INFS 4203 / 7203 Data Mining

Tutorial 7: Answers for Mid-term Exam

	P1	P2	P3	P4	P5
P1	0.0	2.3	4.5	9.1	3.0
P2	2.3	0.0	2.2	7.0	1.4
P3	4.5	2.2	0.0	5.1	2.5
P4	9.1	7.0	5.1	0.0	6.1
P5	3.0	1.4	2.5	6.1	0.0

→

	P1	P2, P5	P3	P4
P1	0.0	3.0	4.5	9.1
P2, P5	3.0	0.0	2.5	7.0
P3	4.5	2.5	0.0	5.1
P4	9.1	7.0	5.1	0.0

1.1

- We have seven two-dimensional data points, whose positions are summarized below:

	x	y
P1	9	9
P2	1	8
P3	5	2
P4	3	10
P5	10	8
P6	2	9
P7	6	1

If using the Chebyshev distance

$$dist\left(x_i,\ x_j\right) = \max_{u=\{1,\ 2,\ \ldots,\ d\}} \left| x_{iu} - x_{ju} \right|$$

, setting k = 3, the centroids at iteration *t* are (2, 5), (5, 6) and (7, 4). Use the following table to illustrate the next iteration of running K-means on the given data set and centroids (rounded to 1 decimal place). [5’]

1.1

		(2,5)	(5,6)	(7,4)
P1	(9,9)	7	4	5
P2	(1,8)	3	4	6
P3	(5,2)	3	4	2
P4	(3,10)	5	4	6
P5	(10,8)	8	5	4
P6	(2,9)	4	3	5
P7	(6,1)	4	5	3

Calculate the distance between Pi and three centroids.

Assign Pi to the centroid with smallest distance.

Last Iteration Centroids	Cluster Assignments	Updated Centroids
(2, 5)	P2[0.5’]	
(5, 6)	P1, P4, P6[1/3’ each]	
(7, 4)	P3, P5, P7[1/3’ each]	

1.1

		(2,5)	(5,6)	(7,4)
P1	(9,9)	7	4	5
P2	(1,8)	3	4	6
P3	(5,2)	3	4	2
P4	(3,10)	5	4	6
P5	(10,8)	8	5	4
P6	(2,9)	4	3	5
P7	(6,1)	4	5	3

Calculate new centroids based on assignments in step 2.

e.g.

$x_{c2}=(9+3+2)/3 = 14/3 \approx 4.7$

$y_{c2}=(9+10+9)/3 = 28/3 \approx 9.3$

C2=(4.7, 9.3)

Last Iteration Centroids	Cluster Assignments	Updated Centroids
(2, 5)	P2	(1.0, 8.0)[0.5’]
(5, 6)	P1, P4, P6	(4.6/4.7, 9.3/9.4)[1’]
(7, 4)	P3, P5, P7	(7.0, 3.6/3.7)[1’]

1.2

Discuss whether the K-means algorithm converges at the end of Iteration (t+1) and point out the reason why it converges or not converges? [5’]

Not converged [2’]
Condition for convergence for this special case [3’]
(General condition for convergence [1’ if special condition is not given])

Example1: Not converged, because the last iteration centroids and the updated centroids are different. [5’]
Example2: Not converged, because the cluster assignment will change in the next iteration. [5’]
Example3: Not converged, because K-means algorithm stops when centroids remain unchanged. [3’]

1.3

Calculate the SSE ($SSE = \sum_{i=1}^{k} \sum_{x \in C_i} dist(x, c_i)^2$) with **the cluster assignments and the updated centroids at iteration (*t*+1),** using the **Chebyshev distance** (rounded to **1 decimal place**). Note that the calculation procedure needs to be shown explicitly; no mark will be given with only a final result. [5']

Distance between Pi and its centroid.

	P1 (9, 9)	P2 (1, 8)	P3 (5, 2)	P4 (3, 10)	P5 (10, 8)	P6 (2, 9)	P7 (6, 1)
c1 (1.0, 8.0)		0					
c2 (4.6/4.7, 9.3/9.4)	4.3/4.4			1.6/1.7		2.6/2.7	
c3 (7.0, 3.6/3.7)			2		4.3/4.4		2.6/2.7

$$SSE = dist(P1,c2)^2 + dist(P2,c1)^2 + dist(P3,c3)^2 + dist(P4,c2)^2 + dist(P5,c3)^2 + dist(P6,c2)^2 + dist(P7,c3)^2$$
$$= 4.3^2 or 4.4^2 + 0^2 + 2^2 + 1.7^2 or 1.6^2 + 4.3^2 or 4.4^2 + 2.7^2 or 2.6^2 + 2.7^2 or 2.6^2$$ [any one of the first two steps 4']
$$= 58.4 \ or \ 58.5 \ or \ 58.7 \ or \ 58.8 \ or \ 58.9$$ [1']

(Calculate mean SSE [2.5'])

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

2.1

Compute the **support count** for itemsets {Melon}, {Banana, Strawberry}, and {Banana, Melon, Strawberry} by treating each transaction ID as a market basket. [5’]

Transaction ID	Items bought
010	Apple, Strawberry, Melon
022	Apple, Banana, Orange, Melon
025	Apple, Banana, Strawberry, Melon
031	Apple, Orange, Strawberry, Melon
033	Banana, Orange, Melon
037	Banana, Strawberry, Melon
040	Orange, Strawberry
059	Apple, Banana, Orange
060	Apple, Strawberry, Melon
071	Apple, Banana, Melon

Support count for {Melon}: 8 [1’]

Support count for {Banana, Strawberry}: 2 [2’]

Support count for {Banana, Melon, Strawberry}: 2 [2’]

2.2

Compute the confidence for the rule {Banana, Strawberry} -> {Melon} and {Melon} -> {Banana, Strawberry}. [5’]

confidence for {Banana, Strawberry}->{Melon}:

#{Banana, Strawberry, Melon}/#{Banana, Strawberry} = 2/2 [1.5’] = 1 [1’]

confidence for {Melon}->{Banana, Strawberry }:

#{Banana, Strawberry, Melon}/#{Melon} = 2/8 [1.5’] = 0.25/0.2/0.3 [1’]

2.3

List all candidate 4-itemsets in the following two tables obtained after the **candidate generation procedure** and the **candidate pruning procedure** in the Apriori algorithm with Join operation (**by Lexicographic order**) and Prune operation, respectively. [15’]

{Apple, Banana, Orange}	{Apple, Banana, Strawberry}
{Apple, Orange, Melon}	{Banana, Orange, Strawberry}
{Apple, Banana, Melon}	{Apple, Orange, Strawberry}
{Banana, Orange, Melon}	{Orange, Strawberry, Melon}

- The Apriori Algorithm
  - Let k=1
  - Generate F_1 = {frequent 1-itemsets}
  - Repeat until F_k is empty
    - **Candidate Generation**: Generate L_{k+1} from by Join operation on F_k
    - **Candidate Pruning**: Prune candidate itemsets in L_{k+1} containing subsets of length k that are infrequent
    - **Support Counting**: Count the support of each candidate in L_{k+1} by scanning the DB
    - **Candidate Elimination**: Eliminate candidates in L_{k+1} that are infrequent, leaving only those that are frequent => F_{k+1}

2.3

The candidate 4-itemsets after the candidate generation procedure:

F_3
{Apple, Banana, Orange}
{Apple, Orange, Melon}
{Apple, Banana, Melon}
{Banana, Orange, Melon}
{Apple, Banana, Strawberry}
{Banana, Orange, Strawberry}
{Apple, Orange, Strawberry}
{Orange, Strawberry, Melon}

→

F_3
{Apple, Banana, Melon}
{Apple, Banana, Orange}
{Apple, Banana, Strawberry}
{Apple, Melon, Orange}
{Apple, Orange, Strawberry}
{Banana, Melon, Orange}
{Banana, Orange, Strawberry}
{Orange, Strawberry, Melon}

→

L_4
{Apple, Banana, Melon, Orange} [3’]
{Apple, Banana, Melon, Strawberry} [3’]
{Apple, Banana, Orange, Strawberry} [3’]

[-1’ for each additional set]

☑ Candidate generation by “join”

- Sorting the items in each frequent *k*-itemset
  - –Lexicographic order
- If two frequent *k*-itemsets
  - –share the first (*k*-1) items
  - –different in the last item
- Concatenate the first (*k*-1) items and the last two items
  - –A candidate (*k*+1)-itemset is generated.

2.3

The candidate 4-itemsets after the candidate pruning procedure:

F_3
{Apple, Banana, Melon}
{Apple, Banana, Orange}
{Apple, Banana, Strawberry}
{Apple, Melon, Orange}
{Apple, Orange, Strawberry}
{Banana, Melon, Orange}
{Banana, Orange, Strawberry}
{Orange, Strawberry, Melon}

→

L_4	3-subsets	F_4
{Apple, Banana, Melon, Orange}	{Banana Melon Orange}	{Apple, Banana, Melon, Orange} [3']
	{Apple Melon Orange}	{Apple, Banana, Orange, Strawberry} [3']
	{Apple Banana Orange}	
	{Apple Banana Melon}	
~~{Apple, Banana, Melon, Strawberry}~~	{Banana Melon Strawberry}	$\notin F_3$
	{Apple Melon Strawberry}	$\notin F_3$
	{Apple Banana Strawberry}	
	{Apple Banana Melon}	
{Apple, Banana, Orange, Strawberry}	{Banana Orange Strawberry}	
	{Apple Orange Strawberry}	
	{Apple Banana Strawberry}	
	{Apple Banana Orange}	

[2' for 1 correct answer with additional sets]
[3-5' for 2 correct answers with additional sets]

☑ Any subset of a frequent itemset must be frequent.

All Frequent 3-itemsets are listed in F_3.

→ If any 3-subset of a 4-itemset is not in F_3, then this 4-itemset is infrequent.

3.1

We have the following distance matrix between five points:

	P1	P2	P3	P4	P5
P1	0.0	2.3	4.5	9.1	**3.0**
P2	2.3	0.0	**2.2**	7.0	1.4
P3	4.5	2.2	0.0	5.1	**2.5**
P4	9.1	7.0	5.1	0.0	**6.1**
P5	3.0	1.4	**2.5**	6.1	0.0

P2<P5<P3<P4

P5<P3<P1<P4

P2<P5<P1<P4

P3<P5<P2<P1

P2<P3<P1<P4

When **k = 2**, list all the points within the 2-nearest neighbourhood of P1, P2, …, P5 in the following table. **Note that the point itself is not counted in the 2-nearest neighbourhood**. [5’]

P1	P2, P5 [0.5’ each]
P2	P3, P5 [0.5’ each]
P3	P2, P5 [0.5’ each]
P4	P3, P5 [0.5’ each]
P5	P2, P3 [0.5’ each]

3.2

When **k = 2**, fill the reachability distance matrix where are marked "?" in the following table **rounded to 1 decimal place**. Note that **the direction of the reachability distance** is from the row index to the column index (i.e., for the (i, j)th entry of the matrix on the row of Pi and column of Pj, it represents the reachability distance from Pi to Pj). [5']

$$reachdist_2(P_2 \leftarrow P_1) = max\{dist(P_1, P_2), dist_2(P_2)\}$$
$$= max(2.3, 2.2)$$
$$= 2.3$$

$$reachdist_2(P_3 \leftarrow P_2) = max\{dist(P_2, P_3), dist_2(P_3)\}$$
$$= max(2.2, 2.5)$$
$$= 2.5$$

$$reachdist_2(P_4 \leftarrow P_3) = max\{dist(P_3, P_4), dist_2(P_4)\}$$
$$= max(5.1, 6.1)$$
$$= 6.1$$

$$reachdist_2(P_5 \leftarrow P_3) = max\{dist(P_3, P_5), dist_2(P_5)\}$$
$$= max(2.5, 2.5)$$
$$= 2.5$$

	P1	P2	P3	P4	P5
P1	0.0	2.3	4.5	9.1	3.0
P2	3.0	0.0	2.5	7.0	2.5
P3	4.5	2.2	0.0	6.1	2.5
P4	9.1	7.0	5.1	0.0	6.1
P5	3.0	2.2	2.5	6.1	0.0

[1' each]

$$reachdist_2(P_5 \leftarrow P_2) = max\{dist(P_2, P_5), dist_2(P_5)\}$$
$$= max(1.4, 2.5)$$
$$= 2.5$$

☑ Calculate dist(x,y) and $dist_2(x)$ on original distance matrix rather than reachability distance matrix

3.3

Fill where are marked "?" in the following table by calculating the lrd (local reachability density) arlrd (average relative local reachability density) and LOF of the special points **rounded to 1 decimal place**, and **point out** the one which is most likely to be an "outlier". [10']

$$lrd(P_2,2) = \left(\frac{\sum_{y \in N(P_2,2)} reachdist(y \leftarrow P_2)}{|N(P_2,2)|} \right)^{-1}$$

$$= \left(\frac{2.5 + 2.5}{2} \right)^{-1}$$

$$= 0.4$$

	P1	P2	P3	P4	P5
lrd	0.4	0.4	0.4	0.2	0.4
arlrd	1.0	1.0	1.0	0.5	1.0
LOF	1.0	1.0	1.0	2.0	1.0

[1' each]

$$lrd(P_4,2) = \left(\frac{\sum_{y \in N(P_4,2)} reachdist(P_4 \leftarrow y)}{|N(P_4,2)|} \right)^{-1}$$

$$= \left(\frac{5.1 + 6.1}{2} \right)^{-1}$$

$$\approx 0.2$$

$$arlrd(P_1,2) = \frac{lrd(P_1,2)}{\sum_{y \in N(P_1,2)} lrd(y,2)/|N(P_1,2)|}$$

$$= \frac{0.4}{(0.4 + 0.4)/2}$$

$$= 1$$

$$arlrd(P_2,2) = \frac{lrd(P_2,2)}{\sum_{y \in N(P_2,2)} lrd(y,2)/|N(P_2,2)|}$$

$$= \frac{0.4}{(0.4 + 0.4)/2}$$

$$= 1$$

$$arlrd(P_4,2) = \frac{lrd(P_4,2)}{\sum_{y \in N(P_4,2)} lrd(y,2)/|N(P_4,2)|}$$

$$= \frac{0.2}{(0.4 + 0.4)/2}$$

$$= 0.5$$

$$LOF(P_2) = 1/arlrd(P_2,2) = 1/1 = 1$$

$$LOF(P_4) = 1/arlrd(P_4,2) = 1/0.5 = 2$$

Which point is most likely to be an outlier: P4 [3']

(Wrong result but mentioned the highest LOF [1'])

4.1

Complete the **step-by-step** results by the AGNES algorithm using **complete linkage** after Step 1 until finally {P1, P2, P3, P4, P5} is reached: [10’]

	P1	P2	P3	P4	P5
P1	0.0	2.3	4.5	9.1	3.0
P2	2.3	0.0	2.2	7.0	1.4
P3	4.5	2.2	0.0	5.1	2.5
P4	9.1	7.0	5.1	0.0	6.1
P5	3.0	1.4	2.5	6.1	0.0

→

	P1	P2, P5	P3	P4
P1	0.0	3.0	4.5	9.1
P2, P5	3.0	0.0	2.5	7.0
P3	4.5	2.5	0.0	5.1
P4	9.1	7.0	5.1	0.0

→

	P1	P2, P5, P3	P4
P1	0.0	4.5	9.1
P2, P5, P3	4.5	0.0	7.0
P4	9.1	7.0	0.0

table 1

↓

	P1, P2, P3, P5	P4
P1, P2, P3, P5	0.0	9.1
P4	9.1	0.0

table 2

Step 1: P1, {P2, P5}, P3, P4

Step 2: P1, {P2, P5, P3}, P4 / P1, {{P2, P5}, P3}, P4 / table 1 [4’]

Step 3: {P1, P2, P5, P3}, P4 / {P1, {{P2, P5}, P3}}, P4 / table 2 [4’]

Step 4: {P1, P2, P3, P4, P5} / {{P1, {{P2, P5}, P3}}, P4} [2’]

4.2

Based on the answers to 4.1, if we want 3 clusters, give the clustering result: [5’]

P1, {P2, P5, P3}, P4 / P1, {{P2, P5}, P3}, P4 [5’]

(Simply answer ‘step 2’ and the answer for step 2 in Q4.1 is correct [2’])

5.1

We have five numbers {0, 3, 5, 8, 15}, which are assumed to be generated by a univariate gaussian distribution.

What is the mean μ (**rounded to 1 decimal place)** of these five numbers? [2']

[1'] [1']

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i = \frac{0+3+5+8+15}{5} = 6.2$$

5.2

What is the variance σ^2 (**rounded to 1 decimal place)** of these five numbers? [3']

[2'] [1']

$$\sigma^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i-\mu)^2 = \frac{(0-6.2)^2+(3-6.2)^2+(5-6.2)^2+(8-6.2)^2+(15-6.2)^2}{5-1} = 32.7$$

5.3

Which point/points is/are counted as an outlier/outliers given that an outlier is defined as any data point *x* outside of the range $[\mu - 1.5\sigma,\ \mu + 1.5\sigma]$. [5']

A $\frac{0-6.2}{\sqrt{32.7}} \approx 1.08 < 1.5$ $\frac{6.2-3}{\sqrt{32.7}} \approx 0.56 < 1.5$ $\frac{6.2-5}{\sqrt{32.7}} \approx 0.21 < 1.5$

$\frac{8-6.2}{\sqrt{32.7}} \approx 0.31 < 1.5$ $\frac{15-6.2}{\sqrt{32.7}} \approx 1.54 > 1.5$

B $\mu - 1.5\sigma = -2.4$ $\mu + 1.5\sigma = 14.8$

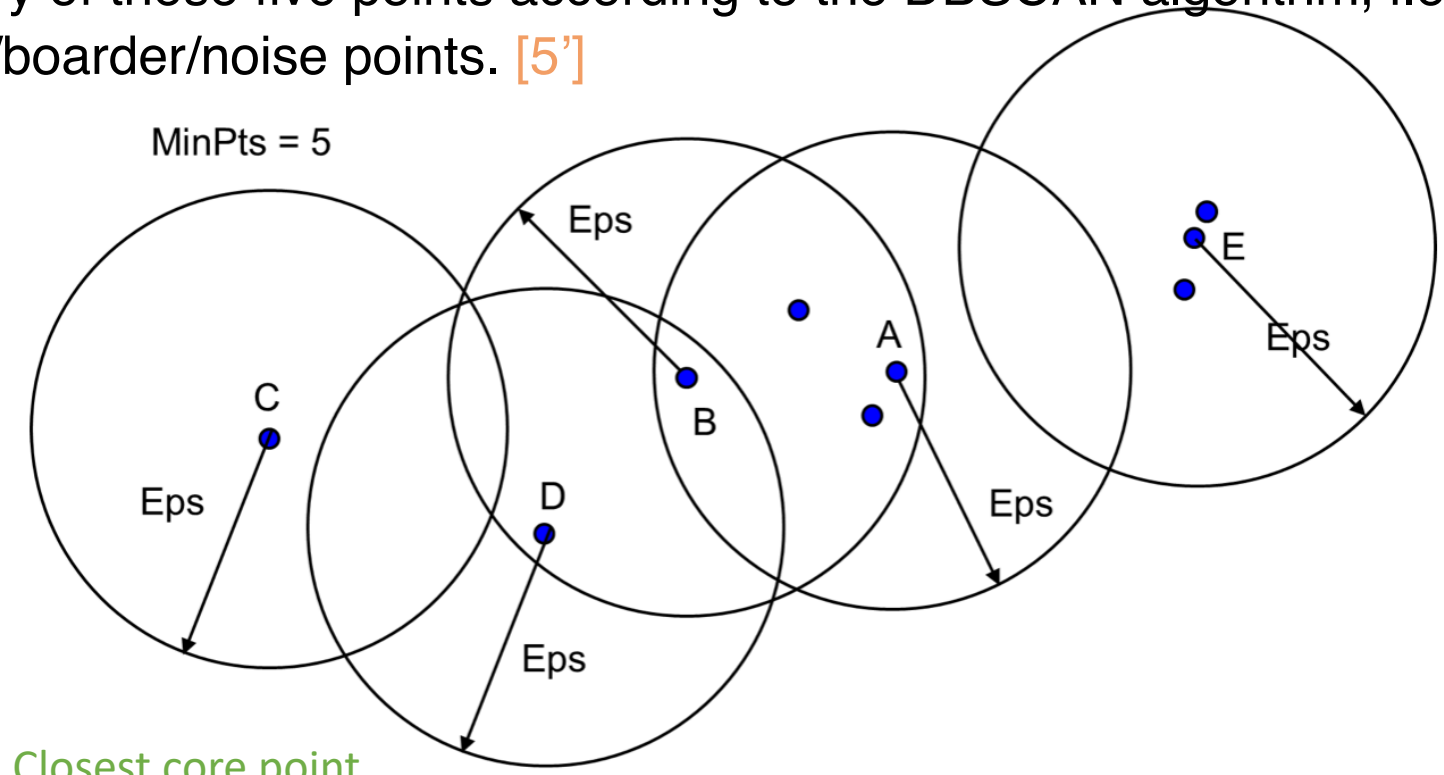
$outlier \in \{x | x < -2.4 \ or \ x > 14.8\}$

[A or B 4.5']

15 is the outlier [0.5']

6

A circle with radius Eps is drawn around each of the five points: A, B, C, D, E. If we set MinPts = 5, discuss the category of these five points according to the DBSCAN algorithm, i.e., which of these points are core/boarder/noise points. [5']



|N(P)|: Number of points within Eps of Point P

|N(A)| = 4 < 5

|N(B)| = 5 -> core point

|N(C)| = 1 < 5

|N(D)| = 2 < 5

|N(E)| = 3 < 5

Closest core point

dist(A, B) < Eps -> boarder point

dist(C, B) > Eps -> noise point

dist(D, B) < Eps -> boarder point

dist(E, B) > Eps -> noise point

Core points: B

Boarder points: D, A [1' each]

Noise points: C, E

7.1

We have three items: A, B, C. For two rules {A, B}→ C and {A}→{B, C}, discuss briefly how their confidences are compared to each other using the Apriori Principle. [5']

[1'] $conf(\{A,B\} \rightarrow \{C\}) = \frac{\#\{A,B,C\}}{\#\{A,B\}}$

[1'] $conf(\{A\} \rightarrow \{B,C\}) = \frac{\#\{A,B,C\}}{\#\{A\}}$

[2'] $\#\{A\} \geq \#\{A,B\}$

$\Rightarrow conf(\{A,B\} \rightarrow \{C\}) \geq conf(\{A\} \rightarrow \{B,C\})$ [1']

- Students tend to calculate the actual confidence of two itemset. However, based on the apriori rule, we don't need to calculate the actual value to compare the confidence of two itemset.

8.1 (Most of students choose to answer Q8.1.)

Discuss briefly the differences between the following three lines of code:

AgglomerativeClustering(n_clusters=3, affinity='euclidean', linkage='complete')

AgglomerativeClustering(n_clusters=2, affinity='l2', linkage='average')

AgglomerativeClustering(n_clusters=3, affinity='manhattan', linkage='single') [5’]

Line1: The number of clusters is 3. Using Euclidean distance, and complete link (max operation) for AGNES [5/3’]

Line2: The number of clusters is 2. Using Euclidean distance, and average link (average operation) for AGNES [5/3’]

Line3: The number of clusters is 3. Using manhattan distance and single link (min operation) for AGNES [5/3’]

Or

These three codes are for AGNES method. They are different in the linkage methods used [5/3’], while code 1 and 3 are the same in the number of clusters [5/3’], and code 1 and 2 are the same in the distance metrics used [5/3’].

☑ ‘l2’ is Manhattan distance rather than Chebyshev distance (‘l_∞’).

8.2 (While Q8.2 is much easier.)

Alice used the following code to generate frequent itemsets:

```
freq_set = apriori(df, min_support=0.4, use_colnames=True)
```

After freq_set is generated, Alice wanted to find the support for {bread, pizza}. However, the itemset {bread, pizza} is not in the current freq_set. Alice is sure that such an itemset exists in at least 20% of the transactions. Could you please help Alice to update the given code, such that {bread, pizza} exists in freq_set? Discuss briefly your solution to help Alice [5']

Q: Why itemset {bread, pizza} is not in the current freq_set when min_support = 0.4?
A: Because sup({bread, pizza}) < 0.4

Q: If sup({bread, pizza}) $\in$ [0.2, 0.4) , how can we make sure {bread, pizza} is in the result of the aprori algorithm?
A: Make sure min_support $\leq$ sup({bread, pizza}) i.e. min_support $\leq$ 0.2.

freq_set = apriori(df, min_support=0.2 (or smaller), use_colnames=True) [5']

Thanks for your attention