

Domanda

For the following vectors, x and y, calculate the indicated similarity or distance measures.

(a) $x = (1, 1, 1, 1)$, $y = (2, 2, 2, 2)$ cosine, correlation, Euclidean, Jaccard

(b) $x = (0, 1, 0, 1)$, $y = (1, 0, 1, 0)$ cosine, correlation, Euclidean, Jaccard

(c) $x = (0, -1, 0, 1)$, $y = (1, 0, -1, 0)$ cosine, correlation, Euclidean, Jaccard

(d) $x = (1, 1, 0, 1, 0, 1)$, $y = (1, 1, 1, 0, 0, 1)$ cosine, correlation, Jaccard

(e) $x = (2, -1, 0, 2, 0, -3)$, $y = (-1, 1, -1, 0, 0, -1)$ cosine, correlation, Jaccard

Risposta

```
In [ ]: import numpy as np
        from scipy.spatial.distance import cosine, euclidean
```

```
In [ ]: # Definizione dei vettori x e y per ogni esercizio.
        # Il primo gruppo di vettori prevede il calcolo della distanza euclidea.
        # Il secondo gruppo di vettori non prevede il calcolo della distanza euclidea.
        vectors1 = {
            'a': {'x': np.array([1, 1, 1, 1]),
                  'y': np.array([2, 2, 2, 2])},
            'b': {'x': np.array([0, 1, 0, 1]),
                  'y': np.array([1, 0, 1, 0])},
            'c': {'x': np.array([0, -1, 0, 1]),
                  'y': np.array([1, 0, -1, 0])}
        }

        vectors2 = {
            'd': {'x': np.array([1, 1, 0, 1, 0, 1]),
                  'y': np.array([1, 1, 1, 0, 0, 1])},
            'e': {'x': np.array([2, -1, 0, 2, 0, -3]),
                  'y': np.array([-1, 1, -1, 0, 0, -1])}
        }
```

```
In [ ]: def correlation (x,y):
        # Calcolo della media dei vettori
        mean_x = np.mean(x)
        mean_y = np.mean(y)

        # Calcolo delle differenze rispetto alla media
        diff_x = x - mean_x
        diff_y = y - mean_y

        # Calcolo del numeratore e denominatore della correlazione
        numerator = np.sum(diff_x * diff_y)
        denominator = np.sqrt(np.sum(diff_x**2) *
                                np.sum(diff_y**2))

        # Calcolo del coefficiente di correlazione
```

```

if denominator == 0:
    correlation_coeff = 0 # Evitare la divisione per zero
else:
    correlation_coeff = numerator / denominator

return correlation_coeff

```

```

In [ ]: # Calcolo delle misure per ogni esercizio
for exercise, data in vectors1.items():
    x = data['x']
    y = data['y']

    cosine_sim = cosine(x, y)
    correlation_coeff = correlation(x,y)
    euclidean_dist = euclidean(x, y)

    # Calcolo della Jaccard similarity
    intersection = np.sum(np.minimum(x, y))
    union = np.sum(np.maximum(x, y))
    jaccard_sim = intersection / union

    # Stampare i risultati
    print(f"Esercizio {exercise}:")
    print(f"Cosine similarity: {cosine_sim}")
    print(f"Correlation coefficient: {correlation_coeff}")
    print(f"Euclidean distance: {euclidean_dist}")
    print(f"Jaccard similarity: {jaccard_sim}\n")

for exercise, data in vectors2.items():
    x = data['x']
    y = data['y']

    cosine_sim = cosine(x, y)
    correlation_coeff = correlation(x,y)

    # Calcolo della Jaccard similarity
    intersection = np.sum(np.minimum(x, y))
    union = np.sum(np.maximum(x, y))
    jaccard_sim = intersection / union

    # Stampare i risultati
    print(f"Esercizio {exercise}:")
    print(f"Cosine similarity: {cosine_sim}")
    print(f"Correlation coefficient: {correlation_coeff}")
    print(f"Jaccard similarity: {jaccard_sim}\n")

```

Esercizio a:

Cosine similarity: 0

Correlation coefficient: 0

Euclidean distance: 2.0

Jaccard similarity: 0.5

Esercizio b:

Cosine similarity: 1.0

Correlation coefficient: -1.0

Euclidean distance: 2.0

Jaccard similarity: 0.0

Esercizio c:

Cosine similarity: 1.0

Correlation coefficient: 0.0

Euclidean distance: 2.0

Jaccard similarity: -1.0

Esercizio d:

Cosine similarity: 0.25

Correlation coefficient: 0.25

Jaccard similarity: 0.6

Esercizio e:

Cosine similarity: 1.0

Correlation coefficient: -5.73316704659901e-17

Jaccard similarity: -1.5