ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Lodha

Date..... Experiment No.....1..... P. No.....1.....

Name Pawan Kumar Meena Faculty No..... Class CSE..... Batch 2024

ASSIGNMENT : 1

Aim: Download, install and explore the features of Numpy, Scipy, Jupyter, Statsmodels and Pandas package.

(a) **Numpy:** Numpy is a fundamental package for scintific computing with Python. It provides support for large, multi-dimension arrays and matrice along with mathematical function to operate on these arrays:

Installation : 'pip install numpy'

(b) **Scipy:** Scipy builds on Numpy and provides additional modules for optimization, integration, interpolation, eigenvalue problem and more.

Installation: 'pip install scipy'

M. B. M. UNIVERSITY JODHPUR

Lodha

Date..... Experiment No..... P. No.....2.....

Name..... Faculty No..... Class..... Batch.....

Code (Numpy and SciPy):

```
# NumPy Example
import numpy as np
arr = np.array([1,2,3,4,5])
print("Numpy Array:", arr)

# SciPy Example:

from scipy.optimize import minimize
def objective_function(x):
    return x**2 + 5*x + 6
result = minimize(objective_function, x0=0)
print("Optimial Solution", result.x)
```

Output:

```
Numpy Array : [1 2 3 4 5]
Optimal Solution : [-2.50000002]
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No. 3

Name..... Faculty No..... Class..... Batch.....

Lodha

(c.) **Jupyter:**

Jupyter is an open-source web applications that allows you to create and share documents containing live code, equations, visualizations, and narrative text.

Installation: 'pip install jupyter'

(d.) **Statsmodels:** Statsmodels is a library for estimating and testing statistical models. It provides classes and functions for a wide range of statistical models.

Installation: 'pip install statsmodels'

Source Code:

```
# Statsmodels Example
import statsmodels.api as sm
import pandas as pd

# load sample dataset
data = sm.datasets.get_rdataset("mtcars").data
```

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No. 4.

Name..... Faculty No..... Class..... Batch.....

Lodha

Fit linear regression model

```
model = sm.OLS(data["mpg"], sm.add_constant(data["wt"])).fit()
print("Linear Regression Results:")
print(model.summary())
```

(e.) **Pandas:** Pandas is a powerful library for data manipulations and analysis. It provides data structure like Series and DataFrame for efficient data cleaning, exploration and analysis

Installation: `pip install pandas`

Source Code:

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie'],
        'Age': [25, 30, 22],
        'City': ['New York', 'San Francisco', 'Los Angeles']}
pd = pd.DataFrame(data)
```

M. B. M. UNIVERSITY JODHPUR

Lodha

Date..... Experiment No..... P. No. 5

Name..... Faculty No..... Class..... Batch.....

```
#Display DataFrame
print ("DataFrame:", df)
print ("\n Filtered Data:")
print (df[df['Age'] >25])
```

Output:

Filtered Data:

	Name	Age	City
1	Bob	30	SanFranisco

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Lodha

Date..... Experiment No.....2..... P. No.....6.....
Name.Pawan....Kumar..... Faculty No..... Class..CSE..... Batch..2024

ASSIGNMENT: 2

Aim: Working with NumPy arrays.

Algarithim:

Step 1: Start the Porogram
Step 2: Import the NumPy Library
Step 3: Define 1D, 2D, 3D array.
Step 4: Create arrays using built-in NumPy Functions.
Step 5: Perform file operations with NumPy arrays.
Step 6: Display the output
Step 7: Stop the Porogram

Porogram:

(i) Creating diff. types of NumPy Arrays and Displaying Basic information.

Output: (i)

1D Array: [1 2 3 4 5]

Data Type: int64

Shape : (5,)

Size : 5

Strides : (8,)

2D Array:

[[1 2 3]
[4 5 6]]

Data Type: int64

Shape : (2,3)

Size : 6

Strides : (24, 8)

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....7.....

Name..... Faculty No..... Class..... Batch.....

Lodha

```
impart numpy as np

#create 1D array:
array_1D = np.array([1,2,3,4,5])
print("1D Array:", array_1D)
print("Data Type:", array_1D.dtype)
print("Shape:", array_1D.shape)
print("Size:", array_1D.size)
print("Strides:", array_1D.strides)

# create a 2D array

arr_2d = np.array([[1,2,3],[4,5,6]])
print("in 2D array:", arr_2d)
print("Data Type:", arr_2d.dtype)
print("Shape:", arr_2d.shape)
print("Size:", arr_2d.size)
print("Strides:", arr_2d.strides)
```

Output -ii

Zeros Arrays:

```
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
```

Ones Arrays:

```
[[1 1 1]
 [1 1 1]]
```

Identity Matrix:

```
[[1 0 0]
 [0 1 0]
 [0 0 1]]
```

Output(iii)

loaded Array from file:

```
[[1 2 3]
 [4 5 6]]
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

Lodha

(ii) Creating an Arrays using Built-in NumPy Functions.

```
# Create an array with zeros
zero_arr = np.zeros((3,4))
print("\n Zeros Array :", zero_arr)

# Create an array with ones
ones_arr = np.ones((2,3))
print("\n Ones array :", ones_arr)

# Create identity matrix
iden_mat = np.eye(3)
print("\n Identity Matrix", iden_mat)
```

(iii) Performing File operations with Numpy Arrays

```
np.save('array_file.npy', arr_2d)

# load array from file
load_arr = np.load('array_file.npy')
print("\n Loaded Array from file:")
print(load_arr)
```

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No. 3 P. No. 9

Name Pawan Ku. Meena Faculty No..... Class. CSE Batch.....

Lodha

ASSIGNMENT : 03

Aim: Working with Pandas Data Frame

Algorithm:

(i) Start the Program
(ii) Import Numpy and Pandas Packages
(iii) Create dataframe for the list of element (number, dictionary, n-D arrays)
(iv) Load dataset from external source into a Pandas dataframe
(v) Display the O/P
(vi) Stop the Program

Program:

(i) Creating a DataFrame from a Series:

```
import Pandas as pd
Series_data = pd.Series([1,2,3,4,5], name='Column name')
df_from_series = pd.DataFrame(Series_data)
print("(i) DataFrame From Series")
print(df_from_series)
```

Output: (i) DataFrame From Series

	Column_Name
0	1
1	2
2	3
3	4
4	5

(ii) DataFrame From Dictionary:

	Column1	Column2
0	1	A
1	2	B
2	3	C

(iii) DataFrame From N-d arrays:

	Column1	Column2	Column3
0	1	2	3
1	4	5	6

(iv) DataFrame From External Source:

#Display Iris D.F.

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No. 10

Name..... Faculty No..... Class..... Batch.....

Lodha

(ii) Creating DataFrame from Dictionary

```
dict_data = {'Column1': [1,2,3], 'Column2':
                                  ['A','B','C']}

df_from_dct = pd.DataFrame(dict_data)
print("\n(ii) DataFrame From Dictionary")
print(df_from_dct)
```

(iii) Creating D.F. from N-D arrays

```
import numpy as np
array_data = np.array([[1,2,3],[4,5,6]])

df_from_array = Pd.DataFrame(array_data,
                columns=['Column1', 'Column2', 'Column3'])
print("\n(iii) DataFrame from N-D array")
print(df_from_array)
```

(iv) Loading Dataset from external source

```
url = 'https://archive.ics.uci.edu/ml/machine-learning-
                    dataset/iris/iris.data'
df_from_external = pd.read_csv(url)
print(df_from_external)
```

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No.. 4. P. No.. 11

Name Pawan Ku. Meena Faculty No..... Class.. CSE Batch.. 2024

Lodha

ASSIGNMENT: 04

Aim: Reading data from text Files, Excel and the web and explaring various command for doing descriptive analytics on the iris dataset ~~etc~~.

Algorithm:

(i) read data from txt file using Pandas Package.
(ii) " " data from excel file using " "
(iii) read data from html web file " "
(iv) Load the iris dataset using seaborn.
(v) Perform Descriptive statistics (count, mean, median, min, max)
(vi) Create Histograms, Box Plot, and Correlation matrix

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....12.....

Name..... Faculty No..... Class..... Batch.....

Lodha

Program:

```
text_file = 'iris.txt' # read text file
df_text = pd.read_csv(text_file)

excel_file = 'iris.xls' # read excel file
df_excel = pd.read_excel(excel_file)

web_url = 'https://archive.ics.uci.edu/ml/
        machine-learning-databases/iris/iris.data'
df_web = pd.read_csv(web_url)

#Display first few rows of each dataframe

print("\n DataFrame from text file")
print(df_text.head())

print("\n DataFrame from excel file")
print(df_excel.head())

print("\n DataFrame from web:")
print(df_web.head())
```

Output:

Summery Statistics:

	Sepal-length	Sepal-width	Petal-length	Petal width
Count	150.000	150.000	150.000	150.000
mean	5.8433	3.0573	3.7580	1.1993
Std.	0.8280	0.4358	1.7652	0.7622
min	4.3000	2.0000	1.0000	0.1000
25%	5.1000	2.8000	1.6000	0.3000
50%	5.8000	3.0000	4.3500	1.3000
75%	6.4000	3.3000	5.1000	1.8000
max	7.9000	4.4000	6.9000	2.5000

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No.....

Name..... Faculty No..... P. No. 13.

Class..... Batch.....

Lodha

```
import seaborn as sns

iris_df = sns.load_dataset('iris')
~~iris~~
# Descriptive Analytics:

summery_statistics = iris_df.describe()
print("\n Summary Statistics:")
print( summery_statistics)

# Visualization
import matplotlib.pyplot as plt
import seaborn as sns

# Histograms

iris_df.hist(figsize=(8,6))
plt.suptitle("Histograms of Iris Dataset")
plt.show()
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No.....

Name..... Faculty No..... P. No. 14

Class..... Batch.....

```
# Box Plot

sns.boxplot(data=iris_df, x='species', y='sepal_length')
plt.title("Box plot of Sepal length by Species")
plt.show()

# Correlation matrix
correlation_matrix = iris_df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt=".2f")
plt.show()
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No.....5..... P. No.....

Name.Pawan Ku. Meena Faculty No..... Class....CSE.... Batch....2024

Lodha

ASSIGNMENT: 05

Aim: Use the diabetes dataset from UCI and Pima Indians Diabetes dataset for performing the following.

(a.) Univeriate Analysis
(b.) Bivariate Analysis: Linear and ~~de~~ logistic regression modeling

(c.) Multiple Regression analysis
(d.) Compare result of both dataset.

Algorithm:

(I) Download UCI and Pima INDians diabetes data using Keggle.
(II) Convert UCI & Pima dataset to Dataframe
(III) Perform the given operations on Datasets

Output:

Univariate Analysis – UCI

Age

Count	520.000
mean	48.028
Std	12.1514
min	16.000
25%	39.000
50%	47.5000
75%	57.000
max	90.000

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

Lodha

Puroguram:

(a.) Univariate analysis:

```
impart pandas as pd
impart statsmodels.api as sm
impart seaborn as sns

uci_df = pd.read_csv("uci_diabetes.csv")
pima_df = pd.read_csv("pima_diabetes.csv")

pima_df.describe()

uci_df.describe()
```

(b.) Bivariate Analysis: Linear and Regression modeling.

```
impart pandas as Pd
impart statsmodels.api as Sm
impart Seaborn as Sns
impart matplotlib.pyplot as plt
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

#Linear Regression modeling

```
X_linear_pima = sm.add_constant(pima_diabetes_df
                > ['Age', 'Glucose'])
y_linear_pima = pima_diabetes_df['Outcome']
linear_model_pima = sm.OLS(y_linear_pima,
                    X_linear_pima).fit()
```

#Logistic Regression modeling

```
X_logistic_pima = pima_df['Age', 'Glucose']
y_logistic_pima = pima_df['Outcome']

X_train_pima, X_test_pima, y_train_pima, y_test_pim
= train_test_split(X_logistic_pima, y_logistic_
pima, test_size = 0.2, random_state = 42)

logistic_model_pima = LogisticRegression()
logistic_model_pima.fit(X_train_pima, y_train_
                        pima)

y_pred_pima = logistic_model_pima.predict
              (X_test_pima)
```

ENGCOL Con. Co-op. Stores Ltd., Jodhpur

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

Lodha

```
plt.figure(figsize=(8,6))

sns.scatterplot(x='Age', y='Glucose', hue='Outcome', data=pima_diabetes_df)

plt.plot(x_linear_pima['Age'], linear_model_pima.predict(x_linear_pima), color='red',
label='Linear Regression Line')
plt.show()

plt.figure(figsize=(8,6))

sns.scatterplt(x='Age', y='Glucose', hue='Outcome', data=pima_df)

#Logistic regression
x_values = x_test_pima['Age'].sort_values()
y_probabilities = logistic_model_pima.predict_proba(x_test_pima)[:,1]
plt.plot(x_values, y_probabilities, color='green',
label='Logistic Regression curve')
plt.show()
```

M. B. M. UNIVERSITY JODHPUR

Lodha

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

#Display logistic regression results

```
print ("Logistic regression for Pima Dataset")
print ("Confusion matrix:")
print (confusion_matrix(y_test_pima, y_pred_pima))
print ("Classification Report:")
print (classification_report(y_test_pima, y_pred_pima))
```

(c) Multiple Regression Analysis

```
X_multi = sm.add_constant(pima_df[['Age', 'Glucose', 'BMI', 'BloodPressure']])
y_multi = pima_df['Outcome']

X_train_multi, X_test_multi, y_train_multi, y_test_multi
= train_test_split(X_multi, y_multi,
    test_size=0.2, random_state=42)
```

M. B. M. UNIVERSITY JODHPUR

Date..... Experiment No..... P. No.....

Name..... Faculty No..... Class..... Batch.....

```
print ("Multiple Regression Analysis")
multi_model = sm.OLS(y_multi, X_multi).fit()
print(multi_model_pima.summary())

# Confusion Matrix Plot
plt.figure(figsize=(6,4))
sns.heatmap(confusion_matrix(y_test_multi,
        y_pred_multi), annot=True, fmt='d',
            cmap='Blues')
plt.title('Confusion Matrix')
plt.show()

# Roc curve
y_prob_multi = logistic_model_multi.predict_proba
        (X_test_multi)[:,1]
fpr, tpr, threshold = roc_curve(y_test_multi,
        y_prob_multi)

plt.figure(figsize=(6,4))
plt.plot(tpr, fpr, color='darkorange', lw=2
        label='ROC Curve')
plt.show()
```

M.B.M. UNIVERSITY
JODHPUR

Date..... Experiment No.....6..... P.No.....

Name Pawan Kumar... Faculty No..... Class....C.S.E.....

ASSIGNMENT : 06

Aim: Apply and explore various plotting functions on UCI datasets

Program:

(a.) NORMAL CURVE

```
import seaborn as sns
flight = sns.load_dataset('flights')
flight.head()
may_flight = flight.query("month=='May'")
sns.lineplot(data=may_flight, x='year',
             y='passengers')
```

(b.) Density And Constour Plots

```
iris = sns.load_dataset('iris')
sns.kdeplot(data=iris)
```

(c.) Correlation and Scatter Plots

Correlation

```
df = Sns.load_dataset('titanic')
ax = Sns.heatmap(df.annot=True, fmt='d')
```

Scatter

```
df = Sns.load_dataset('titanic')
Sns.catplot(data=df, x='age', y='class')
```

(d) Histogram:

```
df = Sns.load_dataset('titanic')
Sns.histplot(data=df, x='age')
```

e) Three Dimensional Plotting

```
import plotly as px
df = Sns.load_dataset('iris')
px.scatter_3d(df, x='PetelLengthCm', y=
'PetelwidthCm', z='SepalWidthCm', size=
'SepalLengthCm',
color='Species', color_discrete_map={'Joly':'blue',
'Bergeron':'Violet', 'Coderre':'Pink'})
```

M.B.M. UNIVERSITY
JODHPUR

Date..... Experiment No. 7 ..... P.No.....

Name Pawan Kumar Faculty No..... Class. C.S.E

ASSIGNMENT : 07

Aim: Visualizing Geographic Data with Basemap.

Program:

(i)
```
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.basemap import Basemap
plt.figure(figsize=(8,8))
m = Basemap(projection='ortho', resolution=
    None, lat_0=50, lon_0=-100)
m.bluemarble(scale=0.5)
```

(ii)
```
fig = plt.figure(figsize=(8,8))
m = Basemap(projection='lcc', resolution=None,
    width=8E6, height=8E6, lat_0=45,
    lon_0=-100)
m.etopo(scale=0.5, alpha=0.5)
```

Date..... Experiment No..... P.No.....

Name ..... Faculty No..... Class.....

#map (long, lat) to (x,y) for plotting

x,y = m(-122.3, 47.6)

plt.plot(x,y, 'ok', markersize = 5)

plt.text(x,y, 'Seattle', fontsize=12);

(iii) fig = plt.figure(figsize=(8,8))

m = Basemap(projection='lcc',

resolution = None, lon_0=0, lat_0=50,

lat_1=45, lat_2=55, width = 1.6E7,

height = 1.2E7)

draw_map(m)