**Please read the mid-term exam information on BB first.**

**This mid-term exam contains 100 points distributed across 8 questions, which takes 25% of the final mark. Please answer in the space provided under each question or sub-question by typing. No scan/image file will be accepted even in part of the answers.**

**It needs to be submitted in 75 minutes, and can only be submitted one time to Turnitin. Please use the following text box to specify any assumptions you have made in completing the exam and which questions those assumptions relate to. You may also include queries you may have made with respect to a particular question, should you have been able to ‘raise your hand’ in an examination room.**

Assumptions you have made in completing the exam and which questions those assumptions relate to, as well as queries:

**Question 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 |

**1.1)[5 points]** If using the **Chebyshev distance ()**, 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**).

Iteration *t*+1:

|  |  |  |
| --- | --- | --- |
| Last Iteration Centroids | Cluster Assignments | Updated Centroids |
| (2, 5) | (1,8) | (1.0, 8.0) |
| (5, 6) | (3,10), (2,9), (9,9), | (4.6, 9.3) |
| (7, 4) | (10,8), (6,1), (5,2) | (7.0, 3.6) |

* 1. **[5 points]** 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?

Write your answer here:

It did not converge, since the centroid has been updated after iteration(t+1).

Example1: Not converged, because the last iteration centroids and the updated centroids are different. [5’]

* 1. **[5 points]** Calculate the SSE () with **the cluster assignments and the updated centroids at iteration (*t*+1),** usingthe **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.

SSE = 0 + 0 + (3-4.6)2 +(10-9.3)2 + (2-4.6)2 + (9-9.3)2 + (9-4.6)2 +(9-9.3)2 +(10-7.0)2 +(8-3.6)2+ (6-7.0)2 + (1-3.6)2+(5-7.0)2 + (2-3.6)2 = 58.4

**Question 2**

We have the following data:

|  |  |
| --- | --- |
| 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 |

**2.1)** **[5 points]** Compute the **support count** for itemsets {Melon}, {Banana, Strawberry}, and {Banana, Melon, Strawberry} by treating each transaction ID as a market basket.

Support count for {Melon}: 8

Support count for {Banana, Strawberry}: 2

Support count for {Banana, Melon, Strawberry}: 2

**2.2)** **[5 points]** Compute the confidence for the rule {Banana, Strawberry}->{Melon} and

{Melon}->{Banana, Strawberry } (**please write down your answer in the decimal form rather than the percentage form and 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.

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

2/2 =1.0

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

2/8 =1/4 = 0.3

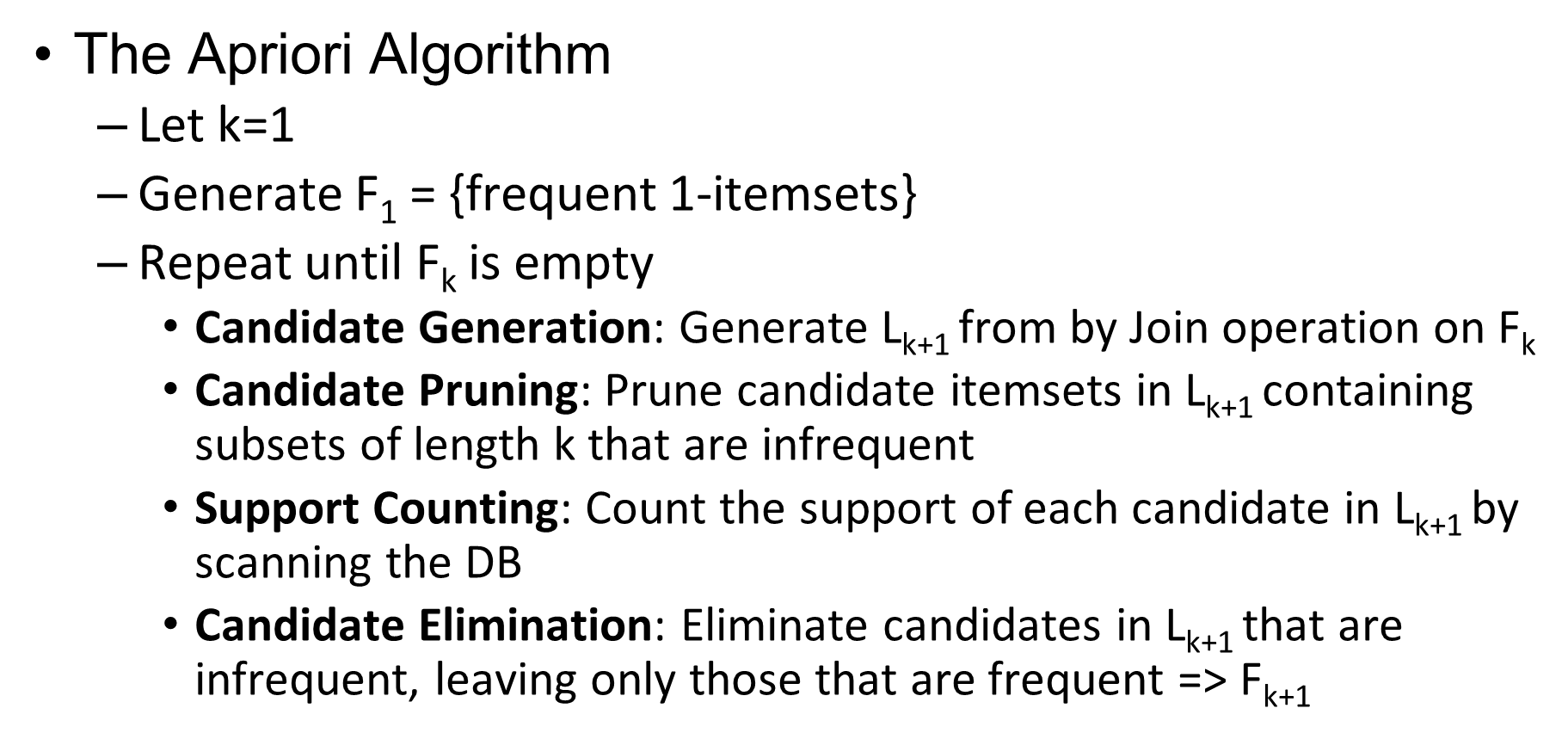
**2.3)** **[15 points]** Consider the following 3-frequent itemsets:

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

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.

Note that the tables are provided to save your time. Depending on your solution, add more rows/columns, or leave part of the tables empty if necessary.

(the general procedure of the Apriori algorithm is shown as a reference:

)

The candidate 4-itemsets after the candidate generation procedure:

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

The candidate 4-itemsets after the candidate pruning procedure:

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

**Question 3**

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 |

**3.1)** **[5 points]** 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**.

|  |  |
| --- | --- |
| P1 | {P2, P5} |
| P2 | {P3, P5} |
| P3 | {P2, P5} |
| P4 | {P3, P5} |
| P5 | {P2, P3} |

**3.2) [5 points]** 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).

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 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 |

**3.3)** **[10 points]** 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”.

, ,

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | P1 | P2 | P3 | P4 | P5 |
|  | 0.4 | 0.4 | 0.4 | 0.2 | 0.4 |
|  | 1.0 | 1.0 | 1.0 | 0.5 | 1.0 |
|  | 1.0 | 1.0 | 1.0 | 2.0 | 1.0 |

Which point is most likely to be an outlier: P4

**Question 4**

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 |

**4.1) [10 points]** 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:

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

Step2: P1, {P2, P3, P5}, P4

Step3: {P1, P2, P3, P5}, P4

Step4: {P1, P2, P3, P4, P5}

**4.2)** **[5 points]** Based on the answers to 4.1, if we want 3 clusters, give the clustering result:

The clustering result containing 3 clusters:

P1, {P2, P3, P5}, P4

**Question 5**

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

**5.1) [2 points]** What is the mean (**rounded to 1 decimal place)** of these five numbers? Note that the calculation procedure needs to be shown explicitly; no mark will be given with only a final result. ()

= 6.2

**5.2) [3 points]** What is the variance (**rounded to 1 decimal place)** of these five numbers?Note that the calculation procedure needs to be shown explicitly; no mark will be given with only a final result. ()

*(0-6.2)2+ (3-6.2)2 + (5-6.2)2 + (8-6.2)2 + (15-6.2)2= 130.8/4 = 32.7*

**5.3) [5 points]** 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 . Note that the calculation procedure needs to be shown explicitly; otherwise no mark will be given with only a final result.

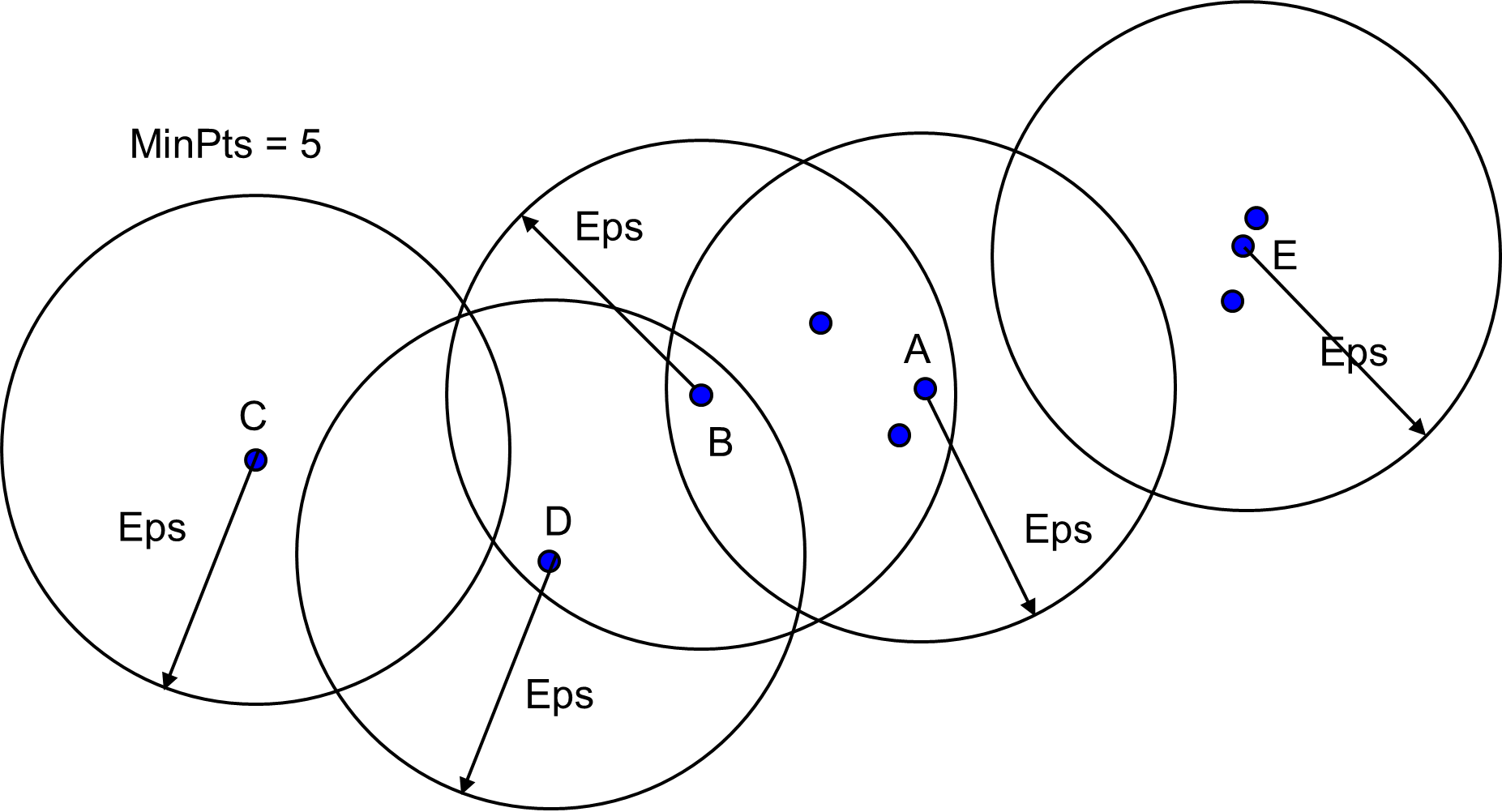
(the answer can contain one point **or** multiple points)

Outlier: 15

[6.2 - 1.5\*, 6.2 + 1.5\*] = [-2.377587073, 14.77758707]

**Question 6 [5 points]**

The following image shows the distribution of nine two-dimensional points:



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.

Core points: B

Boarder points: A, D

Noise points: C, E

**Question 7 [5 points]**

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.

Write your answer here:

The confidence for the first one is #{A,B,C}/#{A,B}

The confidence for the second one is #{A,B,C}/#{A}

Which the denominator is different in two cases. And since {A} is smaller set, which #{A} > #{A,B}

So the confidence of {A}->{B, C} is larger or equal to {A, B}-> C

**Question 8 [5 points]**

Please pick **one** from 8.1) and 8.2) to answer. Write **explicitly** which one you will choose and the question will be **marked according to the chosen sub-question**.

**8.1)** 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')

**8.2)** 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.

Which one will you choose to answer, 8.1) or 8.2)? 8.1)

Write your answer here to the chosen sub-question:

In this AgglomerativeClustering function, three different parameters are specified. In particular, the n\_clusters mean the number of clusters, and the affinity decides Metric used to compute the linkage. Here the first use the Euclidean distance metric to calculate the distance to determine cluster. Second use “l2” distance matrics and third use “manhattan” distance metric. The linkage determines which link standard is used. The linkage criterion determines which distance to use between sets of observation. The first is to use "complete" means to use the maximum distance of each observation set. The second use of "average" means to use the average of the distances of each observation of the two sets. The third use of "single" refers to the use of the smallest distance between all observations in the two sets.