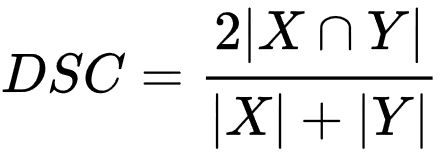
Sørensen–Dice coefficient

**Similarity**

Similarity measures how alike two things are. It helps us compare and find commonalities between entities or data sets.

**Sørensen-Dice similarity**

The Sørensen-Dice similarity coefficient algorithm, also known as the Sørensen-Dice index, is a similarity measure used to compare the similarity or overlap between two sets. It quantifies the resemblance between sets based on the size of their intersection relative to the sum of their sizes.

Sørensen-Dice similarity is particularly useful when dealing with binary data or situations where presence or absence of elements is important. It provides a measure of similarity that takes into account the size of the sets and emphasizes the shared elements.

**step-by-step explanation**

1. Input: Provide two sets, denoted as Set A and Set B, for which you want to calculate the similarity.
2. Intersection: Determine the number of elements that are common to both Set A and Set B, which is the size of the intersection (denoted as |A ∩ B|).
3. Sum of set sizes: Calculate the sum of the sizes of Set A and Set B, denoted as (|A| + |B|) without reptation.
4. Similarity Calculation: Compute the Sørensen-Dice similarity coefficient by dividing twice the size of the intersection (2 \* |A ∩ B|) by the size of the union (|A ∪ B|).

Sørensen-Dice similarity coefficient = (2 \* |A ∩ B|) / (|A ∪ B|)

1. Output: The resulting coefficient represents the similarity or overlap between Set A and Set B. It ranges from 0 to 1, with 1 indicating complete similarity and 0 indicating no similarity.

**Example**

|  |  |
| --- | --- |
| Comm 1 | Comm 2 |
| Tigers | Tigers |
| Lions | Birds |
| Mice | Lions |
| Elephants | Mice |
| Mice | Kangaroo |
|  |  |

Tigers

Lions

Mice

Elephants

Kangaroo

Birds

Comm 2

Comm 1

S1 = 4

S2 = 5

Intersection (S1 ∩ S2) = 3

Union = (S1 ∪ S2) = 9

DCS(Comm1, Comm2) = = = 0.6

**Differences Between Jaccard and Sørensen-Dice**

|  |  |  |
| --- | --- | --- |
| Difference | Jaccard Similarity | Sørensen-Dice Similarity |
| Calculation | Intersection divided by Union | Twice the Intersection divided by Sum of Set Sizes |
| Interpretation | Measures relative size of intersection | Measures relative size of intersection with emphasis on set sizes |
| Range | 0 to 1 | 0 to 1 |
| Penalties | No penalty for missing elements | Penalizes for missing elements |
| Application | Information retrieval, data mining | Image segmentation, region comparison |
| Data Type | Binary or sparse data | Binary or sparse data |
| Sensitivity to Small Intersection | Less sensitive | More sensitive |

**Code in Python:**

    import pandas as pd

    import matplotlib.pyplot as plt

    from matplotlib\_venn import venn2

    def calculate\_dice\_coefficient(set1, set2):

        intersection = len(set1.intersection(set2))

        dice\_coefficient = (2.0 \* intersection) / (len(set1) + len(set2))

        return dice\_coefficient

    def visualize\_venn\_diagram(set1, set2):

        plt.figure(figsize=(6, 6))

        venn2([set1, set2], set\_labels=('Comm 1', 'Comm 2'))

        plt.title("Sørensen-Dice Coefficient Venn Diagram")

        plt.show()

    # Read the dataset from CSV file

    data = pd.read\_csv('dataset.csv')

    # Convert the columns to sets

    word\_set1 = set(data['Comm 1'])

    word\_set2 = set(data['Comm 2'])

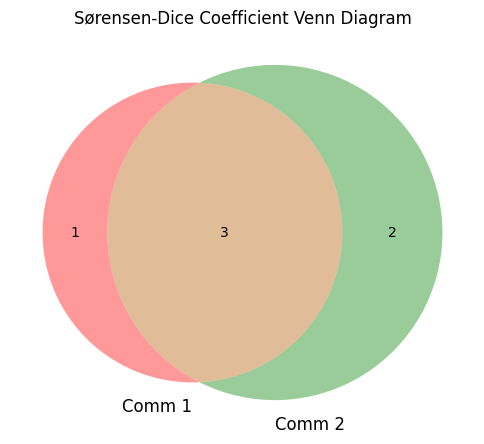
    dice\_coefficient = calculate\_dice\_coefficient(word\_set1, word\_set2)

    print("Sørensen-Dice Coefficient:", dice\_coefficient)

    visualize\_venn\_diagram(word\_set1, word\_set2)

**output:**

# Sørensen-Dice Coefficient: 0.666666666

****