Student name: Davis, Caleb Student ID: S5175471

Family name, Given name

School of Information and Communication Technology (ICT)

2030ICT Intro to Big Data Analytics

Trimester 2, 2020

Version A

Writing time

2 hours

Reading time

10 minutes

Materials permitted

Yes

Question 1 (7 points)

A company has collected quarterly sales for the past two years which are shown in the following table. The company wants to forecast the next year's seasonal sales.

Index	Time	Sales ($)	Index	Time	Sales ($)
1	Spring 2017	4836	5	Spring 2018	5412
2	Summer 2017	5890	6	Summer 2018	6138
3	Fall 2017	6510	7	Fall 2018	6666
4	Winter 2017	7564	8	Winter 2018	8184
SUM	**2017**	**24800**	**SUM**	**2018**	**26400**

TOTAL 51200

a) Let A_1 and A_2 be the actual total sales (i.e., the sum of all four seasonal sales) in 2017 and 2018, respectively. Assume current time is t, the n-moving average (MV) technique makes forecast for the time $t+1$ by taking the average of previous n actual values where $n \leq t$. (The formula can be written as $F_{t+1} = \frac{\sum_{i=t-n+1}^{t} A_i}{n}$ where A_i is the i-th actual data). Predict the total sales for 2019 using MV with 2 actual values A_1 and A_2.

25600

b) In general, we would expect the total sales gets increased in both 2018 and 2019 if the economy situation has been keeping going well since 2017. Use this and your answer to sub-question **a)** to explain the limitation of moving average (MV) method in forecasting.

Moving average has trouble forecasting trends as it works better with patterns. In the example you can see that the sales are going up each year, but the moving average predicts that it will come back down

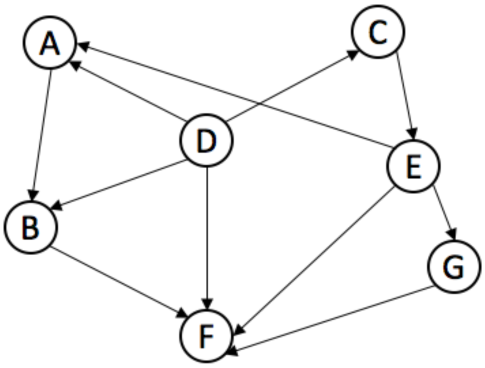
c) Calculate the average seasonal sales for both 2017 and 2018. (see '**d)**')

d)

Quarter	2017	Seasonal Index	2018	Seasonal Index	Average Index	2019
Spring	4836	0.78	5412	0.82	0.8	5520
Summer	5890	0.95	6138	0.93	0.94	6486
Fall	6510	1.05	6666	1.01	1.03	7107
Winter	7564	1.22	8184	1.24	1.23	8487
Average	6200		6600			6900

Question 2 (3 points)

a) Degree centrality is the most basic centrality metric. Find the degree centrality of each vertex in the graph and rank the vertices using their degree centrality. Fill out the following table. (**Note**, you can copy the form to your answer sheet then fill blank cells.)



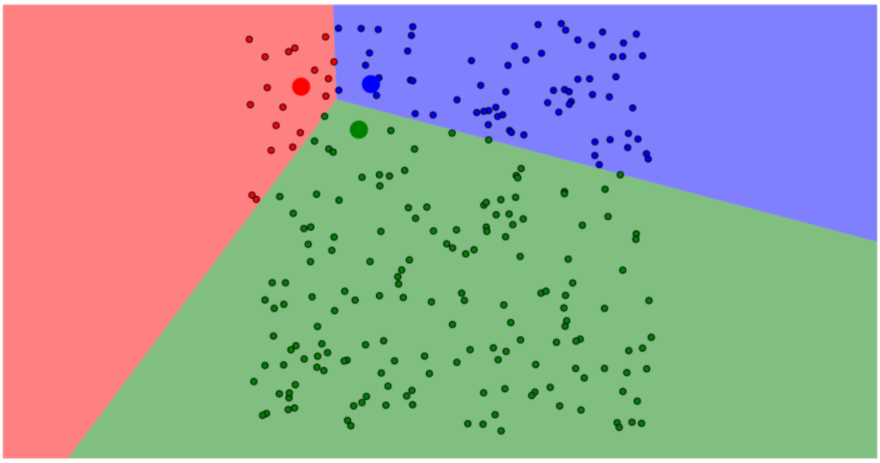
Vertex	Centrality	Ranking
A	3	5
B	3	4
C	2	7
D	4	2
E	4	3
F	4	1
G	2	6

Question 3 (5 points)

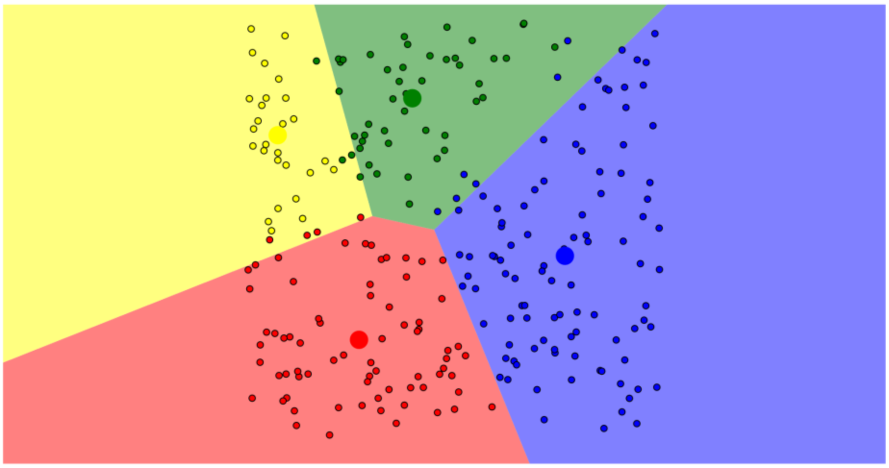
Answer the following questions about K-means clustering.

1) The following picture shows a status of running K-means clustering algorithm on sample data points. What is the value of K here? Does current status give the final clustering result? If yes, give the reason. If no, what should be done in the next step?

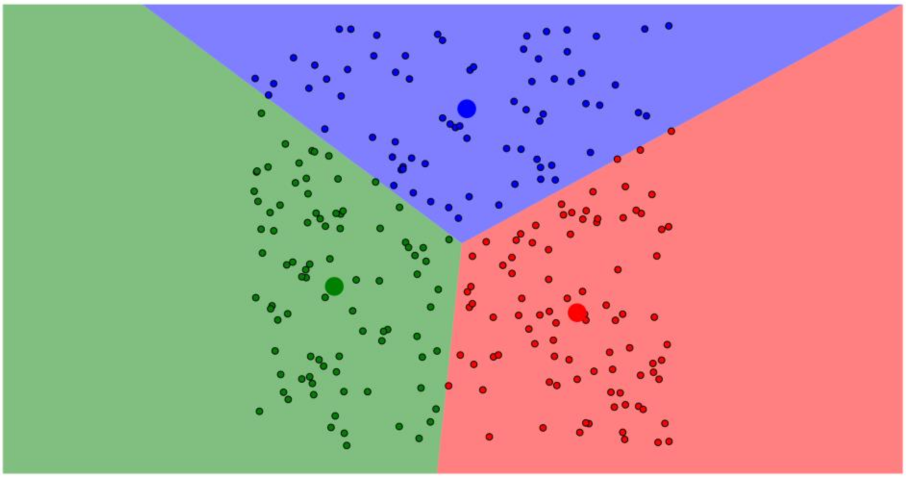
   K = 3 No this is not the final clustering as it now needs to find the center of those clusters and retest



2) The following picture shows a status of running K-means clustering algorithm on sample data points. What is the next step of the algorithm?



3) Is the following a convergence status of the K-means clustering? Explain why.



This appears to be a convergence state as the cluster centres are central to the clusters but without running it again, it cannot be certain. If the next time this is run the centres of the clusters do not move, this is the convergence status.

4) Is K-means clustering used for supervised machine learning? Briefly explain what supervised machine learning is.

Supervised machine learning is when the Ai is trained by handing it data with the answers attached K-means clustering could definitely be used to label the data with the ‘answer’ however I am unsure of if it is actually used professionally.

Question 4 (5 points)

Schema reuse is a new trend in creating database schemas by allowing users to copy and adapt existing ones. The motivation behind schema reuse is the slight differences between schemas in the same domain; thus, making schema design more efficient. Reusing existing schemas supports reducing not only the effort of creating a new schema but also the heterogeneity between schemas.

Finding related schemas is one of the core problems of schema reuse. You work as a data engineer at Oracle. Oracle has a large repository of schemas. Each database schema has a set of attributes. Some attributes are common among schemas, while others are not. Your task is to support database designers to create new schemas via the schema reuse paradigm. For instance, when a database designer wants to create a new schema, he wants to query the schema repository for references:

- He can start with a few attributes and query the schema repository for hints to finish his design.
- Alternatively, he can complete a schema and query the schema repository to check his design.

Example: we have a repository of schemas:

- *S1: {a1, a3, a7}*
- *S2: {a1, a4, a8}*
- *S3: {a2, a6, a9}*
- *S4: {a1, a5, a10}*

Given a query Q = {a1, a2}, we should rank these schemas as S3 > S1=S2=S4. S3 has the highest rank since attribute a1 occurs frequently in many schemas and thus has less discriminatory power (i.e. the more schemas contain an attribute, less information it provides). Design an algorithm to find related schemas ranked by their similarity to the query.

a) How do you model the problem (input, output, etc.)? Justify your model.

```
for i in range(len(input)):
        Weight {input[i]: 1/NoTimesInCorpus}
Ratings = dict{}
Def addtoratings(Doc, x):
        if Doc not in Ratings:
                Ratings{Doc: Weight{x}}
        else:
                curr = Ratings{Doc}
                Ratings{Doc: curr + Weight{x}}

for each x in input:
        for each Doc in Corpus:
                if x in Doc:
                        addtoratings(Doc, x)
Ratings.sort()
```

What steps should be involved? Provide a quantitative measure for each step if needed. Justify the design choice for each step.

1. Each part of the input would be checked against the corpus to find its weight
2. The schema database would be searched for matches.
3. The results would be ordered using the sum of the weights of the matching parts

b) Apply your approach to the above example and calculate the quantitative results.

S3 = 1, S1 = .3, S2 = .3, S4 = .3

Question 5 (5 points)

You work as a software engineer for Netflix, a popular movie streaming service. Customer feedback suggests that they want Netflix to provide them with recommendations. Netflix has collected lots of data. You were working on the team that decides if two movies are similar to each other. To do this for two movies A and B, you computed the Pearson correlation coefficient between the ratings given to show A, and the ratings given to show B.

Your colleague Thomas works on another team that builds a social network for Netflix. She proposes to use a similar strategy to determine if users are similar to each other, based on whether other users have followed them or not. Whenever a user u follows another user f, Thomas gives the (user, user) pair a "rating" of r(u,f) = 1. Currently, since there are (330 million choose 2) pairs of users, Thomas does not store any ratings for pairs of users who don't follow each other. Thomas again proposes to compute the Pearson correlation coefficient between two sets of ratings.

Is Thomas's strategy a good strategy? Why or why not?

No
This will be highly resource draining task and relies on people following other people on Netflix for their similarities to be calculated. To get a better result you would want to check people against people they do not know however this will be even more draining as it will require a massive database

Question 6 (5 points)

For the following problem describe how you would solve it using MapReduce. The input is a list of documents (ID, text). The output should be the count of each word over all documents. You are given only two machines.

ID	Text
1	*Peter Piper picked a peck of pickled peppers*
2	*A peck of pickled peppers Peter Piper picked*
3	*If Peter Piper picked a peck of pickled peppers*
4	*Where's the peck of pickled peppers Peter Piper picked*

1) You should explain how the input is mapped into (key, value) pairs by the map stage, i.e., specify what is the key and what is the associated value in each pair, and, if needed, how the key(s) and value(s) are computed. Then you should explain how the (key, value) pairs produced by the map stage are processed by the reduce stage to get the final answer(s). If the job cannot be done in a single MapReduce pass, describe how it would be structured into two or more map - reduce jobs with the output of the first job becoming input to the next one(s).

   With 2 machines I would use one for Sorting but both for mapping and reducing. If machine A is the Sorting Machine less documents should be sent to Machine A and the count of documents should be sent to machine A.

   Both Machines
   The document would be counted and turned into a list of words and their counts.
   The Key and value would be {Word: Number of instances}

   Machine A (waits until number of key value pair lists = No. Docs)
   sorts all lists into 1 list with {key: V1,V2,V3,ect}
   Send half of the list back to Machine 2

   Sum the values attached to each key.
   Recombine list with Machine 2's

   Output final {Key: Value} pair list

2) If there are 5 documents, how do you distribute them into two machines? Explain your criteria for this distribution.

   Give most of the docs to Machine 2 based on Wordcount. Keep the wordcount total between the 2 machines as close as possible

END OF EXAM