Experiment Number : 5 - Applying similarity measures on the numeric datasets

---

Batch: FDS-2 **Roll Number: 16010422234** **Name: Chandana Ramesh Galgali**

Aim of the Experiment: Applying similarity measures on the numeric datasets and textual datasets

---

Program/ Steps:

Identify the suitable attributes to apply the numeric similarity measures and write python code to calculate Euclidean, Manhattan similarity measures on it.

Code with Output/Result:

1. Importing Libraries and creating dataset:

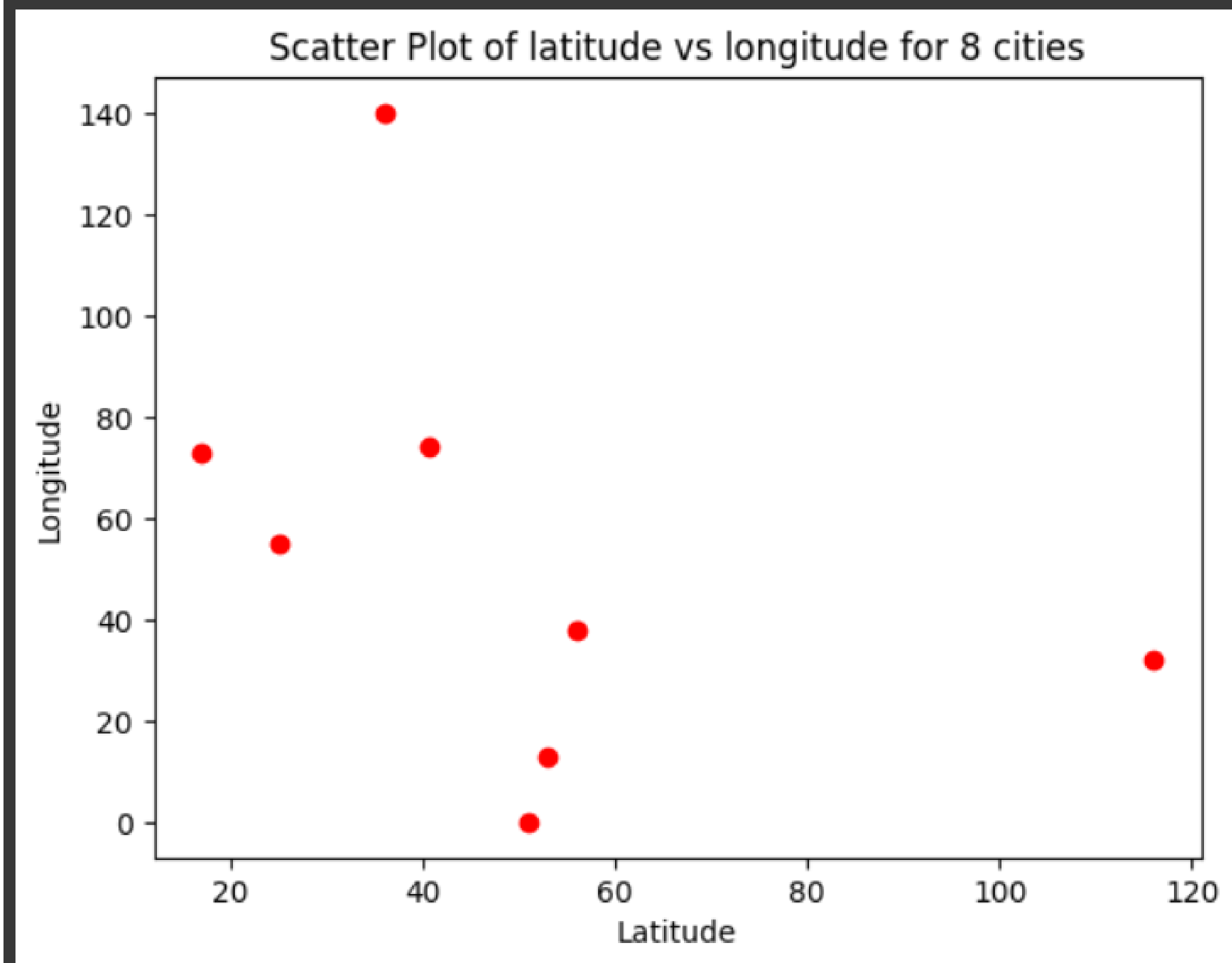
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
data = {
    'longitude': [73, 55, 0.12, 13, 38, 32, 140, 74],
    'latitude': [17, 25, 51, 53, 56, 116, 36, 40.7],
    'city': ['Mumbai', 'Dubai', 'London', 'Berlin', 'Moscow', 'Perth', 'Tokyo', 'New York']
}
df=pd.DataFrame(data)
print(df)
```

```
   longitude  latitude      city
0      73.00      17.0    Mumbai
1      55.00      25.0     Dubai
2       0.12      51.0    London
3      13.00      53.0    Berlin
4      38.00      56.0    Moscow
5      32.00     116.0     Perth
6     140.00      36.0     Tokyo
7      74.00      40.7  New York
```

2. Scatter Plot:

```
plt.scatter(df["latitude"], df["longitude"], c="red")
plt.xlabel("Latitude")
plt.ylabel("Longitude")
plt.title("Scatter Plot of latitude vs longitude for 8 cities")
plt.show()
```



3. Euclidean Distance:

```
def euclidean(info):
  dist=[]
  for _, i in df.iterrows():
    d=((i.latitude-info.latitude)**2 +(i.longitude-info.longitude)**2)**0.5
    dist.append(round(d, 2))
  return dist
```

```
df_euclidean=df.copy(deep=True)
for _, i in df.iterrows():
  df_euclidean[f"Euclidean Distance from{i.city}"]=euclidean(i)
```

```
print(df_euclidean)
```

```
   longitude  latitude      city  Euclidean Distance fromMumbai  \
0      73.00      17.0    Mumbai                           0.00
1      55.00      25.0     Dubai                          19.70
2       0.12      51.0    London                          80.42
3      13.00      53.0    Berlin                          69.97
4      38.00      56.0    Moscow                          52.40
5      32.00     116.0     Perth                         107.15
6     140.00      36.0     Tokyo                          69.64
7      74.00      40.7  New York                          23.72

   Euclidean Distance fromDubai  Euclidean Distance fromLondon  \
0                         19.70                          80.42
1                          0.00                          60.73
2                         60.73                           0.00
3                         50.48                          13.03
4                         35.36                          38.21
5                         93.86                          72.40
6                         85.71                         140.68
7                         24.65                          74.59

   Euclidean Distance fromBerlin  Euclidean Distance fromMoscow  \
0                          69.97                          52.40
1                          50.48                          35.36
2                          13.03                          38.21
3                           0.00                          25.18
4                          25.18                           0.00
5                          65.80                          60.30
6                         128.13                         103.94
7                          62.23                          39.12

   Euclidean Distance fromPerth  Euclidean Distance fromTokyo  \
0                        107.15                         69.64
1                         93.86                         85.71
2                         72.40                        140.68
3                         65.80                        128.13
4                         60.30                        103.94
5                          0.00                        134.40
6                        134.40                          0.00
7                         86.22                         66.17

   Euclidean Distance fromNew York
0                            23.72
1                            24.65
2                            74.59
3                            62.23
4                            39.12
5                            86.22
6                            66.17
7                             0.00
```

4. Manhattan Distance:

```python
def manhattan(info):
  dist=[]
  for _, i in df.iterrows():
    d=(abs(i.latitude-info.latitude) + abs(i.longitude-info.longitude))
    dist.append(round(d, 2))
  return dist
```

```python
df_manhattan=df.copy(deep=True)
for _, i in df.iterrows():
  df_manhattan[f"Manhattan Distance from {i.city}"]=manhattan(i)
```

```python
print(df_manhattan)
```

```
   longitude  latitude      city  Manhattan Distance from Mumbai  \
0      73.00      17.0    Mumbai                            0.00   
1      55.00      25.0     Dubai                           26.00   
2       0.12      51.0    London                          106.88   
3      13.00      53.0    Berlin                           96.00   
4      38.00      56.0    Moscow                           74.00   
5      32.00     116.0     Perth                          140.00   
6     140.00      36.0     Tokyo                           86.00   
7      74.00      40.7  New York                           24.70   

   Manhattan Distance from Dubai  Manhattan Distance from London  \
0                          26.00                          106.88   
1                           0.00                           80.88   
2                          80.88                            0.00   
3                          70.00                           14.88   
4                          48.00                           42.88   
5                         114.00                           96.88   
6                          96.00                          154.88   
7                          34.70                           84.18   

   Manhattan Distance from Berlin  Manhattan Distance from Moscow  \
0                           96.00                           74.00   
1                           70.00                           48.00   
2                           14.88                           42.88   
3                            0.00                           28.00   
4                           28.00                            0.00   
5                           82.00                           66.00   
6                          144.00                          122.00   
7                           73.30                           51.30   

   Manhattan Distance from Perth  Manhattan Distance from Tokyo  \
0                         140.00                          86.00   
1                         114.00                          96.00   
2                          96.88                         154.88   
3                          82.00                         144.00   
4                          66.00                         122.00   
5                           0.00                         188.00   
6                         188.00                           0.00   
7                         117.30                          70.70   

   Manhattan Distance from New York  
0                             24.70  
1                             34.70  
2                             84.18  
3                             73.30  
4                             51.30  
5                            117.30  
6                             70.70  
7                              0.00  
```

5. Minkowski Distance:

```
def minkowski(info):
  p=3
  dist=[]
  for _, i in df.iterrows():
    d=(abs(i.latitude-info.latitude)**p + abs(i.longitude-info.longitude)**p)**(1/3)
    dist.append(round(d, 2))
  return dist
```

```
df_minkowski=df.copy(deep=True)
for _, i in df.iterrows():
  df_minkowski[f"Minkowski Distance from {i.city}"]=minkowski(i)
```

```
print(df_minkowski)
```

```
   longitude  latitude      city  Minkowski Distance from Mumbai  \
0      73.00      17.0    Mumbai                            0.00
1      55.00      25.0     Dubai                           18.51
2       0.12      51.0    London                           75.27
3      13.00      53.0    Berlin                           64.04
4      38.00      56.0    Moscow                           46.75
5      32.00     116.0     Perth                          101.29
6     140.00      36.0     Tokyo                           67.51
7      74.00      40.7  New York                           23.70

   Minkowski Distance from Dubai  Minkowski Distance from London  \
0                          18.51                           75.27
1                           0.00                           56.76
2                          56.76                            0.00
3                          45.79                           12.90
4                          32.62                           37.91
5                          91.49                           67.46
6                          85.06                          139.94
7                          22.06                           73.95

   Minkowski Distance from Berlin  Minkowski Distance from Moscow  \
0                           64.04                           46.75
1                           45.79                           32.62
2                           12.90                           37.91
3                            0.00                           25.01
4                           25.01                            0.00
5                           63.57                           60.02
6                          127.10                          102.26
7                           61.17                           36.90

   Minkowski Distance from Perth  Minkowski Distance from Tokyo  \
0                         101.29                          67.51
1                          91.49                          85.06
2                          67.46                         139.94
3                          63.57                         127.10
4                          60.02                         102.26
5                           0.00                         121.00
6                         121.00                           0.00
7                          79.43                          66.01

   Minkowski Distance from New York
0                             23.70
1                             22.06
2                             73.95
3                             61.17
4                             36.90
5                             79.43
6                             66.01
7                              0.00
```

6. Mahalanobis Distance:

```
import numpy as np
def mahalanobis_distance(x, mean, covariance):
  d = x - mean
  inv_covariance = np.linalg.inv(covariance)
  distance = np.sqrt(np.dot(np.dot(d.T, inv_covariance), d))
  return distance
```

```
point = np.array([1.5, 2.0])
mean = np.array([1.0, 1.5])
covariance = np.array([[1.0, 0.5], [0.5, 1.0]])
distance = mahalanobis_distance(point, mean, covariance)
print("Mahalanobis Distance: ", distance)
```

```
Mahalanobis Distance:  0.5773502691896257
```

6. Bhattacharyya Distance:

```
import numpy as np
import math
P = np.array([0.3, 0.4, 0.2, 0.1])
Q = np.array([0.2, 0.3, 0.3, 0.2])
```

```
def bhattacharyya_distance(p, q):
  bc = np.sum(np.sqrt(p * q))
  b_distance = -math.log(bc)
  return b_distance
```

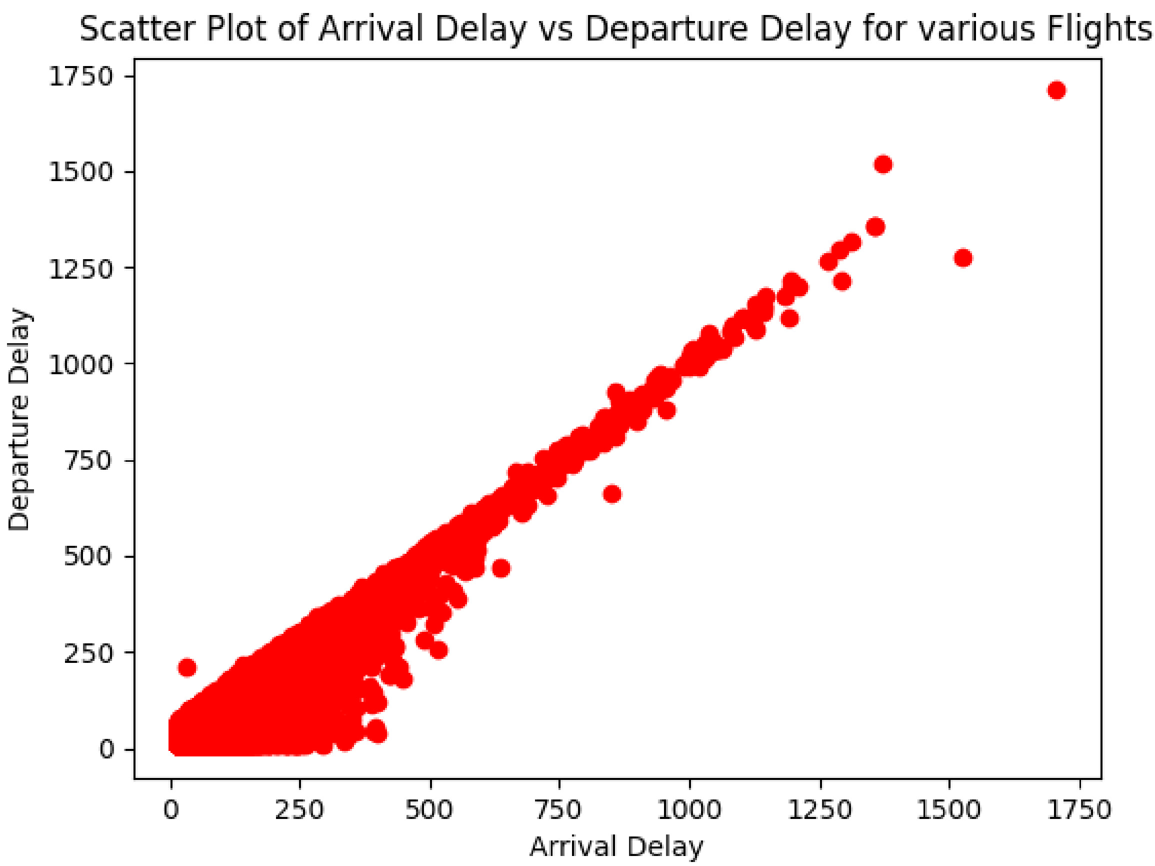
```
b_distance = bhattacharyya_distance(P, Q)
print("Bhattacharyya Distance:", b_distance)
```

```
Bhattacharyya Distance: 0.022522266530759078
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_csv(r'C:\Users\daxay\Downloads\Flight_delay.csv')
data_array = data.to_numpy()
dataframe = {
    'Flight Number': data_array[:, 7],
    'Arrival Delay': data_array[:, 12],
    'Departure Delay': data_array[:, 13]
}
df = pd.DataFrame(dataframe)
print("Dataframe:\n", df)
plt.scatter(df["Arrival Delay"], df["Departure Delay"], c="red")
plt.xlabel("Arrival Delay")
plt.ylabel("Departure Delay")
plt.title("Scatter Plot of Arrival Delay vs Departure Delay for various
Flights")
plt.show()
```

```
Dataframe:
         Flight Number Arrival Delay Departure Delay
0                3920            34              34
1                 509            57              67
2                1333            80              94
3                 675            15              27
4                   4            16              28
...               ...           ...             ...
484546           1496            27              34
484547           1496            39              41
484548           1496            47              42
484549           1496            26              32
484550           1496            18              33

[484551 rows x 3 columns]
```




```
import numpy as np
import pandas as pd

data = pd.read_csv(r'C:\Users\daxay\Downloads\Flight_delay.csv')

data_array = data.to_numpy()

dataframe = {
    'Flight Number': data_array[:4, 7],
    'Arrival Delay': data_array[:4, 12],
    'Departure Delay': data_array[:4, 13],
}

df = pd.DataFrame(dataframe)
print("Dataframe:\n", df)

def euclidean(info, df):
    dist = []
```

```
    for _, i in df.iterrows():
        d = ((i['Arrival Delay'] - info['Arrival Delay'])**2 +
(i['Departure Delay'] - info['Departure Delay'])**2)**0.5
        dist.append(round(d, 2))
    return dist

df_euclidean = df.copy(deep=True)
for _, i in df.iterrows():
    df_euclidean[f"Euclidean Distance from {i['Flight Number']}"] =
euclidean(i, df)
print(df_euclidean)

def manhattan(info, df):
    dist = []
    for _, i in df.iterrows():
        d = (abs(i['Arrival Delay'] - info['Arrival Delay']) +
abs(i['Departure Delay'] - info['Departure Delay']))
        dist.append(round(d, 2))
    return dist

df_manhattan = df.copy(deep=True)
for _, i in df.iterrows():
    df_manhattan[f"Manhattan Distance from {i['Flight Number']}"] =
manhattan(i, df)
print(df_manhattan)

def minkowski(info, df):
    p = 3
    dist = []
    for _, i in df.iterrows():
        d = (abs(i['Arrival Delay'] - info['Arrival Delay'])**p +
abs(i['Departure Delay'] - info['Departure Delay'])**p)**(1/3)
        dist.append(round(d, 2))
    return dist

df_minkowski = df.copy(deep=True)
for _, i in df.iterrows():
    df_minkowski[f"Minkowski Distance from {i['Flight Number']}"] =
minkowski(i, df)
print(df_minkowski)
```

```
Dataframe:
   Flight Number Arrival Delay Departure Delay
0           3920            34              34
1            509            57              67
2           1333            80              94
3            675            15              27
   Flight Number Arrival Delay Departure Delay  ... Euclidean Distance from 509 Euclidean Distance from 1333 Euclidean Distance from 675
0           3920            34              34  ...                       40.22                        75.60                       20.25
1            509            57              67  ...                        0.00                        35.47                       58.00
2           1333            80              94  ...                       35.47                         0.00                       93.35
3            675            15              27  ...                       58.00                        93.35                        0.00

[4 rows x 7 columns]
   Flight Number Arrival Delay Departure Delay  ... Manhattan Distance from 509 Manhattan Distance from 1333 Manhattan Distance from 675
0           3920            34              34  ...                          56                          106                          26
1            509            57              67  ...                           0                           50                          82
2           1333            80              94  ...                          50                            0                         132
3            675            15              27  ...                          82                          132                           0

[4 rows x 7 columns]
   Flight Number Arrival Delay Departure Delay  ... Minkowski Distance from 509 Minkowski Distance from 1333 Minkowski Distance from 675
0           3920            34              34  ...                       36.37                        67.92                       19.31
1            509            57              67  ...                        0.00                        31.70                       51.69
2           1333            80              94  ...                       31.70                         0.00                       83.17
3            675            15              27  ...                       51.69                        83.17                        0.00
```

Post Lab Question-Answers:

1. What is distance in Data Science and what is its importance?

Ans: In the context of data science, distance refers to a measure of dissimilarity or similarity between two data points. It quantifies the separation or similarity between observations in a dataset. Distance metrics are used in various data science tasks, such as clustering, classification, and anomaly detection.

The importance of distance in data science lies in its ability to provide a quantitative measure of how different or similar data points are. It allows us to compare and analyze patterns, relationships, and structures within the data. By calculating distances, we can identify similar data points that belong to the same group or cluster, or detect outliers that are significantly different from the rest of the data.

Distance metrics, such as Euclidean distance, Manhattan distance, or cosine similarity, enable us to perform calculations and make informed decisions based on the proximity or dissimilarity of data points. They are fundamental tools in data science algorithms and techniques, helping us uncover insights, make predictions, and solve real-world problems.

2. What are the different applications of Numeric similarity measure?

Ans: Numeric similarity measures, also known as distance metrics, have various applications in data science. Some of the key applications include:

1. Clustering: Numeric similarity measures are used to group similar data points together in clustering algorithms. By calculating the distances between data points, clustering algorithms can identify natural groupings or clusters within a dataset.

2. Classification: Similarity measures are used in classification algorithms to determine the similarity between a new data point and existing labeled data points. This similarity is then used to assign a class label to the new data point.

3. Anomaly detection: Numeric similarity measures can help identify anomalies or outliers in a dataset. Data points that are significantly different from the majority of the data can be detected by calculating their distances from the rest of the data points.

4. Recommender systems: Similarity measures are used in recommender systems to find items or products that are similar to a user's preferences. By calculating the similarity between user profiles or item features, recommender systems can provide personalized recommendations.

5. Dimensionality reduction: Similarity measures are used in dimensionality reduction techniques, such as t-SNE or PCA, to preserve the similarity relationships between data points in lower-dimensional representations.

6. Time series analysis: Similarity measures are used to compare and analyze time series data. By calculating the similarity between different time series, patterns, trends, or anomalies can be identified.

These are just a few examples of the applications of numeric similarity measures in data science. The versatility of these measures allows them to be applied in various domains and problem-solving scenarios.

3. Why use Mahalanobis distance if Euclidean distances are available? Give suitable examples with justification.

Ans: The Mahalanobis distance is used when there are correlations or dependencies between variables in the dataset, which cannot be captured by the Euclidean distance. Unlike the Euclidean distance, the Mahalanobis distance takes into account the covariance structure of the data, making it a more appropriate choice in certain scenarios. Here are a few examples where the Mahalanobis distance is preferred over the Euclidean distance:

1. Outlier detection: In outlier detection, the Mahalanobis distance is useful when the variables in the dataset are correlated. By considering the covariance structure, the Mahalanobis distance can identify outliers that deviate from the expected patterns, even if they are not far away in terms of Euclidean distance.

2. Multivariate analysis: When analyzing multivariate data, the Mahalanobis distance is used to measure the similarity or dissimilarity between observations. It accounts for the correlations between variables, allowing for a more accurate assessment of the distance between data points.

3. Anomaly detection in high-dimensional data: In high-dimensional datasets, the Euclidean distance becomes less reliable due to the curse of dimensionality. The Mahalanobis distance, on the other hand, can handle high-dimensional data by considering the covariance structure, making it more suitable for anomaly detection in such cases.

4. Classification with imbalanced data: In classification tasks with imbalanced data, where the number of samples in different classes is significantly different, the Mahalanobis distance can help address the issue. By considering the covariance structure, the Mahalanobis distance can give more weight to the minority class, leading to better classification performance.

In summary, the Mahalanobis distance is preferred over the Euclidean distance when there are correlations or dependencies between variables in the dataset. It provides a more accurate measure of distance by considering the covariance structure, making it suitable for outlier detection, multivariate analysis, anomaly detection in high-dimensional data, and classification with imbalanced data.

Outcomes: Comprehend descriptive and proximity measures of data

---

Conclusion (based on the Results and outcomes achieved):

References:

Books/ Journals/ Websites

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition

---