	?

Background: Average Relative Density (ard)

	P_1	P_2	P_3	P_4
$density$	1	0.8290	0.8290	0.4650

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$

$N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$

$N\{P_4, 2) = \{P_2, P_3\}$

$$ard(\boldsymbol{x}, k) = \frac{density(\boldsymbol{x}, k)}{\sum_{\boldsymbol{y} \in N(\boldsymbol{x},k)} density(\boldsymbol{y}, \boldsymbol{k}) / |N(\boldsymbol{x}, k)|}$$

The average density in the *k-nearest neighbourhood*

	P_1	P_2	P_3	P_4
ard	1.2060	0.9070	0.9070	0.5610

$$ard(P_1, 2) = \frac{density(P_1, 2)}{(density(P_2, 2) + density(P_3, 2))/|N(P_1, 2)|} = \frac{1}{(0.8290 + 0.8290)/2} = 1.2060$$

LOF Calculation

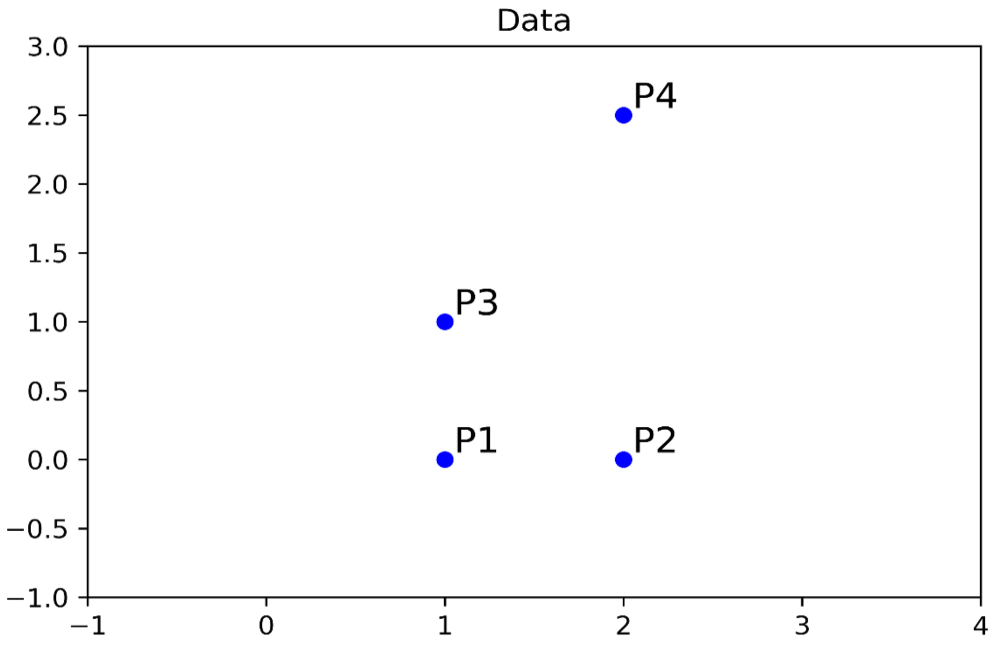
Background Knowledge:

- Density
- Average Relative Density (ard)

Steps of LOF calculation:

- Distance Matrix and k-nearest neighbourhood
- Reachability Distance (reachdist)
- Local Reachability Density (lrd)
- Average Relative lrd (arlrd)
- Anomaly Score

Distance Matrix and k-nearest neighbourhood



Distance Matrix (Euclidean)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$ $\quad$ $N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$ $\quad$ $N\{P_4, 2) = \{P_2, P_3\}$

Reachability Distance (reachdist)

$if\ k = 2:$
$N\{P_1, 2) = \{P_2, P_3\}$
$N\{P_2, 2) = \{P_1, P_3\}$
$N\{P_3, 2) = \{P_1, P_2\}$
$N\{P_4, 2) = \{P_2, P_3\}$

$$reachdist_k(y \leftarrow x) = \max\{dist(x,y), dist_k(y)\}$$

y's distance to its *k*th nearest neighbour

Distance Matrix ($dist$)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

Reachability Distance ($reachdist$)

	P_1	P_2	P_3	P_4
P_1				
P_2				
P_3				
P_4				

$if\ k = 2:$
$N\{P_1, 2) = \{P_2, P_3\}$
$N\{P_2, 2) = \{P_1, P_3\}$
$N\{P_3, 2) = \{P_1, P_2\}$
$N\{P_4, 2) = \{P_2, P_3\}$

Reachability Distance (reachdist)

$$reachdist_k(y \leftarrow x) = \max\{dist(x, y), dist_k(y)\}$$

y's distance to its kth nearest neighbour

Distance Matrix ($dist$)

	P_1	P_2	P_3	P_4
P_1	0	1	1	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	1.803
P_4	2.693	2.5	1.803	0

Reachability Distance ($reachdist$)

	P_1	P_2	P_3	P_4
P_1	0	**1.414**	**1.414**	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	2.5
P_4	2.693	2.5	1.803	0

$$reachdist_2(P_2 \leftarrow P_1) = \max\{dist(P_1, P_2), dist_2(P_2)\} = \max\{1, 1.414\} = 1.414$$

Local Reachability Density (lrd)

Reachability Distance ($reachdist$)

	P_1	P_2	P_3	P_4
P_1	0	**1.414**	**1.414**	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	2.5
P_4	2.693	2.5	1.803	0

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$

$N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$

$N\{P_4, 2) = \{P_2, P_3\}$

$$\overset{lrd}{\cancel{density}}(\boldsymbol{x}, k) = \left(\frac{\sum_{\boldsymbol{y} \in N(\boldsymbol{x},k)} \overset{reachdist}{\cancel{dist}}(\boldsymbol{x}, \boldsymbol{y})}{|N(\boldsymbol{x}, k)|}\right)^{-1}$$

	P_1	P_2	P_3	P_4
lrd				

Local Reachability Density (lrd)

Reachability Distance ($reachdist$)

	P_1	P_2	P_3	P_4
P_1	0	**1.414**	**1.414**	2.693
P_2	1	0	1.414	2.5
P_3	1	1.414	0	2.5
P_4	2.693	2.5	1.803	0

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$

$N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$

$N\{P_4, 2) = \{P_2, P_3\}$

$$lrd(\boldsymbol{x}, k) = \left(\frac{\sum_{\boldsymbol{y} \in N(\boldsymbol{x},k)} reachdist(\boldsymbol{x}, \boldsymbol{y})}{|N(\boldsymbol{x}, k)|}\right)^{-1}$$

	P_1	P_2	P_3	P_4
lrd	0.7070	0.8290	0.8290	0.4650

$$lrd(P_1, 2) = \left(\frac{reachdist(P_1, P_2) + reachdist(P_1, P_3)}{|N(P_1, 2)|}\right)^{-1} = \left(\frac{1.414 + 1.414}{2}\right)^{-1} = 0.707$$

…still a density calculation, but based on $reachdist$

Average Relative lrd (arlrd)

	P_1	P_2	P_3	P_4
lrd	0.7070	0.8290	0.8290	0.4650

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$

$N\{P_2, 2) = \{P_1, P_3\}$

$N\{P_3, 2) = \{P_1, P_2\}$

$N\{P_4, 2) = \{P_2, P_3\}$

$$\overset{arlrd}{\cancel{ard}}(\boldsymbol{x}, k) = \frac{\overset{lrd}{\cancel{density}}(\boldsymbol{x}, k)}{\sum_{\boldsymbol{y} \in N(\boldsymbol{x}, k)} \underset{lrd}{\cancel{density}}(\boldsymbol{y}, \boldsymbol{k}) / |N(\boldsymbol{x}, k)|}$$

	P_1	P_2	P_3	P_4
$arlrd$				
$1/arlrd$				

Average Relative lrd (arlrd)

	P_1	P_2	P_3	P_4
lrd	0.7070	0.8290	0.8290	0.4650

$if\ k = 2:$

$N\{P_1, 2) = \{P_2, P_3\}$
$N\{P_2, 2) = \{P_1, P_3\}$
$N\{P_3, 2) = \{P_1, P_2\}$
$N\{P_4, 2) = \{P_2, P_3\}$

$$arlrd(\boldsymbol{x}, k) = \frac{lrd(\boldsymbol{x}, k)}{\sum_{\boldsymbol{y} \in N(\boldsymbol{x}, k)} lrd(\boldsymbol{y}, \boldsymbol{k}) / |N(\boldsymbol{x}, k)|}$$

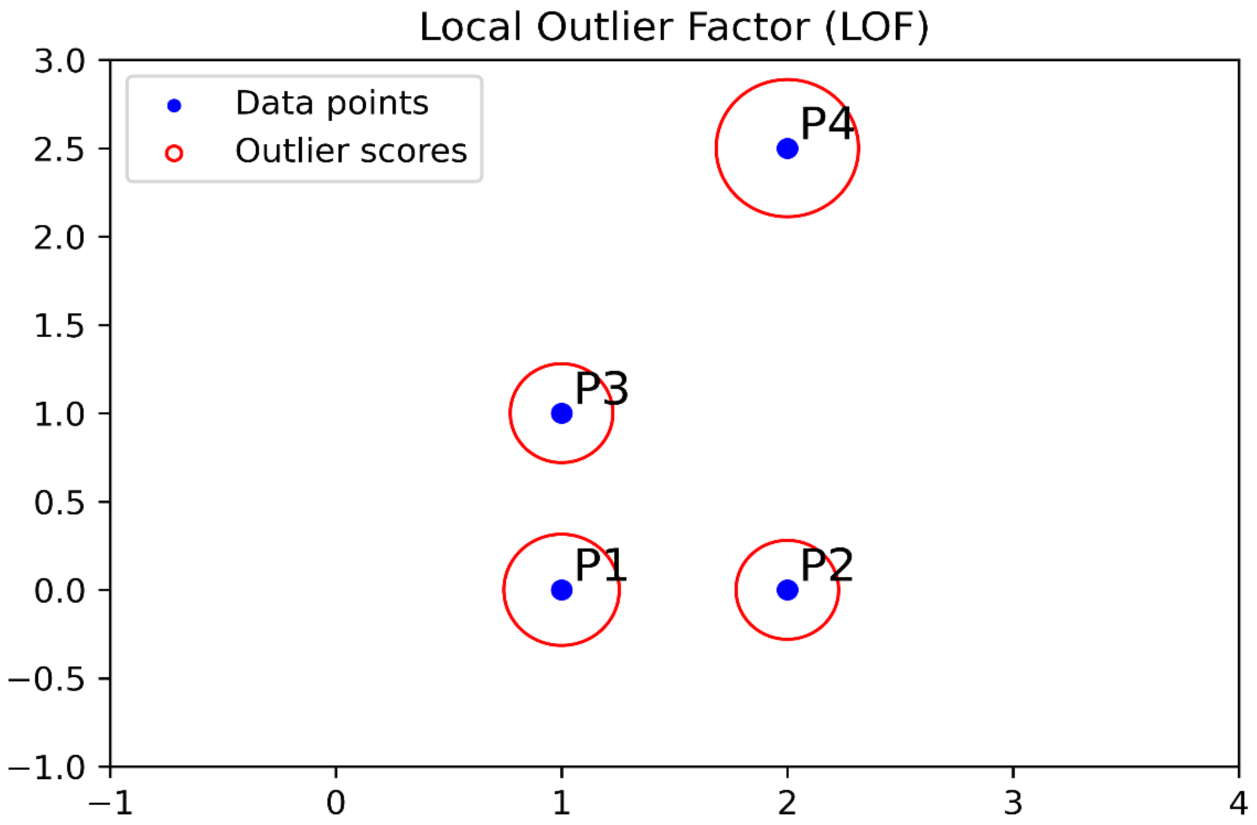
	P_1	P_2	P_3	P_4
$arlrd$	0.8530	1.0790	1.0790	0.5610
$1/arlrd$	1.1723	0.9268	0.9268	1.7825

$$arlrd(P_1, 2) = \frac{lrd(P_1,2)}{(lrd(P_2,2)+lrd(P_3,2))/|N(P_1,2)|} = \frac{0.7070}{(0.8290+0.8290)/2} = 0.853$$

Outliers

	P_1	P_2	P_3	P_4
$1/arlrd$	1.1723	0.9268	0.9268	1.7825

Anomaly score:
Reciprocal of density



Assignment 2 Q&A

- Coding Questions
- Calculation
- Statements True/False

Coding-related Questions

- The source code used in this assignment will be released.
- Data pre-processing (as demonstrated in Week 4 Tutorial):
  - Import libraries, load data, clean data, format transform
- Questions have four types:
  - Q1. Frequent itemset
  - Q2. Itemset support
  - Q3. Number of frequent itemsets
  - Q4. Strong rule

Q1. Frequent itemset

- For the data in groceries.csv file we used in Week 4 tutorial, if we set the **minimum support rate** to be **0.009**, then {**whole milk, tropical fruit, bread**} is a frequent itemset.

```
# define the MIN_SUPP
MIN_SUPP = 0.009

# apply the defined apriori algorithm
freq_set = apriori(df, min_support=MIN_SUPP,use_colnames=True)

check_set = ['whole milk', 'tropical fruit', 'bread']

# Select the idx from the frequent set based on the given check_set
itemset_idx = freq_set.index[freq_set['itemsets'] == frozenset(check_set)].tolist()
if itemset_idx==[]: # given check_set does not exist in the frequent set
    print('Not frequent!')
else:
    print('Found at location '+str(itemset_idx[0]))
```

(Partial code)

```
Not frequent!
```

Answer: False

Q2. Itemset support

- In the groceries.csv file we used in Week 4 tutorial, what is the **support rate** for itemset **{other vegetables, whipped/sour cream, yogurt}** (round to four decimal places)

(Partial code)

```
# define the MIN_SUPP
MIN_SUPP = 0.005

# apply the defined apriori algorithm
freq_set = apriori(df, min_support=MIN_SUPP,use_colnames=True)

print('Done!')
```

Apply Apriori

```
check_set = ['other vegetables', 'whipped/sour cream', 'yogurt']

# Select the idx from the frequent set based on the given check_set
itemset_idx = freq_set.index[freq_set['itemsets'] == frozenset(check_set)].tolist()
if itemset_idx==[]: # given check_set does not exist in the frequent set
    print('Not frequent!')
else:
    print('Found at location '+str(itemset_idx[0]))
    print(freq_set.loc[[itemset_idx[0]], ['support', 'itemsets']])
```

Check set

Supp rate

Answer: 0.0102

Q3. Number of frequent itemsets

- In the groceries.csv file we used in Week 4 tutorial, how many frequent itemsets are there if we set the **minimum support rate** threshold to be **0.01**?

```
# define the MIN_SUPP
MIN_SUPP = 0.01

# apply the defined apriori algorithm
freq_set = apriori(df, min_support=MIN_SUPP,use_colnames=True)

print(freq_set)
```

(Partial code)

Answer: 333

Q4. Strong rule

- For the data in groceries.csv file we used in Week 4 tutorial, if we set the **minimum support rate** to be **0.009** and the **minimum confidence** rate to be **0.2**, then {bottled water}-> {whole milk} is a strong rule

```
# define the MIN_SUPP
MIN_SUPP = 0.009

# apply the defined apriori algorithm
freq_set = apriori(df, min_support=MIN_SUPP,use_colnames=True)

# Specify the content of X and Y
X = ['bottled water']
Y = ['whole milk']

# Get the confidence
get_rule_confidence(freq_set, X, Y)
```

Specify the rule

(Partial code)

```
The confidence of rule {['bottled water']} -> {['whole milk']} is: 0.310948
```

Answer: True

Calculation – DBSCAN Elbow method

- If we set **MinPts = 2**, what would the **Eps** be using the Elbow method for the following dataset (rounding to the nearest integer)

	x	y
P1	119	508
P2	83	490
P3	413	454
P4	395	448
P5	416	427
P6	401	424
P7	284	193

How to determine Eps and MinPts

One recommended Elbow method:

- Fix MinPts to be *k*, (e.g., k=2)
- Calculate all points' distances to their (*k*-1)th nearest point
- Sort the distance in ascending order and plot them
- Find the "elbow" point, whose corresponding distance is Eps

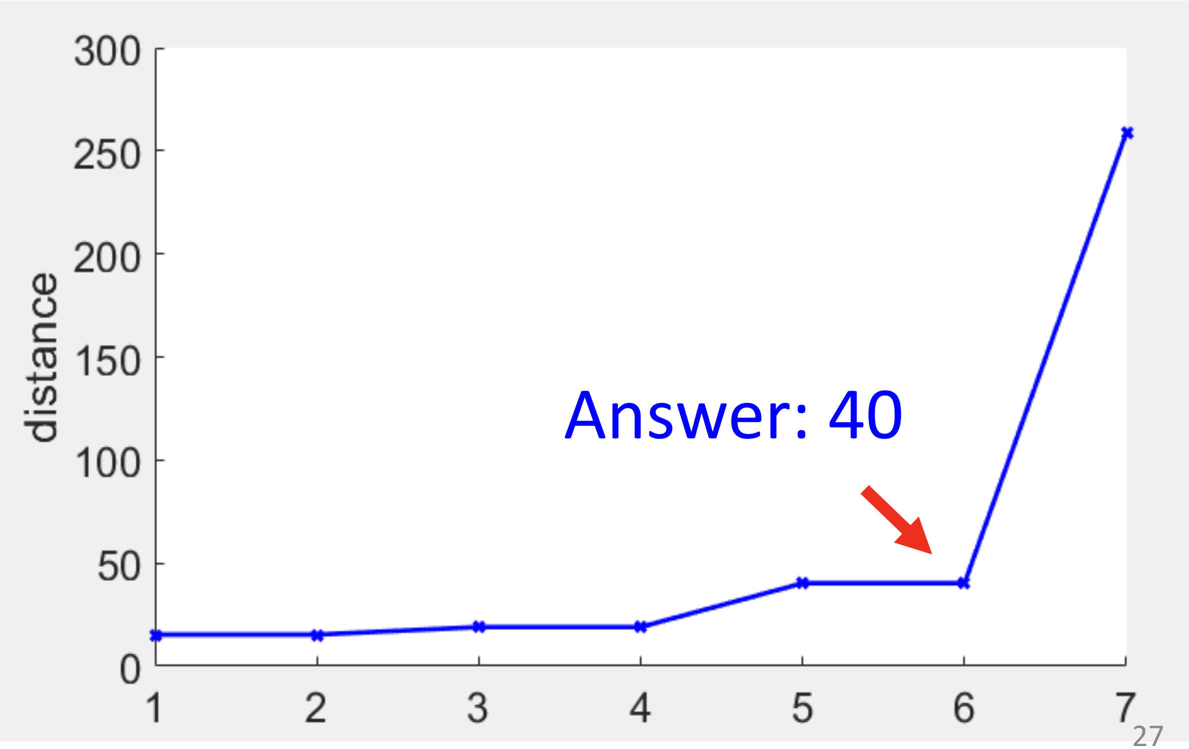
Calculation – DBSCAN Elbow method (Cont'd)

- If we set **MinPts = 2**, what would the **Eps** be using the Elbow method for the following dataset (rounding to the nearest integer)

Points	Distance to the **1st** Nearest Point
P5	15
P6	15
P3	18
P4	18
P1	40
P2	40
P7	258

Small

Large



Calculation – K-means Elbow Method

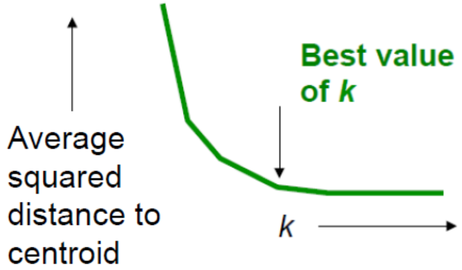
- To run a k-means algorithm, you need to specify k beforehand. If we use the elbow method to determine k, what is the selected k for the following data:

	x	y
P1	302	550
P2	158	469
P3	164	454
P4	359	448
P5	347	427
P6	245	355
P7	242	334

Pain of *k*-means: How to decide *k*

- **Elbow method**: try different *k* and see the average distance to centroid

$$\frac{1}{n}\sum_{i=1}^{k}\sum_{x\in C_i} dist(x, c_i)^2$$

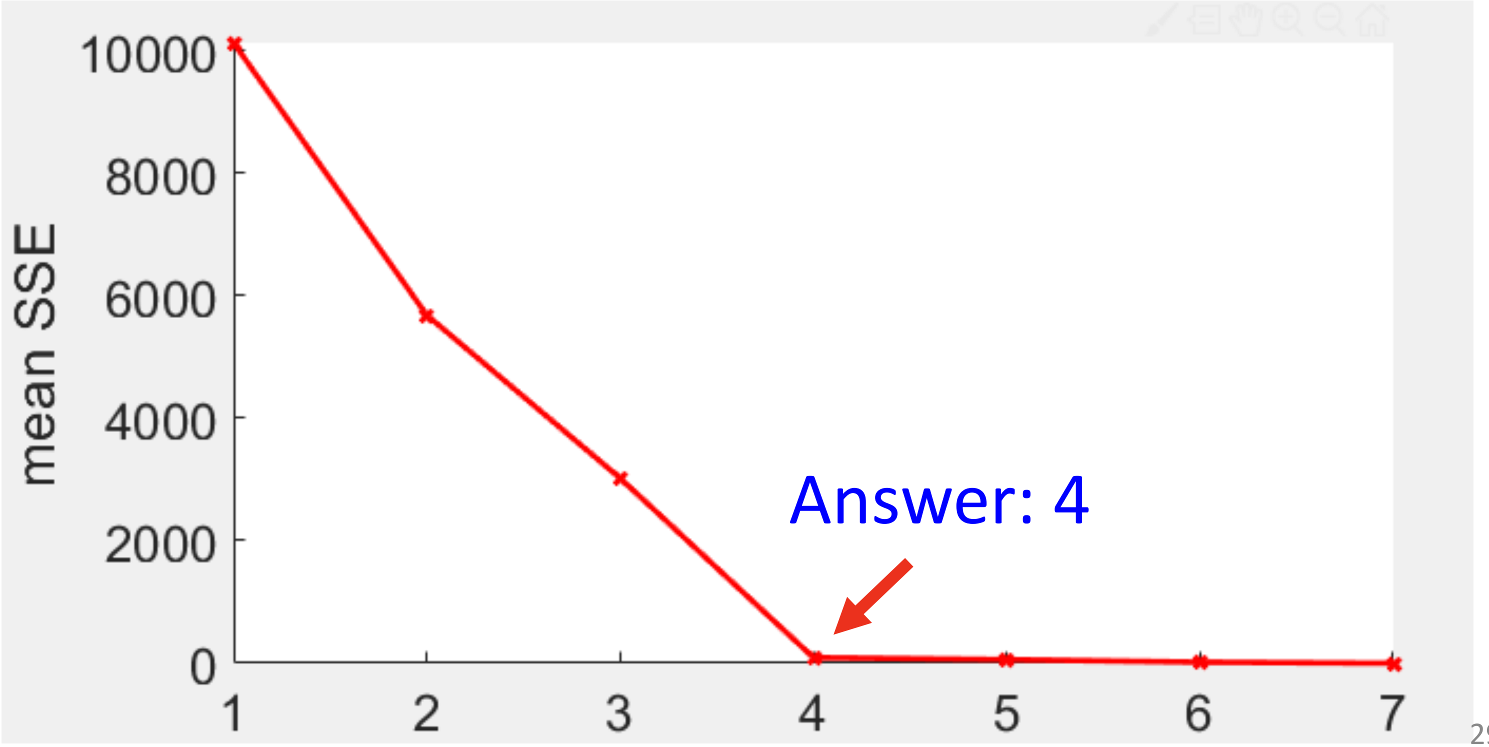


- Try *k* from 2 to $\sqrt{n}$ (n is the number of all data points)

Calculation – K-means Elbow Method (Cont'd)

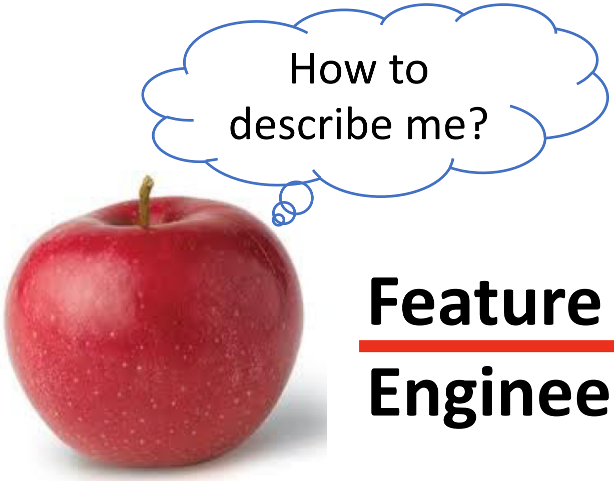
- To run a k-means algorithm, you need to specify k beforehand. If we use the elbow method to determine k, what is the selected k for the following data:

	x	y
P1	302	550
P2	158	469
P3	164	454
P4	359	448
P5	347	427
P6	245	355
P7	242	334



Statements True/False

- 1. Feature engineering is the process to do feature selection among existing features. False
  - Feature engineering is the process to **extract features**. It is an important pre-processing procedure.



object

Raw Data

Feature Engineering

feature/attribute

name	color	shape	weight	PoO	assortment	transport	preservation	growing	weather	taste
A1	red	round	200	Fuji	H	express	frozen	150	sunny	sweet

Features / Attributes

Forms of attribute

- Numerical:
  - The values of the attribute is to indicate the **quantity** of some predefined unit.
- Nominal
  - The values of the attribute are **symbols**, which is used to distinguish each other.
- Ordinal
  - The values of the attribute is to indicate certain **ordering relationship** resided in the attribute.

Statements True/False

- 2. If we use {very sweet, sweet, not sweet} to describe the taste of an apple, we have to use three numerical features to transform the taste feature into numerical. False
  - "Taste" is an ordinal feature to describe *three levels* of sweetness.
  - We could transfer it into numerical form: Very sweet: 5; sweet: 3; not sweet: 1. But *one* numerical feature will be sufficient.

	Taste
Pink Lady	Very sweet
Granny Smith	Not sweet
Golden Delicious	Sweet

ordinal

→

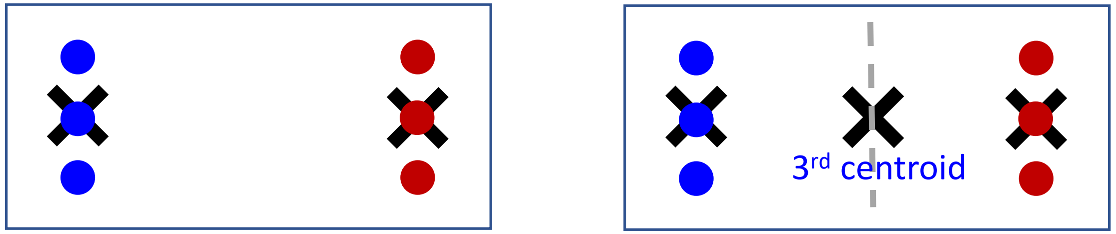
	Taste
Pink Lady	5
Granny Smith	1
Golden Delicious	3

numerical

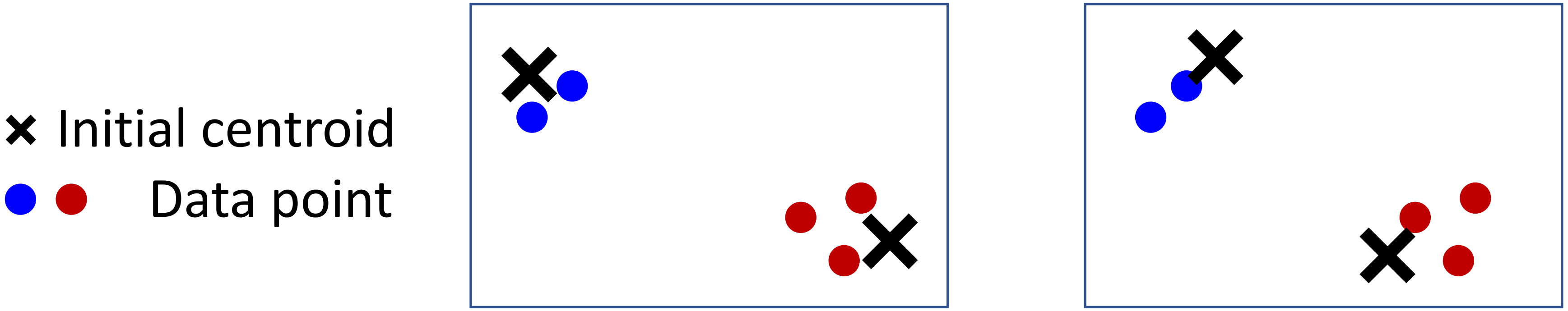
Statements True/False

- 3.1 Different k always gives different clustering results. False
  - Not always. Sometimes different k gives same clustering result.

    An intuitive example:



- 3.2 Different initialization always gives different results. False
  - Not always. Although the initial centroids are different, they may still converge into same clustering result. An intuitive example:

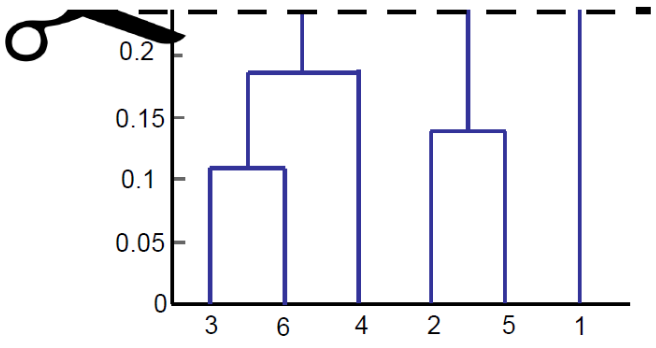


Statements True/False

- 4. We need to fix the “k” parameter for the k-means method. But we do not need to fix any parameter for the AGNES method to get a clustering result. False
- For the parameter "k", since there is some inconsistent in "Parameters in Python Function", this question will not be marked.
- We now provide an updated version (next slide).

Statements True/False

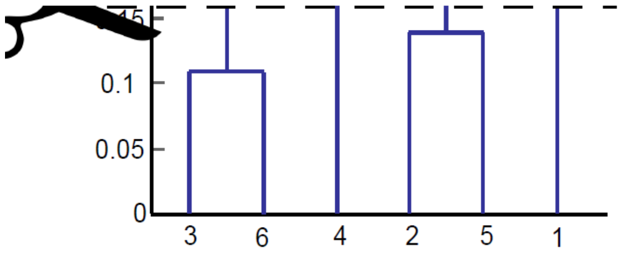
- 4. *(Updated)* We would like to partition a dataset into 4 clusters. The K-Means method would require setting the "k" parameter, but the AGNES method does not require any parameter to be set. Parameter refers to any user-specified measurable or numerical factors. False
  - The dendrogram shows the AGNES algorithm's output.
  - We still need to choose where to cut to produce the clustering result.



Clustering result:

{3,6,4}, {2,5}, {1}

3 clusters



Clustering result:

{3,6}, {4}, {2,5}, {1}

4 clusters

Statements True/False

- 5.1 Features are used to describe the cluster in clustering technique. False
- 5.2 Attributes are used to describe the cluster. False
  - Features/Attributes are used to describe the objects.

Statements True/False

- 6.1 K-means algorithm is simple and straightforward. Although it does not work the best on all datasets, it is the first choice to do clustering. False
  - It not always be the first choice. For example, k-means doesn’t work well on “spiral” dataset in Week 5’s tutorial.
- 6.2 k-means algorithm is simple and straightforward. However, it is not as good as density or hierarchical based clustering methods considering the performance. False
  - The decision making of what algorithm to use is based on the data.

Thanks for your attention