CS 620–Introduction to Data Science and Analytics, HW4, Spring 2022

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16
D1	2	0	0	0	0	4	0	0	2	1	1	0	0	0	0	0
D2	0	0	0	0	1	1	0	0	7	0	2	0	0	0	0	6
D3	0	0	1	0	0	1	0	0	3	0	3	0	2	0	3	0
D4	0	3	1	0	0	0	0	5	0	0	0	3	0	0	0	0
D5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	3	0
D6	0	0	0	0	1	0	0	0	3	0	0	2	0	0	0	0
D7	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	3
D8	0	0	1	0	0	0	0	3	0	0	0	2	0	1	0	0
D9	0	1	2	0	0	0	2	0	3	0	0	0	2	1	0	1
D10	0	0	0	0	1	1	0	0	0	1	0	5	0	0	0	0
Q1	**1**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**1**

Consider the above Document-Term Matrix for Documents D1-D10, Terms T1-T16, and Query Q1. You can download it here. You have 2 options to solve the given problem (1). Pick only one of these options.

Option 1: Create a program using python and other associated libraries and display the results.

Option 2: Detailed calculations by hand.

1) (50 pts) Consider the given documents and the term-frequencies.
   a. Calculate the tf.idf weights for each term. Note: Don't forget to normalize your raw term-frequencies (tf). Use base 2 for log scale ($idf_t = \log_2(N/df_t)$)
   b. Transform the query into the vector space using the same document-frequency (df) values in the above table and calculate the tf.idf weights for the query. (Note: DO NOT normalize the terms of this query when considering the tf values)
   c. Based on the document vectors calculated, rank each document for the given query using cosine similarity.

2) (50 pts) Consider the transaction database in the table below. Show the candidate itemsets and the frequent itemsets in each level-wise pass of the Apriori algorithm at minimum support count of 2.

TID	List of Items
T100	I1, I2, I5
T100	I2, I4
T100	I2, I3
T100	I1, I2, I4
T100	I1, I3
T100	I2, I3
T100	I1, I3
T100	I1, I2 ,I3, I5
T100	I1, I2, I3

What to turn in: Submit your **LastName-hw4.zip (your LastName-hw4.py + Lastname-hw4.pdf)** or **Lastname-hw4.pdf** to Blackboard.