LAB # 2

DATASET PREPARATION WITH EXCEL SPREADSHEET AND DATASET 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.

Code:-

```
import pandas as pd

excel_file = 'Example.xlsx'

sheet1 = pd.read_excel(excel_file, sheet_name=0)  # First sheet
sheet2 = pd.read_excel(excel_file, sheet_name=1)  # Second sheet

print("------ Sheet 1 ------")
print(sheet1)
print("\n------ Sheet 2 ------")
print(sheet2)
```

Output:-

```
------ 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

------ Sheet 2 ------
     Name  Age    Stream  Percentage
0   Amina   28   Science          88
1   Bilal   35  Commerce          78
2   Dania   42      Math          92
3  Hassan   25   Science          80
4    Iqra   22  Commerce          67
```

2. Write a python cade 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 contain the subject names.

Code:-

```
import pandas as pd

data = {
    'Math': [85, 78, 92, 88, 76],
    'English': [79, 85, 90, 82, 88],
    'Science': [91, 80, 87, 89, 84],
    'History': [76, 89, 84, 90, 82]
}

index = ['Ali', 'Amir', 'Kamran', 'Sara', 'Hina']

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

print(df)
```

Output:-

```
        Math  English  Science  History
Ali       85       79       91       76
Amir      78       85       80       89
Kamran    92       90       87       84
Sara      88       82       89       90
Hina      76       88       84       82
```

3. Write a python code to read an excel spreadsheet and only print first two columns using pandas data frame.

Code:-

```
import pandas as pd

df = pd.read_excel('Example.xlsx')

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

Output:-

```
     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 excel spreadsheet and print the output using pandas data frame.

Code:-

```
import pandas as pd

df = pd.read_excel('Example.xlsx',skiprows=2)

print(df)
```

Output:-

```
    Faizan  39   Science  90
0   Abrar   44  Commerce  75
1  Hareem   23      Math  85
2  Kashif   21   Science  70
```

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

Code:-

```
import pandas as pd

df = pd.read_csv('employees.csv')

df['Gender'] = df['Gender'].fillna('No Gender')

print(df)
```

Output:-

```
   EmployeeID    Name  Age     Gender Department  Salary
0         101     Ali   28       Male    Finance   55000
1         102    Amir   32  No Gender         IT   50000
2         103   Sarah   26     Female         HR   60000
3         104  Kamran   29       Male  Marketing   50000
4         105    Hina   31  No Gender    Finance   60000
```

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

Code:-

```
import pandas as pd

df = pd.read_csv('employees.csv')

cols_to_scale = ['Age', 'Salary']

df_scaled = df.copy()
for col in cols_to_scale:
    min_val = df[col].min()
    max_val = df[col].max()
    df_scaled[col] = (df[col] - min_val) / (max_val - min_val)

print("Original Data:")
print(df[['Name', 'Age', 'Salary']])
print("\nNormalized Data (Min-Max Scaled):")
print(df_scaled[['Name', 'Age', 'Salary']])
```

Output:-

```
Original Data:
     Name  Age  Salary
0     Ali   28   55000
1    Amir   32   50000
2   Sarah   26   60000
3  Kamran   29   50000
4    Hina   31   60000

Normalized Data (Min-Max Scaled):
     Name       Age  Salary
0     Ali  0.333333     0.5
1    Amir  1.000000     0.0
2   Sarah  0.000000     1.0
3  Kamran  0.500000     0.0
4    Hina  0.833333     1.0
```

7. Write a python code to scale the values of features (Age and Salary) using 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

Code:-

```python
import pandas as pd
data = {
    'Name': ['Ali', 'Amir', 'Sarah', 'Kamran', 'Hina', 'Rayyan', 'Wania'],
    'Age': [25, 36, 30, 27, 38, 42, 34],
    'Salary': [42000, 50000, 45000, 43000, 51000, 62000, 48000]
}

df = pd.DataFrame(data)

features = ['Age', 'Salary']

df_standardized = df.copy()

for feature in features:
    mean = df[feature].mean()
    std = df[feature].std()
    df_standardized[feature] = (df[feature] - mean) / std
    print(f"\nStandardizing {feature}:")
    print(f"Mean = {mean}, Std Dev = {std}")
    print(df_standardized[[feature]])

# Display
print("\nOriginal Data:")
print(df)
print("\nStandardized Data:")
print(df_standardized)
```

Output:-

```
Standardizing Age:
Mean = 33.142857142857146, Std Dev = 6.12178000880385
        Age
0 -1.330145
1  0.466718
2 -0.513389
3 -1.003443
4  0.793420
5  1.446825
6  0.140015

Standardizing Salary:
Mean = 48714.28571428572, Std Dev = 6775.305299745681
      Salary
0 -0.990994
1  0.189765
2 -0.548209
3 -0.843399
4  0.337360
5  1.960903
6 -0.105425

Original Data:
      Name  Age  Salary
0      Ali   25   42000
1     Amir   36   50000
2    Sarah   30   45000
3   Kamran   27   43000
4     Hina   38   51000
5   Rayyan   42   62000
6    Wania   34   48000

Standardized Data:
      Name       Age    Salary
0      Ali -1.330145 -0.990994
1     Amir  0.466718  0.189765
2    Sarah -0.513389 -0.548209
3   Kamran -1.003443 -0.843399
4     Hina  0.793420  0.337360
5   Rayyan  1.446825  1.960903
6    Wania  0.140015 -0.105425
```

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

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

Code:-

```
import pandas as pd
import numpy as np

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

# Create DataFrame
df = pd.DataFrame(data)

# Interpolate missing values using backward interpolation
df_interpolated = df.interpolate(method='linear', limit_direction='backward')

print("Original DataFrame:\n", df)
print("\nBackward Interpolated DataFrame:\n", df_interpolated)
```

Output:-

```
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

Backward Interpolated DataFrame:
    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           NaN         98.0
```