# DELHI TECHNOLOGICAL UNIVERSITY


**IT-106**

**PRACTICAL FILE**

# OPEN SOURCE PROGRAMMING

**Submitted By:**

Vaibhav Jain
23/IT/168
2nd Semester
Delhi Technological University

**Submitted To:**

Ms. Lalita Luthra
Department of Information & Technology
Delhi Technological University

# INDEX

| S.No | Experiment | Date | Sign |
|---|---|---|---|
| 1 | Familiarize with Python software (Fibonacci numbers, sorting a list of numbers) | 22-08-2024 | |
| 2 | Write a program to load a dataset from UCI repository into Python workspace and print its dimensions. Also load the target or class variable and print its dimensions. | 29-08-2024 | |
| 3 | Write a program to clean the data by removing noisy data or outliers and solving missing value problem. | 05-09-2024 | |
| 4 | Write a program to explore different data visualization techniques. | 12-09-2024 | |
| 5 | Write a program to perform statistical analysis of the data in a given dataset (mean, variance, standard deviation, median, mode). | 19-09-2024 | |
| 6 | Write a program to perform a classification experiment on a dataset and its target or class variable (Naïve Bayes, Random Forest). | | |
| 7 | Write a program to perform a regression experiment on a dataset (linear regression). | | |
| 8 | Write a program to perform a clustering experiment on a dataset (K-means, Hierarchical agglomerative clustering). | | |
| 9 | Write a program to perform time series analysis for a given dataset. | | |
| 10 | Write a program to perform association rule mining for a given dataset. | | |

# PROGRAM 1

**Objective:** To familiarize with Python software (Fibonacci numbers, sorting a list of numbers)

## Code:

```
def fibonacci(n):
    fib_sequence = [0, 1]
    while len(fib_sequence) < n:
        fib_sequence.append(fib_sequence[-1] + fib_sequence[-2])
    return fib_sequence
n = 10
print(f"First {n} Fibonacci numbers: {fibonacci(n)}")
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]
    return arr
arr = [64, 34, 25, 12, 22, 11, 90]
sorted_arr = bubble_sort(arr)
print("Sorted array:", sorted_arr)
```

## Output:

```
First 10 Fibonacci numbers: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
Sorted array: [11, 12, 22, 25, 34, 64, 90]

=== Code Execution Successful ===
```

# PROGRAM 2

**Objective:** Write a program to load a dataset from UCI repository into Python workspace and print its dimensions. Also load the target or class variable and print its dimensions.

## Code:

```
import pandas as pd
from ucimlrepo import fetch_ucirepo

rice_dataset = fetch_ucirepo("Rice (Cammeo and Osmancik)")

data = rice_dataset.data.features
target = rice_dataset.data.targets

print("Data dimensions:", data.shape)
print("Target dimensions:", target.shape)
```

## Output:

```
PS C:\Users\vaibh\OneDrive\Desktop\html\college python + c +cpp> & C:/Python312/python.e
xe "c:/Users/vaibh/OneDrive/Desktop/html/college python + c +cpp/abc.py"
Data dimensions: (3810, 7)
Target dimensions: (3810, 1)
PS C:\Users\vaibh\OneDrive\Desktop\html\college python + c +cpp>
```

# PROGRAM 3

**Objective:** Write a program to clean the data by removing noisy data or outliers and solving missing value problem.

## Code:

```
import pandas as pd
from ucimlrepo import fetch_ucirepo

rice_cammeo_and_osmancik = fetch_ucirepo(id=545)
X = rice_cammeo_and_osmancik.data.features
y = rice_cammeo_and_osmancik.data.targets

df = X.copy()
df['Target'] = y
df=df.fillna(df.drop(columns=['Target']).median())

numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns

def remove_outliers_iqr(df, numeric_columns):
    Q1 = df[numeric_columns].quantile(0.25)
    Q3 = df[numeric_columns].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df_cleaned = df[~((df[numeric_columns] < lower_bound) | (df[numeric_columns] >
upper_bound)).any(axis=1)]
    return df_cleaned

df_cleaned = remove_outliers_iqr(df, numeric_columns)
print(f"\nOriginal dataset size: {df.shape[0]}")
print(f"Cleaned dataset size: {df_cleaned.shape[0]}")
```

## Output:

```
PS C:\Users\vaibh\OneDrive\Desktop\html\pythonDSVfiles> & C:/Python312/python.exe c:/Users/vaibh/O
neDrive/Desktop/html/pythonDSVfiles/final/abc2.py

Original dataset size: 3810
Cleaned dataset size: 3725
PS C:\Users\vaibh\OneDrive\Desktop\html\pythonDSVfiles>
```

# PROGRAM 4

**Objective:** Write a program to explore different data visualization techniques

## Code:

```
import pandas as pd
from matplotlib import pyplot as plt
from ucimlrepo import fetch_ucirepo

rice_cammeo_and_osmancik = fetch_ucirepo(id=545)
X = rice_cammeo_and_osmancik.data.features
y = rice_cammeo_and_osmancik.data.targets
df = X.copy()
df['Target'] = y

def pie_chart(df):
    features = ['Area', 'Perimeter', 'Major_Axis_Length']
    for feature in features:
        sums_by_target = df.groupby('Target').sum()

        sums_by_target[feature].plot(kind='pie', fontsize=20)
        plt.ylabel(feature, horizontalalignment='left')
        plt.title('Breakdown for ' + feature, fontsize=25)

        plt.savefig(f'rice_pie_for_{feature}.jpg')
        plt.close()

def bar_chart(df):
    sums_by_Target = df.groupby('Target').sum()
    var = 'Area'
    sums_by_Target[var].plot(kind='bar', fontsize=15, rot=30)
    plt.title('Breakdown for ' + var, fontsize=20)
    plt.savefig('rice_bar_for_one_variable.jpg')
    plt.close()

    sums_by_Target.plot(kind='bar', subplots=True, fontsize=12)
    plt.suptitle('Total Measurements, by Target')
    plt.savefig('rice_bar_for_each_variable.jpg')
    plt.close()

def histogram(df):
    df.drop(columns='Target').plot(kind='hist', subplots=True, layout=(3, 3),
bins=20, figsize=(10, 8))
    plt.suptitle('Rice Histograms', fontsize=20)
    plt.tight_layout(rect=[0, 0.03, 1, 0.95])
    plt.show()

def mean_mediam_mode(df):
    col = df['Area']
    Average = col.mean()
    Std = col.std()
    Median = col.median()
    Perc25 = col.quantile(0.25)
    Perc75 = col.quantile(0.75)
    Clean_Avg = col[(col > Perc25) & (col < Perc75)].mean()

    print(f"Average: {Average}")
    print(f"Standard Deviation: {Std}")
```

```
    print(f"Median: {Median}")
    print(f"25th Percentile: {Perc25}")
    print(f"75th Percentile: {Perc75}")
    print(f"Clean Average (excluding outliers): {Clean_Avg}")
```

## Output:

# PROGRAM 5

**Objective:** Write a program to perform statistical analysis of the data in a given dataset (mean, variance, standard deviation, median, mode).

## Code:

```
import pandas as pd
from ucimlrepo import fetch_ucirepo

rice_cammeo_and_osmancik = fetch_ucirepo(id=545)
X = rice_cammeo_and_osmancik.data.features
y = rice_cammeo_and_osmancik.data.targets

df = pd.DataFrame(X)
df['Target'] = y

def statistical_analysis(dataframe, feature):
    analysis = {}
    analysis['Mean'] = dataframe[feature].mean()
    analysis['Variance'] = dataframe[feature].var()
    analysis['Standard Deviation'] = dataframe[feature].std()
    analysis['Median'] = dataframe[feature].median()
    analysis['Mode'] = dataframe[feature].mode()[0]
    return analysis

feature_to_analyze = 'Area'
stats = statistical_analysis(df, feature_to_analyze)
print(f'Statistical Analysis for {feature_to_analyze}:')
for stat, value in stats.items():
    print(f'{stat}: {value}')
```

## Output:

```
PS C:\Users\vaibh\OneDrive\Desktop\html\pythonDSVfiles> & C:/Python312/python.exe c:/Users/vaibh/OneDrive/Deskto
p/html/pythonDSVfiles/final/abc4.py
Statistical Analysis for Area:
Mean: 12667.727559055118
Variance: 3001097.8690486476
Standard Deviation: 1732.3677060741602
Median: 12421.5
Mode: 11422
PS C:\Users\vaibh\OneDrive\Desktop\html\pythonDSVfiles>
```