

EXPERIMENT - 8

Question: Write a Program to Calculate and visualize a correlation matrix using pandas and seaborn.

Aim: The aim of this program is to calculate and visualize a correlation matrix using pandas and seaborn python library.

Procedure:

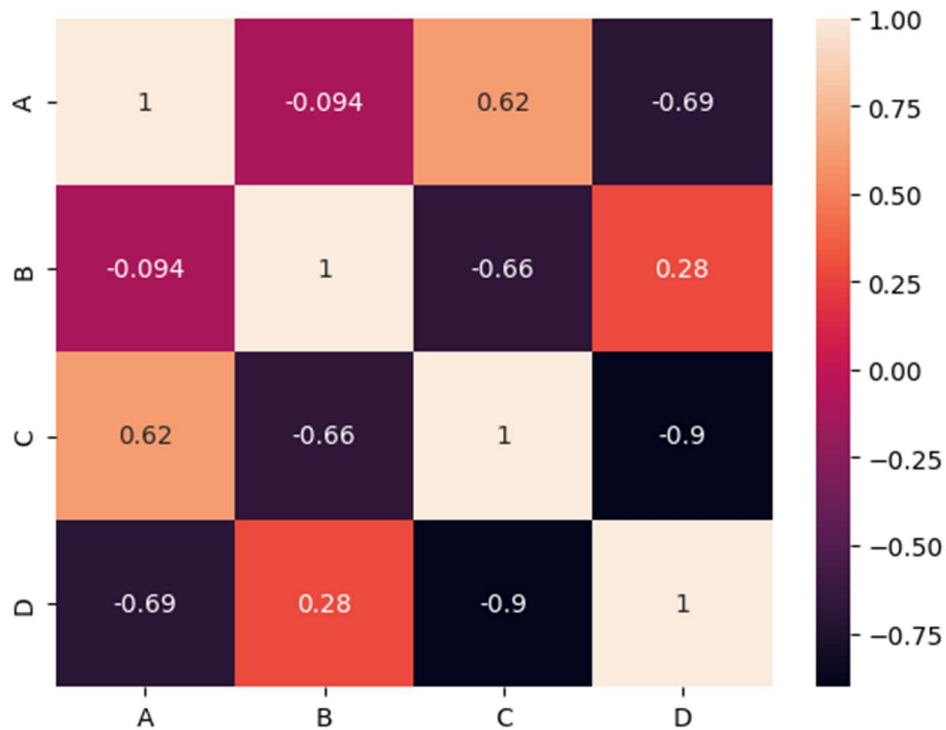
```
import pandas as pd
import seaborn as sns

data = {'A':[34,76,20,23],'B':[87,43,40,45],'C':[10,98,30,76],'D':[49,19,60,28]}

df = pd.DataFrame(data)

sns.heatmap(df.corr(),annot=True,)
```

Output:



Result: We successfully plot the correlation matrix using pandas and seaborn.

EXPERIMENT - 9

Question: Write a program to handle missing values using pandas (Data Cleaning).

Aim: The aim of this program is to handle (fill or drop) missing values using pandas.

Procedure:

```
import pandas as pd
```

```
data =
```

```
{'A':[34,76,20,23,None], 'B':[87,None,24,40,45], 'C':[10,98,30,76,34], 'D':[49,19,60,34,None]}
```

```
df = pd.DataFrame(data)
```

```
print(df)
```

Output (Original Dataframe):

	A	B	C	D
0	34.0	87.0	10	49.0
1	76.0	NaN	98	19.0
2	20.0	24.0	30	60.0
3	23.0	40.0	76	34.0
4	NaN	45.0	34	NaN

1. Remove Missing Values:

```
df_without_mv = df.dropna(axis=0)
```

```
df_without_mv
```

Output:

	A	B	C	D
0	34.0	87.0	10	49.0
2	20.0	24.0	30	60.0
3	23.0	40.0	76	34.0

2. Fill Missing Values:

Output:

```
df_filled_values = df.fillna(df.mean())
```

```
df_filled_values
```

	A	B	C	D
0	34.00	87.0	10	49.0
1	76.00	49.0	98	19.0
2	20.00	24.0	30	60.0
3	23.00	40.0	76	34.0
4	38.25	45.0	34	40.5

Result: We successfully handle the missing values using pandas

EXPERIMENT - 10

Question: Write a program to Normalize data using scikit-learn.

Aim: The aim of this program is to normalize data using python scikit-learn library.

Procedure:

```
import pandas as pd

from sklearn.preprocessing import MinMaxScaler

data = {'A':[34,76,20,23],'B':[87,43,40,45],'C':[10,98,30,76],'D':[49,19,60,28]}

df = pd.DataFrame(data)

scaler = MinMaxScaler()

scaled_data = scaler.fit_transform(df)

print('Original Data:\n',data)

print('Scaled Data:\n',scaled_data)
```

Output:

```
Original Data:
{'A': [34, 76, 20, 23], 'B': [87, 43, 40, 45], 'C': [10, 98, 30, 76],
'D': [49, 19, 60, 28]}

Scaled Data:
[[0.25      1.         0.         0.73170732]
 [1.         0.06382979 1.         0.         ]
 [0.         0.         0.22727273 1.         ]
 [0.05357143 0.10638298 0.75      0.2195122 ]]
```

Result: We successfully normalized the data using python scikit-learn library.

EXPERIMENT - 11

Question: Write a Program to Perform K-Means clustering using scikit-learn.

Aim: The aim of this program is to perform k-means clustering using scikit-learn.

Procedure:

```
from sklearn.cluster import KMeans  
  
kmean = KMeans(n_clusters=2)  
  
data = [[28],[76],[51],[48],[34],[40],[87],[67],[75],[39]]  
  
kmean.fit(data)  
  
pred = kmean.predict(data)  
  
print('Prediction: Cluster allotted to each data points are:\n',pred)
```

Output:

```
Prediction: Cluster allotted to each data points are:  
[0 1 0 0 0 0 1 1 1 0]
```

Result: We successfully executed the k-means clustering program, which divide each data points into cluster.

EXPERIMENT - 12

Question: Write a Program to implement a decision tree classifier using scikit-learn.

Aim: The aim of this program is to implement a decision tree classifier using python scikit-learn library.

Procedure:

```
from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()

X_train = [[28],[76],[51],[48],[34],[40],[87],[67],[75],[39]]

y_train = [[0],[1],[1],[0],[0],[0],[1],[1],[1],[0]]

dtree.fit(X_train,y_train)

# Testing Data

X_test = [[46],[54],[50],[42]]

pred = dtree.predict(X_test)

# Prediction

print('Prediction of Test Data:\n',pred)
```

Output:

```
Prediction of Test Data:
[0 1 1 0]
```

Result: We successfully implemented a decision tree classifier model, which divide the data points into classes.

EXPERIMENT - 13

Question: Write a Program to build a random forest classifier using scikit-learn.

Aim: The aim of this program is to build a random forest classifier using python scikit-learn library.

Procedure:

```
from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()

X_train = [[28],[76],[51],[48],[34],[40],[87],[67],[75],[39]]

y_train = [[0],[1],[1],[0],[0],[0],[1],[1],[1],[0]]

rfc.fit(X_train,y_train)

# Testing Data

X_test = [[46],[54],[50],[42]]

pred = rfc.predict(X_test)

# Prediction

print('Prediction of Test Data:\n',pred)
```

Output:

```
Prediction of Test Data:
[0 1 1 0]
```

Result: We successfully implemented a random forest classifier model, which divide the data points into classes.

EXPERIMENT - 14

Question: Write a Program to apply PCA for dimensionality reduction using scikit-learn.

Aim: The aim of this program is to apply PCA for reduce the data dimensions using python scikit-learn library.

Procedure:

```
from sklearn.decomposition import PCA

import numpy as np

pca = PCA(n_components=2)

data = np.array([[34,76,20,23],[87,43,40,45],[10,98,30,76],[49,19,60,28]])

transformed_data = pca.fit_transform(data)

print('Original Data:\n',data)

print('Data After Applying PCA:\n',transformed_data)
```

Output:

```
Original Data:
[[34 76 20 23]
 [87 43 40 45]
 [10 98 30 76]
 [49 19 60 28]]

Data After Applying PCA:
[[ 16.38102936 -10.97233447]
 [-35.47704737  27.66701872]
 [ 60.25710461   4.67481146]
 [-41.1610866  -21.36949571]]
```

Result: We successfully applied PCA to reduce the dimensions of the data using python scikit-learn library.

EXPERIMENT - 15

Question: Write a Program to create a time series plot using pandas and matplotlib.

Aim: The aim of this program is to create a time-series plot using python pandas and matplotlib library.

Procedure:

```
import pandas as pd

import matplotlib.pyplot as plt

data = {'Date': pd.date_range('2024-01-01', periods=7), 'Value': [90, 115, 80, 120, 85, 97, 150]}

df = pd.DataFrame(data)

plt.figure(figsize=(12,8))

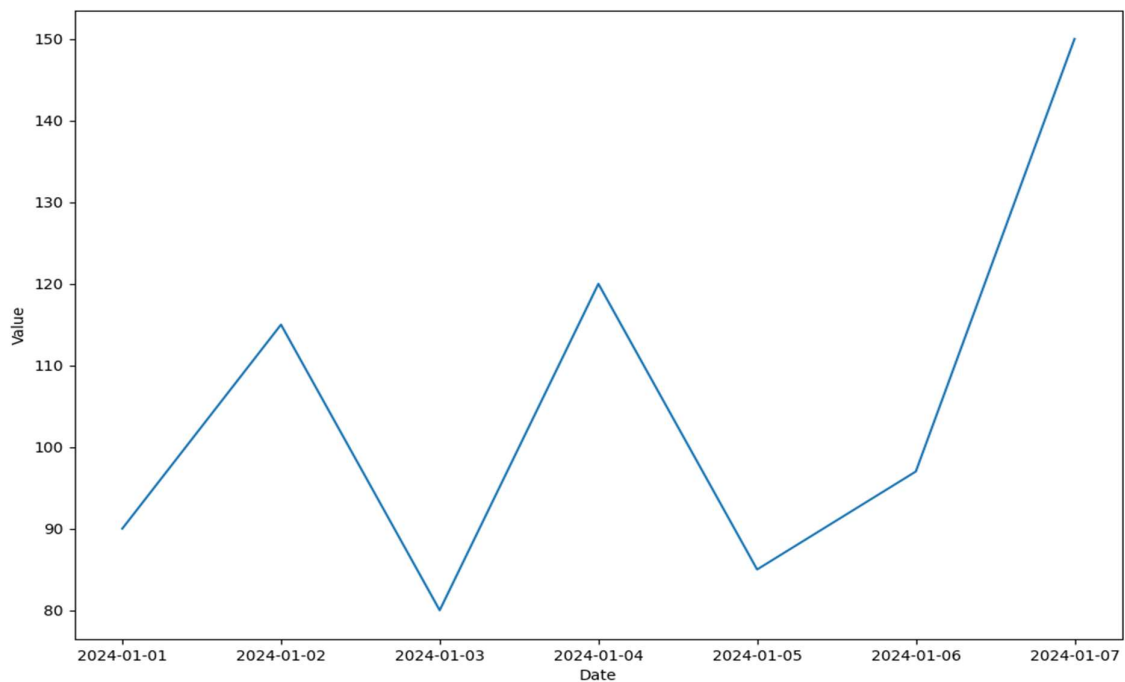
plt.plot(df['Date'], df['Value'])

plt.xlabel('Date')

plt.ylabel('Value')

plt.show()
```

Output:



Result: We successfully plot the time-series graph using python pandas and matplotlib library.

EXPERIMENT - 16

Question: Write a Program to load a dataset using pandas and Basic exploration of dataset properties.

Aim: The aim of this program is to load a dataset using python pandas library and do basic exploration of dataset.

Procedure:

```
import pandas as pd
```

```
df = pd.read_csv('customers-100.csv')
```

```
df.head()
```

Output 1 (Top 5 records of dataset):

	Index	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Email	Subscription Date	
0	1	DD37Cf93aecA6Dc	Sheryl	Baxter	Rasmussen Group	East Leonard	Chile	229.077.5154	397.884.0519x718	zunigavanessa@smith.info	2020-08-24	htt
1	2	1E7b82A4CAAD10	Preston	Lozano	Vega-Gentry	East Jimmychester	Djibouti	5153435776	686-620-1820x944	vmata@colon.com	2021-04-23	
2	3	6F94879bDAfE5a6	Roy	Berry	Murillo-Perry	Isabelborough	Antigua and Barbuda	+1-539-402-0259	(496)978-3969x58947	beckycarr@hogan.com	2020-03-25	
3	4	5Cef8BFA16c5e3c	Linda	Olsen	Dominguez, Mcmillan and Donovan	Bensonview	Dominican Republic	001-808-617-6467x12895	+1-813-324-8756	stanleyblackwell@benenson.org	2020-06-02	ht
4	5	053d585Ab6b3159	Joanna	Bender	Martin, Lang and Andrade	West Priscilla	Slovakia (Slovak Republic)	001-234-203-0635x76146	001-199-446-3860x3486	colinalvarado@miles.net	2021-04-17	

Dataset Information:

```
df.info()
```

Output 2 (Information of dataset):

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100 entries, 0 to 99
Data columns (total 12 columns):
#   Column                Non-Null Count  Dtype
---  ---
0   Index                  100 non-null   int64
1   Customer Id            100 non-null   object
2   First Name             100 non-null   object
3   Last Name              100 non-null   object
4   Company                100 non-null   object
5   City                   100 non-null   object
6   Country                100 non-null   object
7   Phone 1                100 non-null   object
8   Phone 2                100 non-null   object
9   Email                  100 non-null   object
10  Subscription Date      100 non-null   object
11  Website                100 non-null   object
dtypes: int64(1), object(11)
memory usage: 9.5+ KB
```

Dataset Description:

```
df.describe(include='object')
```

Output 3 (Description of dataset):

	Customer Id	First Name	Last Name	Company	City	Country	Phone 1	Phone 2	Email	Subscription Date
count	100	100	100	100	100	100	100	100	100	100
unique	100	93	97	99	100	85	100	100	100	96
top	DD37Cf93aecA6Dc	Sheryl	Greer	Simon LLC	East Leonard	Solomon Islands	229.077.5154	397.884.0519x718	zunigavanessa@smith.info	2021-12-02 http://www.stephen
freq	1	2	2	2	1	4	1	1	1	2

Result: We successfully load the dataset using python pandas library and do some basic exploration of dataset.