Darshan
UNIVERSITY
योगः कर्मसु कौशलम्

Data Mining

Lab - 4

Name - Ritesh Lakhani

Enrollment No - 22010101099

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 as m
x = (22,1,42,10)
y = (20,0,36,8)
sum = 0

#euclidean distance
for i in range(0,len(x)):
    sum += (x[i] - y[i])**2

print(f"Euclidean distance: {m.sqrt(sum)}")
```

```
Euclidean distance: 6.708203932499369
```

In [5]:

```python
x = (22, 1, 42, 10)
y = (20, 0, 36, 8)
sum = 0

# Manhattan distance
for i in range(0, len(x)):
    sum += abs(x[i] - y[i])

print(f"Manhattan distance: {sum}")
```

```
Manhattan distance: 11
```

In [6]:

```python
import math as m

x = (22,1,42,10)
y = (20,0,36,8)
p = 3
sum = 0

#Minkowski distance
for i in range(0,len(x)):
    sum += abs(x[i]-y[i])**p

distance = m.pow(sum,1/p)
print(f"Minkowski distance with p={p}: {distance}")
```

```
Minkowski distance with p=3: 6.153449493663682
```

In [7]:

```
x = (22,1,42,10)
y = (20,0,36,8)

max_diff = 0

for i in range(0,len(x)):
    diff = abs(x[i]-y[i])
    if diff > max_diff:
        max_diff = diff

print(f"Supremum distance:{max_diff}")
```

```
Supremum distance: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 [8]:

```
import pandas as pd
```

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

In [4]:

```
df = pd.read_csv("titanic.csv")
df
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

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

In [7]:

```
df.isnull().sum()
df.dropna()
```

Out[7]:

	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 [8]:

```
#check weather how many null values are present in Age column
print(df['Age'].isnull().value_counts())
print('-------------')
print(df.isnull().sum())
```

```
Age
False    714
True     177
Name: count, dtype: int64
-------------
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

In [16]:

```
df1=df.fillna({'Age':0})
df1
```

Out[16]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	0.0	1	2	W./C. 6607	23.4500	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

891 rows × 12 columns

In [17]:

```
df2=df1.fillna({'Cabin':'0','Embarked':'0'})
df2
```

Out[17]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	0	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	0	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	0	S
...	...	...	...	...	...	...	...	...	...	...	...	...
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	0	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	0.0	1	2	W./C. 6607	23.4500	0	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	0	Q

891 rows × 12 columns

In [18]:

```
df2.isnull().sum()
```

Out[18]:

```
PassengerId    0
Survived       0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```

In [20]:

```
df2 = df['Age'].interpolate(method='linear',limit_direction='backward')
df2
```

Out[20]:

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ... 
886    27.0
887    19.0
888    22.5
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [21]:

```
df3 = df['Age'].interpolate(method='linear',limit_direction='forward')
df3
```

Out[21]:

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ... 
886    27.0
887    19.0
888    22.5
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

In [26]:

```
df4 = df['Age'].interpolate(method='linear',limit_direction='both')
df4
```

Out[26]:

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ... 
886    27.0
887    19.0
888    22.5
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

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

Equal Width Binning

Equal Frequency/Depth Binning

In [ ]:

```
import pandas as pd
import numpy as np

data = [5,10,11,13,15,35,50,55,72,92,204,215]

df = pd.DataFram(data,columns=['Values'])

num_bins = 3

bin_edges = np.linspace(df['Value'].min() , df['Value'].max(), num_bins+1)
print(bin_edges)

df['Equal_Width']
```

In [33]:

```
data = [5,10,11,13,15,35,50,55,72,92,204,215]

num_bins = 3

no_of_data = len(data)
points_in_bin = no_of_data / num_bins

ans = []

for i in range(0,len(data),4):
    print(data[i:i+4])
```

```
[5, 10, 11, 13]
[15, 35, 50, 55]
[72, 92, 204, 215]
```

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

In [3]:

```
import pandas as pd
df = pd.read_csv("titanic.csv")
```

In [4]:

```
#get a maximum age
max = df['Age'].max()
max
```

Out[4]:

```
80.0
```

In [5]:

```
min = df['Age'].min()
min
```

Out[5]:

```
0.42
```

In [7]:

```
import pandas as pd

# Load the dataset
df = pd.read_csv("titanic.csv")

# Min-Max Scaling
min_age = df['Age'].min()
max_age = df['Age'].max()
df['Age_MinMax'] = (df['Age'] - min_age) / (max_age - min_age)

# Decimal Scaling
max_age_abs = df['Age'].abs().max()
j = len(str(int(max_age_abs)))
df['Age_Decimal'] = df['Age'] / (10 ** j)

# Z-Score Normalization
mean_age = df['Age'].mean()
std_age = df['Age'].std()
df['Age_ZScore'] = (df['Age'] - mean_age) / std_age

# Display the first few rows with the new columns
df.head()
```

Out[7]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	Age_MinMax	Age_Decimal
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0.271174	0.22
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C	0.472229	0.38
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0.321438	0.26
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0.434531	0.35
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0.434531	0.35

In [8]:

```python
import pandas as pd

# Load the dataset
df = pd.read_csv("titanic.csv")

# Min-Max Scaling
min_age = df['Age'].min()
max_age = df['Age'].max()
df['Age_MinMax'] = (df['Age'] - min_age) / (max_age - min_age)

# Decimal Scaling
max_age_abs = df['Age'].abs().max()
j = len(str(int(max_age_abs)))
df['Age_Decimal'] = df['Age'] / (10 ** j)

# Z-Score Normalization
mean_age = df['Age'].mean()
std_age = df['Age'].std()
df['Age_ZScore'] = (df['Age'] - mean_age) / std_age

# Calculate the correlation
correlation_matrix = df[['Age', 'Age_MinMax', 'Age_Decimal', 'Age_ZScore']].corr()

# Print the correlation matrix
print(correlation_matrix)
```

```
             Age  Age_MinMax  Age_Decimal  Age_ZScore
Age          1.0         1.0          1.0         1.0
Age_MinMax   1.0         1.0          1.0         1.0
Age_Decimal  1.0         1.0          1.0         1.0
Age_ZScore   1.0         1.0          1.0         1.0
```