CS 4210 – Assignment #5

Maximum Points: 100 pts.

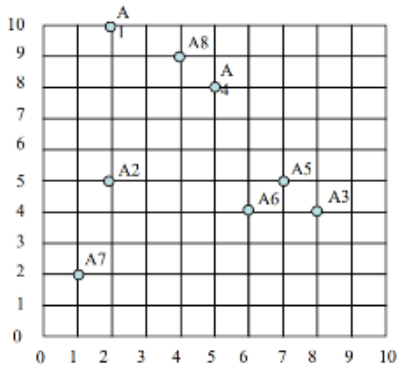
Bronco ID: |0|1|4|8|3|1|5|5|7|

Last Name: Guerra Umana

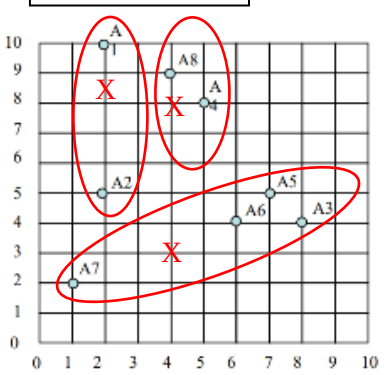
First Name: Raul

Note 1: Your submission header must have the format as shown in the above-enclosed rounded rectangle.
Note 2: Homework is to be done individually. You may discuss the homework problems with your fellow students, but you are NOT allowed to copy – either in part or in whole – anyone else's answers.
Note 3: Your deliverable should be a .pdf file submitted through Gradescope until the deadline. Do not forget to assign a page to each of your answers when making a submission. In addition, source code (.py files) should be added to an online repository (e.g., github) to be downloaded and executed later.
Note 4: All submitted materials must be legible. Figures/diagrams must have good quality.
Note 5: Please use and check the Canvas discussion for further instructions, questions, answers, and hints. The bold words/sentences provide information for a complete or accurate answer.

1. [25 points] By considering the following 8 2D data points below do:
   a. [20 points] Group the points into 3 clusters by using k-means algorithm with Euclidean distance. Show the intermediate clusters (**by drawing ellipses on this 2D space**) and centroids (**by drawing marks like X on this 2D**) in each iteration until convergence. Consider the initial centroids as: C1 = A1, C2 = A4, and C3 = A7.



Example only!



Solution format:

1st iteration								
Centroid: (C1, C2, C3)								
Instance	A1	A2	A3	A4	A5	A6	A7	A8
C1 dist.								
C2 dist.								
C3 dist.								
Cluster Assigned								

2nd iteration centroid: (C1, C2, C3)

   b. [5 points] Calculate the SSE (Sum of Square Errors) of the final clustering.

2. [15 points] Complete the Python program (clustering.py) that will read the file training_data.csv to cluster the data. Your goal is to run k-means multiple times and check which k value maximizes the Silhouette coefficient. You also need to plot the values of k and their corresponding Silhouette coefficients so that we can visualize and confirm the best k value found. Next, you will calculate and print the Homogeneity score (the formula of this evaluation metric is provided in the template) of this best k clustering task by using the testing_data.csv, which is a file that includes ground truth data (classes).

3. [20 points] The dataset below presents the user ratings on a 1-3 scale for 6 different rock bands.

	Bon Jovi	Metallica	Scorpions	AC/DC	Kiss	Guns n' Roses
Fred	1	3	-	3	1	3
Lillian	3	-	2	2	3	1
Cathy	2	2	2	3	-	2
John	3	2	2	2	?	?

   a. [10 points] Apply **user-based** collaborative filtering on the dataset to decide about recommending the bands Kiss and Guns n' Roses to John. You should make a recommendation when the predicted rating is greater than or equal to 2.0. Use cosine similarity, a neutral value (1.5) for missing values, and the top 2 similar neighbors to build your model.
   b. [10 points] Now, apply **item-based** collaborative filtering to make the same decision. Use the same parameters defined before to build your model.

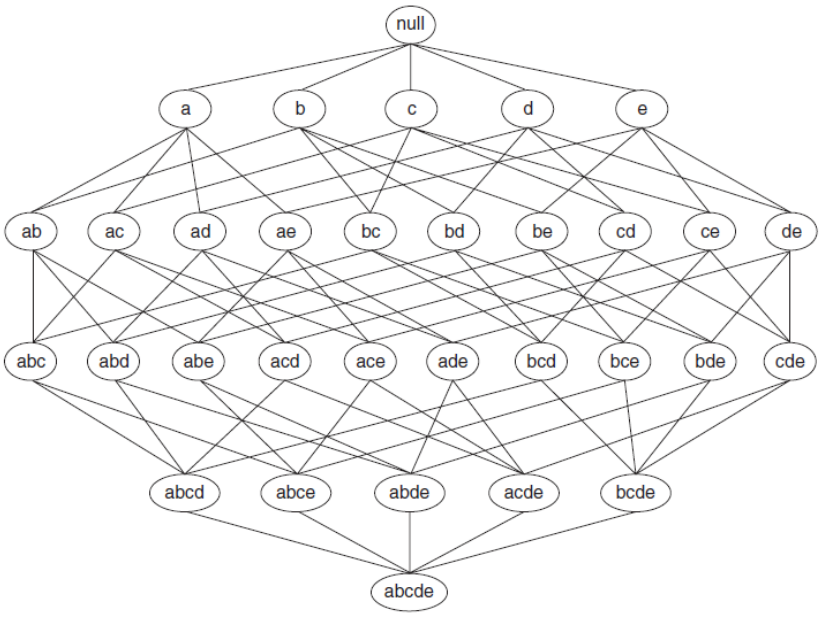
4. [25 points] Consider the following transaction dataset.

Transaction ID	Items Bought
1	$\{a, b, d, e\}$
2	$\{b, c, d\}$
3	$\{a, b, d, e\}$
4	$\{a, c, d, e\}$
5	$\{b, c, d, e\}$
6	$\{b, d, e\}$
7	$\{c, d\}$
8	$\{a, b, c\}$
9	$\{a, d, e\}$
10	$\{b, d\}$

   Suppose that minimum support is set to 30% (*minsup*) and minimum confidence is set to 60%.

   a. [5 points] Rank all frequent itemsets according to their support (list their support values).
   b. [5 points] For all frequent 3-itemsets, rank all association rules - according to their confidence values - which satisfy the requirements on minimum support and minimum confidence (list their confidence values).
   c. [5 points] Show how the 3-itemsets candidates can be generated by the F_{k-1} X F_{k-1} method and if these candidates will be pruned or not.

d. [10 points] Consider the lattice structure given below. Label each node with the following letter(s): *F* if it is frequent and *I* if it is infrequent.



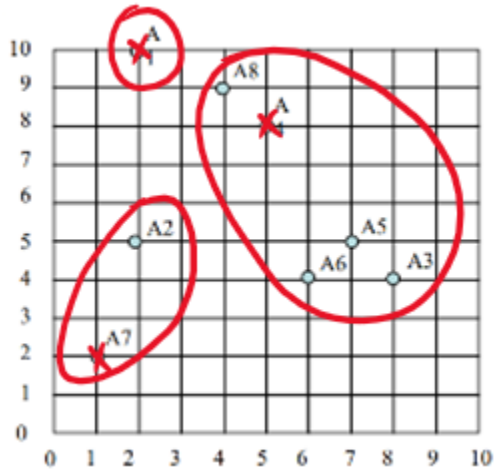
5. [15 points] Complete the Python program (association_rule_mining.py) that will read the file retail_dataset.csv to find strong rules related to supermarket products. You will need to install a python library this time. Just use your terminal to type: pip install mlxtend. Your goal is to output the rules that satisfy *minsup* = 0.2 and *minconf* = 0.6, as well as the priors and probability gains of the rule consequents when conditioned to the antecedents. The formulas for this math are given in the template.

Important Note: Answers to all questions should be written clearly, concisely, and unmistakably delineated. You may resubmit multiple times until the deadline (the last submission will be considered).

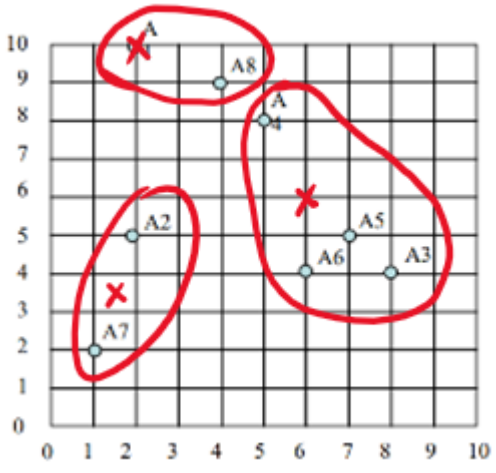
NO LATE ASSIGNMENTS WILL BE ACCEPTED. ALWAYS SUBMIT WHATEVER YOU HAVE COMPLETED FOR PARTIAL CREDIT BEFORE THE DEADLINE!

1. a) All calculations are done in excel. Please see the github link to download the excel file if you would like to see the formulas.

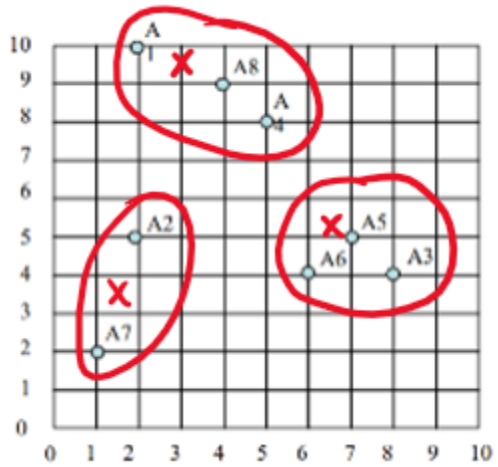
1st iteration								
Centroid: (A1, A4, A7)								
Instance	A1	A2	A3	A4	A5	A6	A7	A8
A1 dist.	0.0	5.0	8.5	3.6	7.1	7.2	8.1	2.2
A4 dist.	3.6	4.2	5.0	0.0	3.6	4.1	7.2	1.4
A7 dist.	8.1	3.2	7.3	7.2	6.7	5.4	0.0	7.6
Cluster Assigned	A1	A7	A4	A4	A4	A4	A7	A4



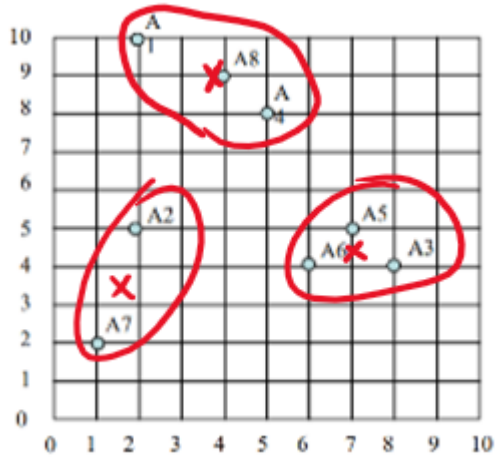
2nd iteration								
Centroid: {(2, 10), (6, 6), (1.5, 3.5)}								
Instance	A1	A2	A3	A4	A5	A6	A7	A8
(2, 10) dist.	0.0	5.0	8.5	3.6	7.1	7.2	8.1	2.2
(6, 6) dist.	5.7	4.1	2.8	2.2	1.4	2.0	6.4	3.6
(1.5, 3.5) dist.	6.5	1.6	6.5	5.7	5.7	4.5	1.6	6.0
Cluster Assigned	(2,10)	(1.5,3.5)	(6,6)	(6,6)	(6,6)	(6,6)	(1.5,3.5)	(2,10)



3rd iteration								
Centroid: {(3, 9.5), (6.5, 5.25), (1.5, 3.5)}								
Instance	A1	A2	A3	A4	A5	A6	A7	A8
(3, 9.5) dist.	1.1	4.6	7.4	2.5	6.0	6.3	7.8	1.1
(6.5, 5.25) dist.	6.5	4.5	2.0	3.1	0.6	1.3	6.4	4.5
(1.5, 3.5) dist.	6.5	1.6	6.5	5.7	5.7	4.5	1.6	6.0
Cluster Assigned	(3, 9.5)	(1.5, 3.5)	(6.5, 5.25)	(3, 9.5)	(6.5, 5.25)	(6.5, 5.25)	(1.5, 3.5)	(3, 9.5)



4th iteration								
Centroid: {(3.66,9), (7,4.33), (1.5,3.5)}								
Instance	A1	A2	A3	A4	A5	A6	A7	A8
(3.66, 9) dist.	1.9	4.3	6.6	1.7	5.2	5.5	7.5	0.3
(7, 4.33) dist.	7.6	5.0	1.1	4.2	0.7	1.1	6.4	5.5
(1.5, 3.5) dist.	6.5	1.6	6.5	5.7	5.7	4.5	1.6	6.0
Cluster Assigned	(3.66,9)	(1.5,3.5)	(7,4.33)	(3.66,9)	(7,4.33)	(7,4.33)	(1.5,3.5)	(3.66,9)



b) All calculations are done in excel. Please see the github link to download the excel file if you would like to see the formulas.

	x	y	cluster-x	cluster-y	Distance from nearest centroid	Squared Distance
A1	2	10	3.67	9	1.94	3.78
A2	2	5	1.5	3.5	1.58	2.50
A3	8	4	7	4.33	1.05	1.11
A4	5	8	3.67	9	1.67	2.78
A5	7	5	7	4.33	0.67	0.44
A6	6	4	7	4.33	1.05	1.11
A7	1	2	1.5	3.5	1.58	2.50
A8	4	9	3.67	9	0.33	0.11
					SSE	**14.33**

2. Github: https://github.com/RaulGuerra/CS4210---Machine-Learning.git

3. A) All calculations are done in excel. Please see the github link to download the excel file if you would like to see the formulas.

	Bon Jovi	Metallica	Scorpions	AC/DC	Kiss	Guns n' Roses
Fred	1	3	1.5	3	1	3
Lillian	3	1.5	2	2	3	1
Cathy	2	2	2	3	1.5	2
John	3	2	2	2	2.33	1.58

Calculations:

Using:

$$Cosine\ Similarity: Sim(u_i, u_k) = \frac{r_i \cdot r_k}{|r_i||r_k|} = \frac{\sum_{j=1}^{m} r_{ij} r_{kj}}{\sqrt{\sum_{j=1}^{m} r_{ij}^2 \sum_{j=1}^{m} r_{kj}^2}}$$

John×Fred	3	6	3	6	sum	18
Fred^2	1	9	2.25	9	sqrt(sum(Fred)^2 * sum(John)^2)	21.12463
John^2	9	4	4	4	Sim()	0.852086

John×Lillian	9	3	4	4	sum	20
Lillian^2	9	2.25	4	4	sqrt(sum(Lillian)^2 * sum(John)^2)	20.10597
John^2	9	4	4	4	Sim()	0.994729

John×Cathy	6	4	4	6	sum	20
Cathy^2	4	4	4	9	sqrt(sum(Cathy)^2 * sum(John)^2)	21
John^2	9	4	4	4	Sim()	0.952381

Using:

$$r_{ij=}\bar{r}_i + \frac{\sum_k Similarities(u_i, u_k)(r_{kj} - \bar{r}_k)}{\sum_k |Similarities(u_i, u_k)|}$$

kiss	2.33

John×Fred	3	6	3	6	2.33	sum	20.33017
Fred^2	1	9	2.25	9	1	sqrt(sum(Fred)^2 * sum(John)^2)	24.24996
John^2	9	4	4	4	5.43	Sim()	0.838359

John×Lillian	9	3	4	4	6.99	sum	26.99051
Lillian^2	9	2.25	4	4	9	sqrt(sum(Lillian)^2 * sum(John)^2)	27.3247
John^2	9	4	4	4	5.43	Sim()	0.98777

John×Cathy	6	4	4	6	3.50	sum	23.49526
Cathy^2	4	4	4	9	2.25	sqrt(sum(Cathy)^2 * sum(John)^2)	24.78892
John^2	9	4	4	4	5.43	Sim()	0.947813

GnR	1.58

Please see Excel file for all calculations. This can be found in the GitHub link provided.

b) All calculations are done in excel. Please see the github link to download the excel file if you would like to see the formulas.

	Fred	Lillian	Cathy	John
Bon Jovi	1	3	2	3
Metallica	3	1.5	2	2
Scorpions	1.5	2	2	2
AC/DC	3	2	3	2
Kiss	1	3	1.5	3.07
Guns n' Roses	3	1	2	1.59

Calculations:

Using cosine similarity and:

$$r_{ij=}\bar{r}_j + \frac{\sum_k \boldsymbol{Similarities}(u_j, u_k)(r_{ki} - \bar{r}_k)}{\sum_k |\boldsymbol{Similarities}(u_j, u_k)|}$$

Kiss×Bon Jovi	1	9	3	Sum	13
Bon Jovi ^2	1	9	4	sqrt(sum(BJ)^2 * sum(Kiss)^2)	13.10
Kiss^2	1	9	2.25	Sim()	0.99

Kiss×Metallica	3	4.5	3	Sum	10.5
Metallica^2	9	2.25	4	sqrt(sum(Metallica)^2 * sum(Kiss)^2)	13.67
Kiss^2	1	9	2.25	Sim()	0.77

Kiss×Scorpions	1.5	6	3	Sum	10.5
Scorpions^2	2.25	4	4	sqrt(sum(Scorpions)^2 * sum(Kiss)^2)	11.21
Kiss^2	1	9	2.25	Sim()	0.94

Kiss×AC/DC	3	6	4.5	Sum	13.5
AC/DC^2	9	4	9	sqrt(sum(AC/DC)^2 * sum(Kiss)^2)	16.42
Kiss^2	1	9	2.25	Sim()	0.82

Kiss	3.07

GnR×Bon Jovi	3	3	4	Sum	10
Bon Jovi ^2	1	9	4	sqrt(sum(BJ)^2 * sum(GnR)^2)	14.00
GnR^2	9	1	4	Sim()	0.71

GnR×Metallica	9	1.5	4	Sum	14.5
Metallica^2	9	2.25	4	sqrt(sum(Metallica)^2 * sum(GnR)^2)	14.61
GnR^2	9	1	4	Sim()	0.99

GnR×Scorpions	4.5	2	4	Sum	10.5
Scorpions^2	2.25	4	4	sqrt(sum(Scorpions)^2 * sum(GnR)^2)	11.98
GnR^2	9	1	4	Sim()	0.88

GnR×AC/DC	9	2	6	Sum	17
AC/DC^2	9	4	9	sqrt(sum(AC/DC)^2 * sum(GnR)^2)	17.55
GnR^2	9	1	4	Sim()	0.97

GnR×Kiss	3	3	3	Sum	9
Kiss^2	1	9	2.25	sqrt(sum(kiss)^2 * sum(GnR)^2)	13.10
GnR^2	9	1	4	Sim()	0.69

GnR	1.59

4.

a) All calculations are done in excel. Please see the github link to download the excel file if you would like to see the formulas.

1-itemsets		
set	count	support
a	5	0.5
b	7	0.7
c	6	0.6
d	9	0.9
e	6	0.6

2-itemsets		
set	count	support
ab	3	0.3
ac	2	0.2
ad	4	0.4
ae	4	0.4
bc	3	0.3
bd	6	0.6
be	4	0.4
cd	4	0.4
ce	2	0.2
de	6	0.6

3-itemsets		
set	count	support
abc	1	0.1
abd	2	0.2
abe	2	0.2
acd	1	0.1
ace	1	0.1
ade	4	0.4
bcd	2	0.2
bce	1	0.1
bde	4	0.4
cde	2	0.2

4-itemsets		
set	count	support
abcd	0	0
abce	0	0
abde	2	0.2
acde	1	0.1
bcde	1	0.1

b)

ade -Rule	Confidence
a→de	0.8
ad→e	1
ae→d	1
d→ae	0.444444
de→a	0.666667
e→ad	0.666667

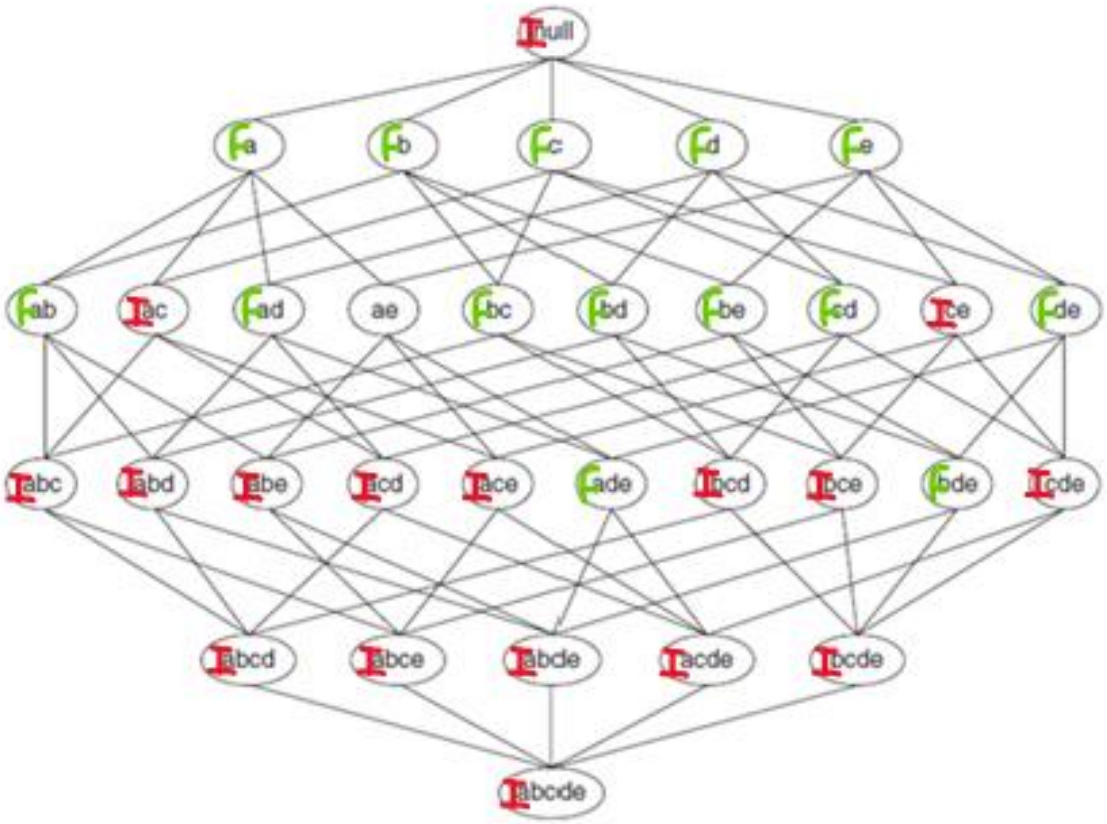
bde -Rule	Confidence
b→de	0.571429
bd→e	0.666667
be→d	1
d→be	0.444444
de→b	0.666667
e→bd	0.666667

c)

2-itemsets		
set	count	support
ab	3	0.3
ac	2	0.2
ad	4	0.4
ae	4	0.4
bc	3	0.3
bd	6	0.6
be	4	0.4
cd	4	0.4
ce	2	0.2
de	6	0.6

abd	infrequent
abe	infrequent
ade	pass
bcd	infrequent
bce	infrequent
bde	pass

d)



5. Github: https://github.com/RaulGuerra/CS4210---Machine-Learning.git