**PRACTICAL 7**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VII | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** | 10-09-24 | **Batch:** | G11 |
| **Aim:** | Understanding Jaccard Similarity. | | | | |

**Jaccard Similarity**

Jaccard Similarity is a measure of similarity between two asymmetric binary vectors or we can say a way to find the similarity between two sets. It is a common proximity measurement used to compute the similarity of two items, such as two text documents. The index ranges from 0 to 1. Range closer to 1 means more similarity in two sets of data.

**Code**

def jaccard\_similarity(list1, list2):

intersection = 0

union = 0

for a, b in zip(list1, list2):

if a == 1 or b == 1:

union += 1

if a == 1 and b == 1:

intersection += 1

if union == 0:

return 0

return intersection / union

C1 = [0, 1, 0, 0, 0, 1, 0, 0, 1]

C2 = [0, 0, 1, 0, 0, 0, 0, 0, 1]

C3 = [1, 1, 0, 0, 0, 1, 0, 0, 0]

similarity\_C1\_C2 = jaccard\_similarity(C1, C2)

similarity\_C1\_C3 = jaccard\_similarity(C1, C3)

similarity\_C2\_C3 = jaccard\_similarity(C2, C3)

print(f"Similarity - Customer C1 and C2 is {similarity\_C1\_C2}")

print(f"Similarity - Customer C1 and C3 is {similarity\_C1\_C3}")

print(f"Similarity - Customer C2 and C3 is {similarity\_C2\_C3}")

def jaccard\_similarity\_sets(set1, set2):

intersection = len(set(set1).intersection(set2))

union = len(set(set1).union(set2))

return intersection / union

S1 = [0, 2, 5, 7, 9]

S2 = [0, 1, 2, 4, 5, 6, 8]

similarity\_S1\_S2 = jaccard\_similarity\_sets(S1, S2)

print(f"Similarity between Set S1 and S2 is {similarity\_S1\_S2}")

**Output**

