Darshan UNIVERSITY

योग: कर्मसु कौशलम्

Data Mining

Lab - 4

Enrollment no.:- 22010101478

Name:- Jay Ramani

Batch:- A-3

Roll no.:- 156

Part -1

1) Write a python program to compute distance between Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):

(a) Compute the Euclidean distance between the two objects.

(b) Compute the Manhattan distance between the two objects.

(c) Compute the Minkowski distance between the two objects, using q = 3.

(d) Compute the supremum distance between the two objects.

In [3]:

```python
import math

Euclidean = math.sqrt((22-20)**2 + (1-0)**2 + (42-36)**2 + (10-8)**2)
Manhattan = abs(22-20) + abs(1-0) + abs(42-36) + abs(10-8)
Minkowski = ((abs(22-20)**3 + abs(1-0)**3 + abs(42-36)**3 + abs(10-8)**3))**(1/3)
supremum = max(abs(22-20) , abs(1-0) , abs(42-36) , abs(10-8))
print("Euclidean: ",Euclidean)
print("Manhattan: ",Manhattan)
print("Minkowski: ",Minkowski)
print("supremum: ",supremum)
```

```
Euclidean:  6.708203932499369
Manhattan:  11
Minkowski:  6.153449493663682
supremum:  6
```

2) Perform Preprocessing on Titanic Data set Using Orange Tools

3) Kindly Perform Data Exploration on New Restaurant Data Set

Link - https://github.com/guipsamora/pandas_exercises/blob/master/01_Getting_%26_Knowing_Your_Data/Chipotle/Exercises.ipynb

In [ ]:

PART - 2

In [4]:

```python
import pandas as pd
```

1) First, you need to read the titanic dataset from local disk and display Last five records

In [5]:

```python
df = pd.read_csv('titanic.csv')
```

In [6]:

```python
df.tail()
```

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

2) Handle Missing Values in data set [use dropna(), fillna(), and interpolate]

In [7]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [8]:

```
df.dropna()
```

Out[8]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

183 rows × 12 columns

In [9]:

```
# df.fillna({'Age': df['Age'].mean()}, inplace=True)
df.fillna({'Age': df['Age'].mean()})
```

Out[9]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.000000	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.000000	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.000000	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.000000	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.000000	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.000000	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.000000	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	29.699118	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.000000	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.000000	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [10]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [11]:

```
df['Age'] = df['Age'].interpolate(method='linear', limit_direction='forward')
```

In [12]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  891 non-null    int64  
 1   Survived     891 non-null    int64  
 2   Pclass       891 non-null    int64  
 3   Name         891 non-null    object 
 4   Sex          891 non-null    object 
 5   Age          891 non-null    float64
 6   SibSp        891 non-null    int64  
 7   Parch        891 non-null    int64  
 8   Ticket       891 non-null    object 
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object 
 11  Embarked     889 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

3) Write programs to perform the following tasks of preprocessing.

Equal Width Binning
Equal Frequency/Depth Binning

In [13]:

```
import pandas as pd
import numpy as np

data = [5, 10, 8, 2, 5, 6, 23, 18, 6, 9]

data_pandas = pd.DataFrame(data, columns=['Values'])
num_bin = 3

bin_edges = np.linspace(data_pandas['Values'].min(), data_pandas['Values'].max(), num_bin+1)
data_pandas['equal_width'] = pd.cut(data_pandas['Values'], bins=bin_edges, labels=False, include_lowest=True)
```

In [14]:

```
data = [5, 10, 8, 2, 5, 6, 23, 18, 6, 9, 48, 23]
num_of_bin = 3

seperator = len(data) / num_of_bin
for element in range(0, len(data), int(seperator)):
    print(data[element: element+int(seperator)])
```

```
[5, 10, 8, 2]
[5, 6, 23, 18]
[6, 9, 48, 23]
```

4) Apply Scaling to AGE attribute with min max, decimal scaling and z score.

In [18]:

```
df['Age_MinMax'] = (df['Age'] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())
print(df['Age_MinMax'])
```

```
0      0.271174
1      0.472229
2      0.321438
3      0.434531
4      0.434531
         ...
886    0.334004
887    0.233476
888    0.277457
889    0.321438
890    0.396833
Name: Age_MinMax, Length: 891, dtype: float64
```

In [19]:

```
max_abs_age = df['Age'].abs().max()
j = np.ceil(np.log10(max_abs_age + 1))
df['Age_Decimal'] = df['Age'] / (10 ** j)
print(df['Age_Decimal'])
```

```
0      0.220
1      0.380
2      0.260
3      0.350
4      0.350
       ...
886    0.270
887    0.190
888    0.225
889    0.260
890    0.320
Name: Age_Decimal, Length: 891, dtype: float64
```

In [20]:

```
df['Age_ZScore'] = (df['Age'] - df['Age'].mean()) / df['Age'].std()
print(df['Age_ZScore'])
```

```
0     -0.555738
1      0.595147
2     -0.268017
3      0.379356
4      0.379356
         ...
886   -0.196086
887   -0.771528
888   -0.519772
889   -0.268017
890    0.163565
Name: Age_ZScore, Length: 891, dtype: float64
```

In [ ]:

```

```