Name of the student:	Tanmay Prashant Rane	**Roll No.**	8031
Assignment Number:	2	**Date of Assignment:**	
Relevant CO's ITC801.4	**At the end of the course students will be able to apply Big data analytics in real life applications.**		
Sign here to indicate that you have read all the relevant material provided before attempting this assignment		**Sign:**	

Assignment grading using Rubrics

Indicator	**Very Poor**	**Poor**	**Average**	**Good**	**Excellent**
Timeline (2)	More than a session late (0)	NA (0.5)	NA (1)	NA	Early or on time (2)
Organization of Content (2)	N/A	No sense of organization, Paragraphs lack clear ideas (0.5)	Some paragraphs have clear ideas, support from examples may be missing and transitions are weak (1)	Most paragraphs have clear ideas, are supported with some examples and have transitions. (1.5)	All paragraphs have clear ideas, are supported with examples and have smooth transitions. (2)
Level of content (4)	N/A	Major points omitted or addressed minimally(1)	Content is sound and solid; ideas are present but not particularly developed or supported; some evidence, but usually of a generalized nature.(2)	Well-presented and argued; ideas are detailed, developed and supported with evidence and details, mostly specific. (3)	Exceptionally well-presented and argued; ideas are detailed, well-developed, supported with specific evidence & facts, as well as examples and specific details. (4)
Grammar and Mechanics (2)	N/A	Spelling, punctuation and grammatical errors create distraction, making reading difficult(0.5)	Most spelling, punctuation and grammar correct. Some errors remain(1)	Few spelling, punctuation and grammatical errors allowing reader to follow ideas clearly (1.5)	Assignment is free of distracting spelling, punctuation and grammatical errors(2)

Late submission details (if any)

Reason(s) of late submission	Submission date	Actual submission date	sign of student

Assignment 2
Assignment on Recommendation systems and mining social network graphs

Course title: Big Data Analytics
Course term: 2019-2020
Instructor name: Saurabh Kulkarni

Q.1.

	a	b	c	d	e	f	g	h
A	4	5		5	1		3	2
B		3	4	3	1	2	1	
C	2		1	3		4	5	3

Figure 1: A utility matrix

Fig.1 is a utility matrix representing the ratings, on a 1-5 star scale, of eight items, a through h, by three users A, B, and C. Compute the following from the data of this matrix.

1. **Treating the utility matrix as boolean, compute the Jaccard distance between each pair of users.**
2. **Repeat Part (1), but use the cosine distance.**
3. **Treat ratings of 3, 4, and 5 as 1 and 1, 2, and blank as 0. Compute the Jaccard distance between each pair of users.**
4. **Repeat Part (3), but use the cosine distance.**
5. **Normalize the matrix by subtracting from each nonblank entry the average value for its user.**
6. **Using the normalized matrix from Part (5), compute the cosine distance between each pair of users.**

Ans:

(1) Jaccard_Index(X,Y) = $|X \cap Y| / |X \cup Y|$
Jaccard_index(A, B) =4/8
=1/2
Jaccard_distance(A,B) = 1 – ½ = ½

Jaccard_index(B, C) =4/8
=1/2

Jaccard_distance(B,C) = 1 – ½ = ½

Jaccard_index(C, A) = 4/8
Jaccard_distace(C,A) = 1 – ½ = ½

2)

Cosine Similarity $\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$

cos(A, B) = 34 / (√80 * √(9+16+9+1+4+1))
= 0.601

cos(B, C) =26 / (√(4+1+9+16+25+9) * √9+16+9+1+4+1
= 0.435

cos(A, C) =44 / (√80 * √(4+1+9+16+25+9))
= 0.615

3)

	a	b	c	d	e	f	g	h
A	4	5	0	5	1	0	3	2
B	0	3	4	3	1	2	1	0
C	2	0	1	3	0	4	5	3

⇓ {3,4,5} ← {1}
{1,2,0} ← {0}

	a	b	c	d	e	f	g	h
A	1	1	0	1	0	0	1	0
B	0	1	1	1	0	0	0	0
C	0	0	0	1	0	1	1	1

Jaccard - dist (A,B) = $1 - \frac{n(A \cap B)}{n(A \cup B)}$

$= 1 - \frac{2}{8}$

$= 1 - \frac{1}{4} = \frac{3}{4}$

Ily Jaccard - dist (A,C) = $1 - \frac{2}{8}$

$= \frac{3}{4}$

Jaccard - dist (B,C) = $1 - \frac{2}{8}$

$= \frac{3}{4}$

4)

End of Report

$$\cos(A,B) = \frac{2}{\sqrt{4}\sqrt{3}} = \frac{1}{\sqrt{3}}$$

$$\cos(B,C) = \frac{1}{\sqrt{3}\sqrt{4}} = \frac{1}{2\sqrt{3}}$$

$$\cos(A,C) = \frac{2}{\sqrt{4}\sqrt{4}} = \frac{1}{2}$$

5) mean (A) = 2.50
mean (B) = 1.75
mean (C) = 2.00

∴

	a	b	c	d	e	f	g	h
A	1.5	2.5	-2.5	2.5	-1.5	-2.5	0.5	-0.5
B	-1.75	1.25	2.25	1.25	-0.75	0.25	-0.75	-1.75
C	0	-2	-1	+1	-2	2	1	1

6. cosine dist^n

$$\cos(A,B) = \frac{(1.5\times(-1.75) + 2.5\times1.25 + (-2.5)\times2.25 + 2.5\times1.25 \times -1.5\times-0.75 + (-2.5)\times 0.25 + 0.5\times(-0.75) + (-0.5)\times(-1.75)}{\sqrt{(1.5)^2+(2.5)^2+(-2.5)^2+(2.5)^2+(-1.5)^2+(-2.5)^2+(0.5)^2+(-0.5)^2} \times \sqrt{(-1.75)^2+(1.25)^2+(2.25)^2+(1.25)^2+(-0.75)^2+(0.25)^2+(0.75)^2+(-1.75)^2}}$$

$$= -0.05$$

iii) $\cos(B,C) = \frac{(1.25\times(-2) + 2.25\times(-1) + (1.25\times1) + (-0.75)\times(-2) + 0.5 + (-0.75) - 1.75)}{\sqrt{(-1.75)^2+(1.25)^2+(2.25)^2+(1.25)^2+(-0.75)^2+(0.25)^2+(0.75)^2+(-1.75)^2} \times \sqrt{(2)^2+(1)^2+(1)^2+(2)^2+(2)^2+(1)^2+(1)^2}}$

$$= -0.25$$

iii) $\cos(A,C) = -0.09$

Q.2 Write an algorithm for finding triangles in social network graphs. How to use the algorithm using map reduce?

Ans:

$\succ$

Let $\succ$ be a total order on all of the vertices, with the property that v u if dv > du, with ties broken arbitrarily (but Consistently).

Algorithm 2 NodeIterator++(V,E)

```
1: T ← 0;
2: for v ∈ V do
3:     for u ∈ Γ(v) and u ≻ v do
4:         for w ∈ Γ(v) and w ≻ u do
5:             if ((u, w) ∈ E) then
6:                 T ← T + 1;
7: return T;
```

```
public static class Map extends MapReduceBase implements Mapper<LongWritable, Text, Text, IntWritable> {
		private final static IntWritable one = new IntWritable(1);
		private Text word = new Text();

		public void map(LongWritable key, Text value, OutputCollector<Text, IntWritable> output,
Reporter reporter) throws IOException
		{

			String line = value.toString();
			String [] split = line.split("\\s+");
			if(split.length > 0){
				String host = split[0];
		for(int i = 0; i < split.length; i++){
					for(int j = 0; j < split.length; j++){
						for(int h = 0; h < split.length; h++){
							if(((i != j && j != h) && i != h) && ((split[i] ==
host || split[j] == host) || split[h] == host)) {
								word.set("<" + split[i] + "," + split[j] + ","
+ split[h] + ">");
								output.collect(word,one);
							}
						}
					}
				}
			}
		}


public static class Reduce extends MapReduceBase implements Reducer<Text, IntWritable, Text, NullWritable> {
		private final static NullWritable nw = NullWritable.get();

		public void reduce(Text key, Iterator<IntWritable> values, OutputCollector<Text, NullWritable>
output, Reporter reporter) throws IOException {


			int sum = 0;
			while (values.hasNext()) {
				sum += values.next().get();
			}
			if(sum > 1){
				output.collect(key, nw);
			}
		}
	}

public static void main(String[] args) throws Exception {
		JobConf conf = new JobConf(TriangleCount.class);
		conf.setJobName("trianglecount");

		conf.setOutputKeyClass(Text.class);
		conf.setOutputValueClass(NullWritable.class);

		conf.setMapOutputKeyClass(Text.class);
		conf.setMapOutputValueClass(IntWritable.class);

		conf.setMapperClass(Map.class);
		conf.setReducerClass(Reduce.class);
```

```
conf.setInputFormat(TextInputFormat.class);
                conf.setOutputFormat(TextOutputFormat.class);

                FileInputFormat.setInputPaths(conf, new Path(args[0]));
                FileOutputFormat.setOutputPath(conf, new Path(args[1]));

                JobClient.runJob(conf);
        }
```