# LAB # 2

## DATASET PREPARATION WITH EXCEL SPREADSHEET AND DATASET WITH PREPROCESSING AND SCALING TECHNIQUES

**LAB TASKS:**

1. **Write a python code to load an excel spreadsheet containing two different sheets and print both of them**

```
[18]: import pandas as pd
      import numpy as np

      sheet1 = pd.read_excel("DATASHEETLAB2_C.xlsx", sheet_name='Sheet1')
      sheet2 = pd.read_excel("DATASHEETLAB2_C.xlsx", sheet_name="Sheet2")

      print("Contents of Sheet 1: \n")
      print(sheet1)
      print("Contents of Sheet 2: \n")
      print(sheet2)
```

```
Contents of Sheet 1:

     NAME  AGE    STREAM  PERCENTAGE
0     ALI   30      MATH          65
1  FAIZAN   39   SCIENCE          90
2   ABRAR   44  COMMERCE          75
3  HAREEM   23      MATH          85
4  KASHIF   21   SCIENCE          70
Contents of Sheet 2:

   Math  English  Urdu  Itc  Total result
0    85       80    78   79    322   PASS
1    90       50    70   65    275   PASS
2    80       60    40   50    230   FAIL
3    25       60    30   80    195   FAIL
4    55       35    90   50    230   FAIL
5    85       40    70   60    255   PASS
```

2. **Write a Python code to generate a pandas data frame having 4 columns and 5 rows. Column 1 must contain the index values like Ali, Amir, Kamran, etc., and Row 1 must include the subject names.**

```
[21]: import pandas as pd

data = {
    "Math" : [80, 20, 45, 50, 85],
    "History" : [50, 40, 37, 37, 54],
    "Science" : [80, 98, 95, 92, 90],
    "Computer" : [32, 89, 91, 93, 93],
}

index_values = ["Ali", "Amir", "Kamran", "Zaid", "Sarah"]

df = pd.DataFrame(data, index=index_values)

print(df)

        Math  History  Science  Computer
Ali       80       50       80        32
Amir      20       40       98        89
Kamran    45       37       95        91
Zaid      50       37       92        93
Sarah     85       54       90        93
```

3. **Write a Python code to read an Excel spreadsheet and only print the first two columns using pandas data frame.**

```
[24]: import pandas as pd

excel_file = "C:\\Users\\ABC\\Desktop\\DATASHEETLAB2_C.xlsx"

df = pd.read_excel(excel_file)

print(df.iloc[:, :2])

     NAME  AGE
0     ALI   30
1  FAIZAN   39
2  ABRAR   44
3  HAREEM   23
4  KASHIF   21

[ ]:
```

4. **Write a Python code to skip the first two rows of an Excel spreadsheet and print the output using a pandas data frame.**

```
[40]: import pandas as pd

      excel_file = "C:\\Users\\ABC\\Desktop\\DATASHEETLAB2_C.xlsx"

      df = pd.read_excel(excel_file)

      print(df.iloc[2:,:])

         NAME  AGE    STREAM  PERCENTAGE
2   ABRAR   44  COMMERCE          75
3  HAREEM   23      MATH          85
4  KASHIF   21   SCIENCE          70

[ ]:
```

5. **Write a Python code to fill all the null values in the Gender column of employees.csv with "No Gender". Print the first 10 to 30 rows of the data frame for visualization.**

C: > Users > Shaikh > Desktop > LEARNING PYTHON > employees.csv > data

```
EmployeeID,FirstName,LastName,Age,Gender,Salary
1,John,Doe,30,,50000
2,Jane,Smith,25,Female,45000
3,Michael,Johnson,35,Male,60000
4,Emily,Brown,28,Female,55000
5,David,Davis,32,,
6,Sarah,Miller,27,,
7,Robert,Wilson,38,Male,65000
8,Maria,Anderson,31,Female,52000
9,James,Thomas,33,,
10,Jennifer,Lee,29,Female,48000
```

AI_lab2.py × employees.csv

> Users > Shaikh > Desktop > LEARNING PYTHON > AI_lab2.py > ...

```
import pandas as pd

# Read the CSV file
df = pd.read_csv("C:\\Users\\Shaikh\\Desktop\\employees.csv.txt")

# Fill null values in 'Gender' column with 'No Gender'
df['Gender'].fillna('No Gender', inplace=True)

# Print the first 10 rows of the DataFrame
print(df.head(10))
```

```
df['Gender'].fillna('No Gender', inplace=True)
   EmployeeID FirstName  LastName  Age     Gender   Salary
0           1      John       Doe   30  No Gender  50000.0
1           2      Jane     Smith   25     Female  45000.0
2           3   Michael   Johnson   35       Male  60000.0
3           4     Emily     Brown   28     Female  55000.0
4           5     David     Davis   32  No Gender      NaN
5           6     Sarah    Miller   27  No Gender      NaN
6           7    Robert    Wilson   38       Male  65000.0
7           8     Maria  Anderson   31     Female  52000.0
8           9     James    Thomas   33  No Gender      NaN
9          10  Jennifer       Lee   29     Female  48000.0
PS C:\Users\Shaikh>
```

6. **Write a Python code to scale the values of features (Age and Salary) using the Min-Max Normalization technique. Verify your answers by applying the formula mentioned above.**

```python
import pandas as pd
from sklearn import preprocessing

# Create a Pandas DataFrame from the data
data = {'Age': [25, 36, 30, 27, 38, 42, 34],
        'Salary': [42000, 50000, 45000, 43000, 51000, 62000, 48000]}
df = pd.DataFrame(data)

# Create a MinMaxScaler object
scaler = preprocessing.MinMaxScaler(feature_range=(0, 1))

scaler.fit(df)

scaled_data = scaler.transform(df)

# Create a new DataFrame with the scaled values
scaled_df = pd.DataFrame(scaled_data, columns=['Age', 'Salary'])

print(f"{scaled_df} \n")
```

```
C:\Users\Shaikh\AppDat
          Age  Salary
0  0.000000    0.00
1  0.647059    0.40
2  0.294118    0.15
3  0.117647    0.05
4  0.764706    0.45
5  1.000000    1.00
6  0.529412    0.30
```

```python
# Calculating the scaled values manually using the formula
for i in range(len(df)):
    scaled_age = (df['Age'][i] - df['Age'].min()) / (df['Age'].max() - df['Age'].min())
    scaled_salary = (df['Salary'][i] - df['Salary'].min()) / (df['Salary'].max() - df['Salary'].min())
    print(f"Age: {scaled_age: .4f}, Salary: {scaled_salary: .4f}")
```

```
Age:  0.0000, Salary:  0.0000
Age:  0.6471, Salary:  0.4000
Age:  0.2941, Salary:  0.1500
Age:  0.1176, Salary:  0.0500
Age:  0.7647, Salary:  0.4500
Age:  1.0000, Salary:  1.0000
Age:  0.5294, Salary:  0.3000
```

7. **Write a Python code to scale the values of features (Age and Salary) using the Standardization technique. Verify your answers by applying the formula mentioned above.**

| Age | Salary |
|---|---|
| 25 | 42000 |
| 36 | 50000 |
| 30 | 45000 |
| 27 | 43000 |
| 38 | 51000 |
| 42 | 62000 |
| 34 | 48000 |

```python
import pandas as pd
from sklearn import preprocessing

# Create a Pandas DataFrame from the data
data = {'Age': [25, 36, 30, 27, 38, 42, 34],
        'Salary': [42000, 50000, 45000, 43000, 51000, 62000, 48000]}
df = pd.DataFrame(data)

# Create a StandardScaler object
standard = preprocessing.StandardScaler()

standard.fit(df)

Xnew = standard.transform(df)

# Create a new DataFrame with the scaled values
scaled_df = pd.DataFrame(Xnew, columns=['Age', 'Salary'])

print(f"{scaled_df} \n")
```

```
C:\Users\Shaikh\AppData\Local\P
         Age    Salary
0 -1.436721 -1.070396
1  0.504113  0.204969
2 -0.554524 -0.592134
3 -1.083842 -0.910975
4  0.856992  0.364390
5  1.562749  2.118017
6  0.151234 -0.113872
```

```python
# Calculating the scaled values manually using the formula
for i in range(len(df)):
    scaled_age = (df['Age'][i] - df['Age'].mean()) / df['Age'].std()
    scaled_salary = (df['Salary'][i] - df['Salary'].mean()) / df['Salary'].std()
    print(f"Age: {scaled_age: .4f}, Salary: {scaled_salary: .4f}")
```

```
Age: -1.3301, Salary: -0.9910
Age:  0.4667, Salary:  0.1898
Age: -0.5134, Salary: -0.5482
Age: -1.0034, Salary: -0.8434
Age:  0.7934, Salary:  0.3374
Age:  1.4468, Salary:  1.9609
Age:  0.1400, Salary: -0.1054
```

**8. Given this dictionary, create a dataframe from dictionary and interpolate the missing values using backward interpolation. Hint: use interpolate().**

C: > Users > Shaikh > Desktop > LEARNING PYTHON > AI_lab2.py > ...

```python
import numpy as np
import pandas as pd

# Given dictionary
data_dict = {
    'First Score': [100, 90, np.nan, 95],
    'Second Score': [30, 45, 56, np.nan],
    'Third Score': [np.nan, 40, 80, 98]
}

# Create a DataFrame from the dictionary
df = pd.DataFrame(data_dict)

print("\nOriginal DataFrame:")
print(df)

# Interpolate missing values using backward interpolation
df_interpolated = df.interpolate().bfill()

print("\nDataFrame after backward interpolation:")
print(df_interpolated)
```

```
Original DataFrame:
   First Score  Second Score  Third Score
0        100.0          30.0          NaN
1         90.0          45.0         40.0
2          NaN          56.0         80.0
3         95.0           NaN         98.0

DataFrame after backward interpolation:
   First Score  Second Score  Third Score
0        100.0          30.0         40.0
1         90.0          45.0         40.0
2         92.5          56.0         80.0
3         95.0          56.0         98.0
PS C:\Users\Shaikh>
```