Assignment # 04

Introduction to Data Science

Name: Mohammad Ahmed Shahbaz

Reg. #: FA21-BSE-050

Section: BSE-C

Question 01:

Sentences:

S1. “data science is one of the most important courses in computer science”

S2. “this is one of the best data science courses”

S3. “the data scientists perform data analysis”

BOW:-

	S1	S2	S3
this	0	1	0
the	1	1	1
is	1	1	0
one	1	1	0
of	1	1	0
most	1	0	0
important	1	0	0
best	0	1	0
perform	0	0	1
data	1	1	2
computer	1	0	0
analysis	0	0	1
courses	1	1	0
science	2	1	0
in	1	0	0
scientists	0	0	1
Total Words	12	9	6

Vectors:	
Vector S1	[0 1 1 1 1 1 1 0 0 1 1 0 1 2 1 0]
Vector S2	[1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0]
Vector S3	[0 1 0 0 0 0 0 0 1 2 0 1 0 0 0 1]

TF:-

Total Number of words in S1: 12

TF(the) = 1/12
TF(this) = 0/12 = 0
TF(is) = 1/12
TF(one) = 1/12
TF(of) = 1/12
TF(most) = 1/12
TF(important) = 1/12
TF(best) = 0/12 = 0
TF(perform) = 0/12 = 0
TF(data) = 1/12
TF(computer) = 1/12
TF(analysis) = 0/12 = 0
TF(courses) = 1/12
TF(science) = 2/12 = 1/6
TF(in) = 1/12
TF(scientists) = 0/12 = 0

Total Number of words in S2: 9

TF(the) = 1/12
TF(this) = 1/12
TF(is) = 1/12
TF(one) = 1/12

TF(of) = 1/12
TF(most) = 0/12 = 0
TF(important) = 0/12 = 0
TF(best) = 1/12
TF(perform) = 0/12 = 0
TF(data) = 1/12
TF(computer) = 0/12 = 0
TF(analysis) = 0/12 = 0
TF(courses) = 1/12
TF(science) = 1/12
TF(in) = 0/12 = 0
TF(scientists) = 0/12 = 0

<u>**Total Number of words in S3: 6**</u>
TF(the) = 1/12
TF(this) = 0/12 = 0
TF(is) = 0/12 = 0
TF(one) = 0/12 = 0
TF(of) = 0/12 = 0
TF(most) = 0/12 = 0
TF(important) = 0/12 = 0
TF(best) = 0/12 = 0
TF(perform) = 1/12
TF(data) = 2/12 = 1/6
TF(computer) = 0/12 = 0
TF(analysis) = 1/12
TF(courses) = 0/12 = 0
TF(science) = 0/12 = 0
TF(in) = 0/12 = 0

this	the	is	one	of	most	import	best	perfo	data	compu	Anal	cours	scien	in	scientists	TotalW

							an t		r m		te r			ys is	e s		c e			or ds
S1	0	1/12	1/12	1/12	1/12	1/12	1/12	0	0	1/12	1/12			0	1/12		1/6	1/12	0	**12**
S2	1/12	1/12	1/12	1/12	1/12	0	0	1/12	0	1/12	0			0	1/12		1/12	0	0	**9**
S3	0	1/12	0	0	0	0	0	0	1/12	1/6	0			1/12	0		0	0	1/12	**6**

TF(scientists) = 1/12

IDF:-

IDF for S1:

IDF(data) = log(3/3) = log(1) = 0
IDF(science) = log(3/2) = log(1.5) = 0.176
IDF (is) = log(3/2) = log(1.5) = 0.176
IDF (one) = log(3/2) = log(1.5) = 0.176
IDF(of) = log(3/2) = log(1.5) = 0.176
IDF (the) = log(3/3) = log(1) = 0
IDF(most) = log(3/1) = log(3) = 0.477

IDF(important) = log(3/1) = log(3) = 0.477
IDF(courses) = log(3/2) = log(1.5) = 0.176
IDF(in) = log(3/1) = log(3) = 0.477
IDF(computer) = log(3/2) = log(1.5) = 0.176

<u>IDF for S2:</u>
IDF(this) = log(3/1) = log(3) = 0.477
IDF(is) = log((3/2) = log(1.5) = 0.176
IDF(one) = log(3/2) = log(1.5) = 0.176
IDF(of) = log(3/2) = log(1.5) = 0.176
IDF (the) = log(3/3) = log(1) = 0
IDF(best) = = log(3/1) = log(3) = 0.477
IDF(data) = log(3/3) = log(1) = 0
IDF(science) = log(3/2) = log(1.5) = 0.176
IDF(courses) = log(3/2) = log(1.5) = 0.176

<u>IDF for S3:</u>
IDF (the) = log(3/3) = log(1) = 0
IDF(data) = log(3/3) = log(1) = 0
IDF(scientists) = log(3/1) = log(3) = 0.477
IDF(perform) = log(3/1) = log(3) = 0.477
IDF(analysis) = log(3/1) = log(3) = 0.477

	IDF
this	0.477
the	0
is	0.176
one	0.176
of	0.176
most	0.477
important	0.477
best	0.477

perform	0.477
data	0
computer	0.176
analysis	0.477
courses	0.176
science	0.176
in	0.477
scientists	0.477

TF-IDF:-

TF-IDF for S1:

Tf-idf(data) = 1/12 * 0 = 0
Tf-idf(science) = 1/6 * 0.176 = 0.0293
Tf-idf(is) = 1/12 * 0.176 = 0.014
Tf-idf(one) = 1/12 * 0.176 = 0.014
Tf-idf(of) = 1/12 * 0.176 = 0.014
Tf-idf(the) = 1/12 * 0 = 0
Tf-idf(most) = 1/12 * 0.477 = 0.039
Tf-idf(important) = 1/12 * 0.477 = 0.039
Tf-idf(courses) = 1/12 * 0.176 = 0.014
Tf-idf(in) = 1/12 * 0.477 = 0.039
Tf-idf(computer) = 1/12 * 0.176 = 0.014

TF-IDF for S2:

Tf-idf(this) = 1/9 * 0.477 = 0.053
Tf-idf(is) = 1/9 * 0.176 = 0.019
Tf-idf(one) = 1/9 * 0.176 = 0.019
Tf-idf(of) = 1/9 * 0.176 = 0.019
Tf-idf(the) = 1/9 * 0 = 0
Tf-idf(best) = 1/9 * 0.477 = 0.053
Tf-idf(data) = 1/9 * 0 = 0
Tf-idf(science) = 1/9 * 0.176 = 0.019
Tf-idf(courses) = 1/9 * 0.176 = 0.019

<u>TF-IDF for S3:</u> "the data scientists perform data analysis"

Tf-idf(the) = 1/6 * 0 = 0

Tf-idf(data) = 1/3 * 0 = 0

Tf-idf(scientists) = 1/6 * 0.477 = 0.079

Tf-idf(perform) = 1/6 * 0.477 = 0.079

Tf-idf(analysis) = 1/6 * 0.477 = 0.079

	TF-IDF S1	TF-IDF S2	TF-IDF S3
this	0	0.053	0
the	0	0	0
is	0.014	0.019	0
one	0.014	0.019	0
of	0.014	0.019	0
most	0.039	0	0
important	0.039	0	0
best	0	0.053	0
perform	0	0	0.079
data	0	0	0
computer	0.014	0	0
analysis	0	0	0.079
courses	0.176	0.019	0
science	0.029	0.019	0
in	0.039	0	0
scientists	0	0	0.079

<u>Python Code:-</u>

```
import math
s1 = "data science is one of the most important courses in computer
science";
s2 = "this is one of the best data science courses"
s3 = "the data scientists perform data analysis"
len1 = len(s1);
len2 = len(s2);
len3 = len(s3);
sp_s1 = s1.split()
sp_s2 = s2.split();
sp_s3 = s3.split();
def calculate_term_frequency(words_list):
    total_words = len(words_list)
    word_frequency = {}
    for word in words_list:
        if word in word_frequency:
            word_frequency[word] += 1
        else:
            word_frequency[word] = 1
            term_frequency = {word: freq / total_words for word, freq in
word_frequency.items()}
    return term_frequency
tf_s1 = calculate_term_frequency(sp_s1)
```

```
tf_s2 = calculate_term_frequency(sp_s2)
tf_s3 = calculate_term_frequency(sp_s3)
print("Term Frequency in s1:", tf_s1)
print("Term Frequency in s2:", tf_s2)
print("Term Frequency in s3:", tf_s3)
def calculate_idf(documents):
    document_frequency {}

    for doc in documents:
        words = set(doc.split())
        for word in words:
            if word in document_frequency:
              document_frequency:[word] += 1
              else
              document_frequency:[word] = 1

    total_documents = len(documents)
    idf = {word: math.log(total_documents /
(document_frequency[word] + 1)) for word in document_frequency}
    return idf

documents = [s1, s2, s3];
idf_values = calculate_idf(documents)
print("IDF Values:")
for word, idf in idf_values.items():
```

```
    print(f"{word}: {idf}")
tfidf_values = [{word: tf_values[i][word] * idf_values[word] for word in tf_values[i]} for i in range(len(tf_values))]
print("TF-IDF Values:")
for i, tfidf in enumerate(tfidf_values):
    print(f"TF-IDF Values for Document {i + 1}: {tfidf}")
```

Question 02:

Cosine Similarity:-

Vector S1	**[0 1 1 1 1 1 1 0 0 1 1 0 1 2 1 0]**
Vector S2	[1 1 1 1 1 0 0 1 0 1 0 0 1 1 0 0]
Vector S3	[0 1 0 0 0 0 0 0 1 2 0 1 0 0 0 1]

Vector Length of S1 = **($\sum (S1i)^2$) ^ 1.5** = 0 + 1 + 1 + 1 + 1 + 1 + 1 + 0 + 0 + 1 + 1 + 0 + 1 + 4 + 1 + 0 = **14**
Vector Length of S2 = **($\sum (S2i)^2$) ^ 1.5 =** 1 + 1 + 1 + 1 + 1 + 0 + 0 + 1 + 0 + 1+ 0 + 0 + 1 + 1 + 0 + 0 = **9**

Vector Length of S3 = **(∑ $(S3i)^2$) ^ 1.5 =** 0 + 1 + 1 + 4 + 1 + 1 = **8**
Dot Product of S1 and S2 = 0*1 + 1*1 + 1*1 + 1*1 + 1*1 + 1*0 + 1*0 + 0*1 + 0*0 + 1*1 + 1*0 + 0*0 + 1*1 + 2*1 + 1*0 + 0*0 = 1+1+1+1+1+1+2 = 8
Dot Product of S1 and S3 = 0*0 + 1*1 + 1*0 + 1*0 + 1*0 + 1*0 + 1*0 + 0*0 + 0*1 + 1*2 + 1*0 + 0*1 + 1*0 + 2*0 + 1*0 = 1+2 = 3
Dot product of S2 and S3 = 1*0 + 1*1 + 1*0 + 1*0 + 1*0 + 0*0 +0*0 + 1*0 + 0*1+ 1*2+ 0*0 + 1*1+ 1*0 + 1*0 + 0*0 + 0*1 = 1+2+1 = 3
Cosine S1-S2 = 8/ 14* 9 = 0.0634
Cosine S2-S3 = 3/ 9*8 = 0.04166
Cosine S1-S3 = 3/ 14* 8 = 0.0267

Manhattan Distance:

Manhattan distance is the sum of the absolute differences between corresponding components of vectors.

Manhattan distance between:

- S1 and S2: ∑∣S1i−S2i∣
- S1 and S3: ∑∣S1i−S3i∣
- S2 and S3: ∑∣S2i−S3i∣

Results:

- Manhattan distance between S1 and S2: 1.7045
- Manhattan distance between S1 and S3: 1.9502
- Manhattan distance between S2 and S3: 1.4861

Euclidean Distance:

Euclidean distance is the square root of the sum of the squared differences between corresponding components of vectors.

Euclidean distance between:

- S1 and S2: $\sum(S1i-S2i)^{(1/2)2}$
- S1 and S3: $\sum(S1i-S3i)^{(1/2)2}$
- S2 and S3: $\sum(S2i-S3i)^{(1/2)2}$

Results:

- Euclidean distance between S1 and S2: 0.9474
- Euclidean distance between S1 and S3: 1.1491
- Euclidean distance between S2 and S3: 0.8855

Python Code:-

```
from collections import Counter

import math

s1 = "data science is one of the most important courses in computer science"

s2 = "this is one of the best data science courses"

s3 = "the data scientists perform data analysis"

tokens_s1 = s1.split()
```

```
tokens_s2 = s2.split()
tokens_s3 = s3.split()
vector_s1 = Counter(tokens_s1);
vector_s2 = Counter(tokens_s2);
vector_s3 = Counter(tokens_s3);
def cosine_similarity(vec1, vec2):
    intersection = set(vec1.keys()) & set(vec2.keys())
    numerator = sum(vec1[word] * vec2[word] for word in
intersection)
    sum_sq_vec1 = sum(vec1[word] ** 2 for word in vec1.keys())
    sum_sq_vec2 = sum(vec2[word] ** 2 for word in vec2.keys())
    denominator = math.sqrt(sum_sq_vec1) * math.sqrt(sum_sq_vec2)
    if not denominator:
        return 0.0
    else:
        return float(numerator) / denominator
sim_s1_s2 = cosine_similarity(vector_s1, vector_s2)
sim_s1_s3 = cosine_similarity(vector_s1, vector_s3)
sim_s2_s3 = cosine_similarity(vector_s2, vector_s3)

# Display cosine similarity results
print("Cosine Similarity between s1 and s2:", sim_s1_s2)
print("Cosine Similarity between s1 and s3:", sim_s1_s3)
print("Cosine Similarity between s2 and s3:", sim_s2_s3)
```