CSBB 311 : Machine Learning

LAB ASSIGNMENT 1: Analysis Of Data Using Python Library

Submitted By:

Name: KARTIK MITTAL

Roll No: 221210056

Branch: CSE

Semester: 5th

Group: 2

Submitted To: Dr. Preeti Mehta

Department of Computer Science and Engineering

NATIONAL INSTITUTE OF TECHNOLOGY DELHI

2024

Q1 : - Loading of data file

Code :-

```python
import pandas as pd

# Load the Titanic dataset from seaborn's built-in dataset
data = pd.read_csv('tested.csv')

# Display the first few rows of the dataset
print("Data Loaded:")
print(data.head())
```

Result :-

```
PS C:\Users\HP\Desktop\college\semester 5\Machine Learning> python -u "c:\Users\HP\Desktop\college\semester 5\Machine Learn
ing\loadData.py"
Q
1          893         1       3                                Wilkes, Mrs. James (Ellen Needs)  ...    363272   7.0000    NaN        S
2          894         0       2                                       Myles, Mr. Thomas Francis  ...    240276   9.6875    NaN        Q
3          895         0       3                                            Wirz, Mr. Albert  ...    315154   8.6625    NaN        S
4          896         1       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  ...   3101298  12.2875    NaN        S

[5 rows x 12 columns]
PS C:\Users\HP\Desktop\college\semester 5\Machine Learning>
```

Q2 : - Size of features and names of features

Code :-

```python
import loadData
from loadData import data

# Size of the dataset
data_shape = data.shape
print(f"Size of the dataset: {data_shape}")

# Names of the features
features = data.columns.tolist()
print(f"Names of Features: {features}")
```

Result :-

```
python -u "c:\Users\HP\Desktop\college\semester 5\Machine Learning\2_checkFeatures.py"
Data Loaded:
   PassengerId  Survived  Pclass                                          Name  ...   Ticket     Fare  Cabin  Embarked
0          892         0       3                              Kelly, Mr. James  ...   330911   7.8292    NaN         Q
1          893         1       3              Wilkes, Mrs. James (Ellen Needs)  ...   363272   7.0000    NaN         S
2          894         0       2                     Myles, Mr. Thomas Francis  ...   240276   9.6875    NaN         Q
3          895         0       3                              Wirz, Mr. Albert  ...   315154   8.6625    NaN         S
4          896         1       3  Hirvonen, Mrs. Alexander (Helga E Lindqvist)  ...  3101298  12.2875    NaN         S

[5 rows x 12 columns]
Size of the dataset: (418, 12)
Names of Features: ['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin',
 'Embarked']
```

Q3 : - Finding the missing entities

Code :-

```python
# Finding missing data
missing_data = data.isnull().sum()
print("Missing Entities in Each Feature:")
print(missing_data)
```

Result :-

```
Missing Entities in Each Feature:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64
PS C:\Users\HP\Desktop\college\semester 5\Machine Learning>
```

Q4 : - Creating of file 1 and file 2 for the missing entities

Code :-

```python
import pandas as pd

# Load the Titanic dataset
df = pd.read_csv('tested.csv')

# Identify missing values in critical and non-critical columns
missing_critical = df[df[['Age', 'Fare']].isnull().any(axis=1)]
missing_non_critical = df[df[['Cabin', 'Ticket']].isnull().any(axis=1)]

# Save to file1.csv (critical columns)
missing_critical.to_csv('file1.csv', index=False)

# Save to file2.csv (non-critical columns)
missing_non_critical.to_csv('file2.csv', index=False)

```

Q5 : - Normalization of new files using two approaches

APPROACH 1 :- Using Normalization

Code :-

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the normalized files
file1 = pd.read_csv('file1_normalized.csv')
file2 = pd.read_csv('file2_normalized.csv')

# Plot histograms for numerical columns in file1
file1.select_dtypes(include=['float64', 'int64']).hist(figsize=(12, 10), bins=30)
plt.suptitle('Histograms for file1')
plt.show()

# Plot histograms for numerical columns in file2
file2.select_dtypes(include=['float64', 'int64']).hist(figsize=(12, 10), bins=30)
plt.suptitle('Histograms for file2')
plt.show()

```

APPROACH 2 :- Using Standardized

Code :-

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

# Load the files
file1 = pd.read_csv('file1.csv')
file2 = pd.read_csv('file2.csv')

# Initialize the StandardScaler
scaler = StandardScaler()

# Normalize the data in both files
file1_normalized = file1.copy()
file2_normalized = file2.copy()

# Apply Z-Score Normalization to numerical columns
for column in file1_normalized.select_dtypes(include=['float64', 'int64']).columns:
    file1_normalized[[column]] = scaler.fit_transform(file1_normalized[[column]])

for column in file2_normalized.select_dtypes(include=['float64', 'int64']).columns:
    file2_normalized[[column]] = scaler.fit_transform(file2_normalized[[column]])

# Save the normalized files
file1_normalized.to_csv('file1_standardized.csv', index=False)
file2_normalized.to_csv('file2_standardized.csv', index=False)
```

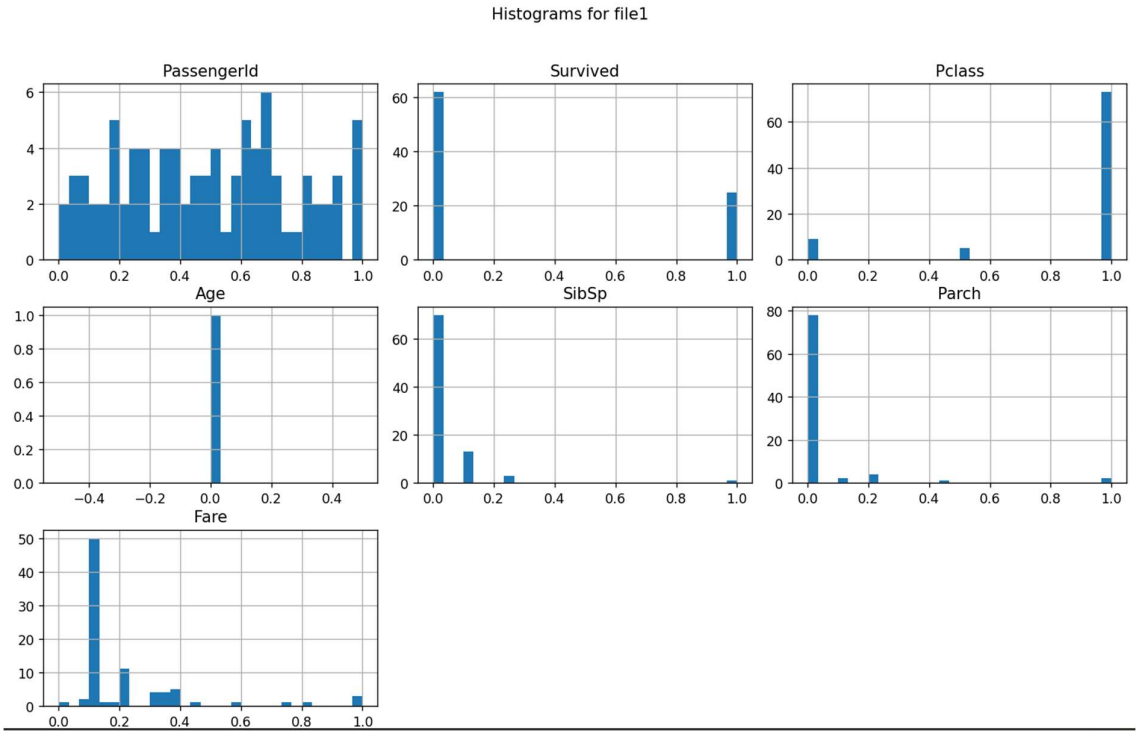
Q5 : - Visualization of features using different plots

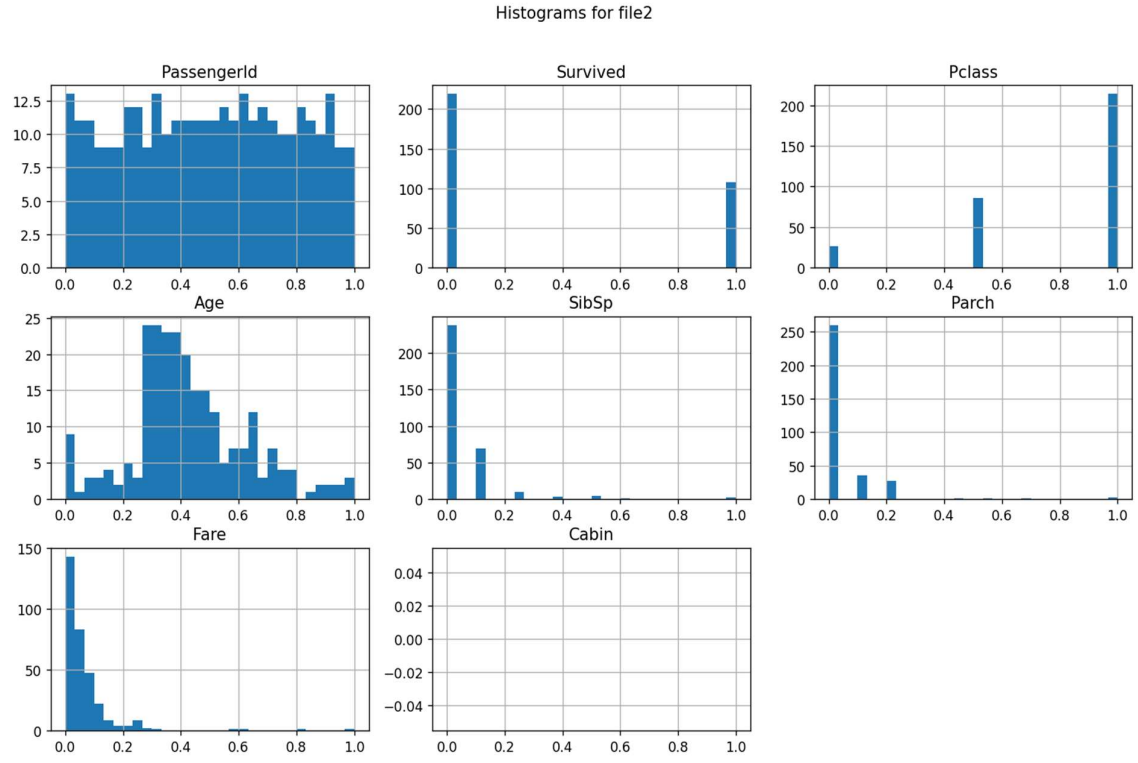
Using Histograms :-

Code :-

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load the normalized files
file1 = pd.read_csv('file1_normalized.csv')
file2 = pd.read_csv('file2_normalized.csv')

# Plot histograms for numerical columns in file1
file1.select_dtypes(include=['float64', 'int64']).hist(figsize=(12, 10), bins=30)
plt.suptitle('Histograms for file1')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust layout to make room for the title
plt.show()

# Plot histograms for numerical columns in file2
file2.select_dtypes(include=['float64', 'int64']).hist(figsize=(12, 10), bins=30)
plt.suptitle('Histograms for file2')
plt.tight_layout(rect=[0, 0.03, 1, 0.95])  # Adjust layout to make room for the title
plt.show()
```

Result :-



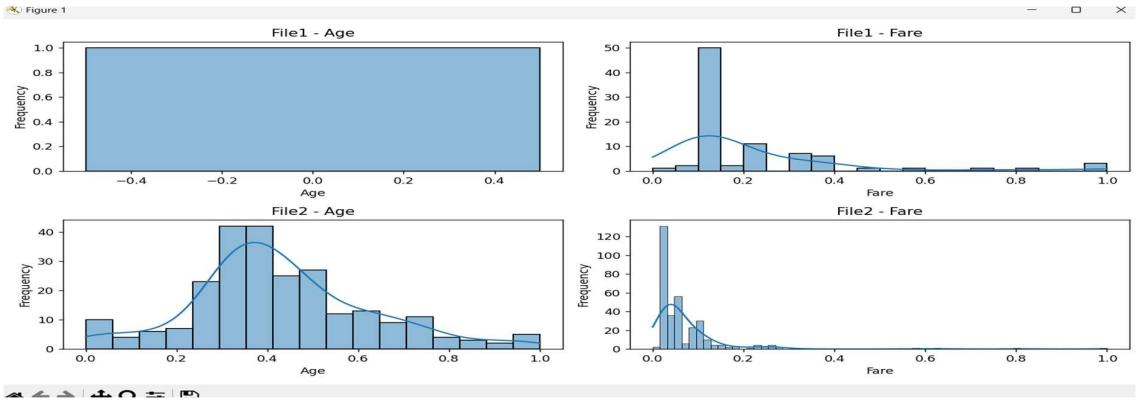


Using Scatters :-

Code :-

```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load the normalized files
file1_normalized = pd.read_csv('file1_normalized.csv')
file2_normalized = pd.read_csv('file2_normalized.csv')

# Replace 'Feature1' and 'Feature2' with actual column names
features = ['Age', 'Fare']  # Example feature names

# Plot histograms for file1
plt.figure(figsize=(12, 6))
for i, feature in enumerate(features):
    plt.subplot(2, len(features), i + 1)
    sns.histplot(file1_normalized[feature], kde=True)
    plt.title(f'File1 - {feature}')
    plt.xlabel(feature)
    plt.ylabel('Frequency')

# Plot histograms for file2
for i, feature in enumerate(features):
    plt.subplot(2, len(features), len(features) + i + 1)
    sns.histplot(file2_normalized[feature], kde=True)
    plt.title(f'File2 - {feature}')
    plt.xlabel(feature)
    plt.ylabel('Frequency')

plt.tight_layout()
plt.show()

# Plot scatter plots for pairs of features
plt.figure(figsize=(12, 6))
for i in range(len(features)):
    for j in range(i + 1, len(features)):
        plt.subplot(len(features) - 1, len(features) - 1, (i * (len(features) - 1)) + j)
        plt.scatter(file1_normalized[features[i]], file1_normalized[features[j]], alpha=0.5)
        plt.title(f'File1: {features[i]} vs {features[j]}')
        plt.xlabel(features[i])
        plt.ylabel(features[j])

        plt.subplot(len(features) - 1, len(features) - 1, (i * (len(features) - 1)) + j + len(features) - 1)
        plt.scatter(file2_normalized[features[i]], file2_normalized[features[j]], alpha=0.5)
        plt.title(f'File2: {features[i]} vs {features[j]}')
        plt.xlabel(features[i])
        plt.ylabel(features[j])

plt.tight_layout()
plt.show()
```

Result :-

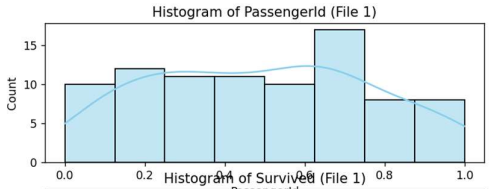


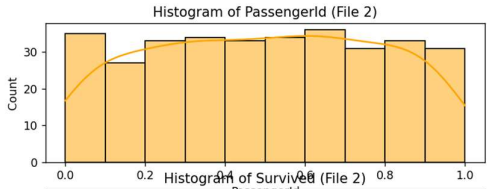
Using Mean Errors :-

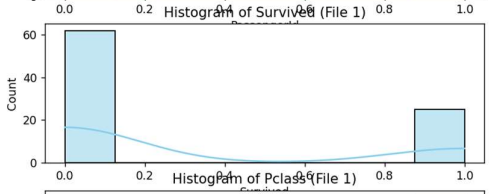
```python
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Load the normalized data
file1_normalized = pd.read_csv('file1_normalized.csv')
file2_normalized = pd.read_csv('file2_normalized.csv')

# Select 2-3 features to visualize
features = ['PassengerId', 'Survived', 'Pclass']  # Replace with actual feature names

# Set up the plotting environment
fig, axes = plt.subplots(nrows=3, ncols=2, figsize=(15, 12))

# Plot histograms
for i, feature in enumerate(features):
    sns.histplot(file1_normalized[feature], kde=True, ax=axes[i, 0], color='skyblue')
    axes[i, 0].set_title(f'Histogram of {feature} (File 1)')

    sns.histplot(file2_normalized[feature], kde=True, ax=axes[i, 1], color='orange')
    axes[i, 1].set_title(f'Histogram of {feature} (File 2)')

# Create a new figure for scatter plots
fig, axes = plt.subplots(nrows=1, ncols=3, figsize=(18, 5))
```

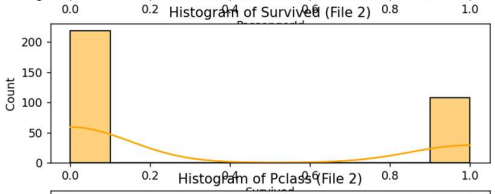
```python
# Scatter plots between features
for i, feature in enumerate(features):
    sns.scatterplot(x=file1_normalized[feature], y=file2_normalized[feature], ax=axes[i])
    axes[i].set_title(f'Scatter Plot of {feature} (File 1 vs File 2)')
    axes[i].set_xlabel(f'{feature} (File 1)')
    axes[i].set_ylabel(f'{feature} (File 2)')

# Create a new figure for mean error plots
fig, ax = plt.subplots(figsize=(8, 6))

# Calculate the mean and standard deviation for each feature in both files
means_file1 = file1_normalized[features].mean()
stds_file1 = file1_normalized[features].std()
means_file2 = file2_normalized[features].mean()
stds_file2 = file2_normalized[features].std()

# Plot mean and error bars
ax.errorbar(features, means_file1, yerr=stds_file1, fmt='o', capsize=5, label='File 1', color='skyblue'
ax.errorbar(features, means_file2, yerr=stds_file2, fmt='o', capsize=5, label='File 2', color='orange')
ax.set_title('Mean and Error Bars for Selected Features')
ax.set_xlabel('Features')
ax.set_ylabel('Mean ± Standard Deviation')
ax.legend()

# Adjust the layout
plt.tight_layout()

# Show all plots
plt.show()
```

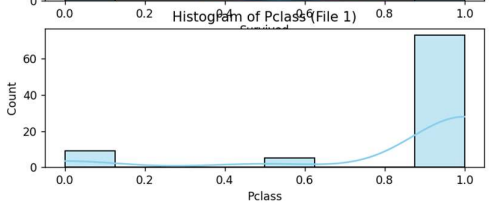
Result :-

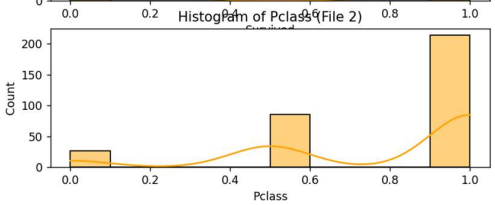

Histogram of PassengerId (File 1)
Count

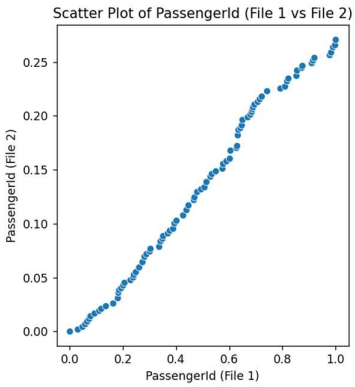

Histogram of PassengerId (File 2)
Count

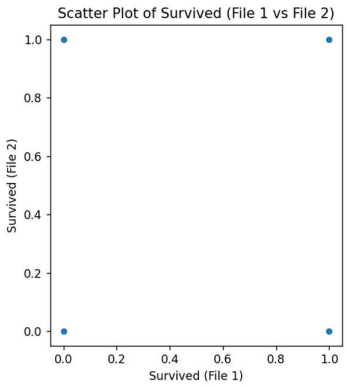

Histogram of Survived (File 1)
Count

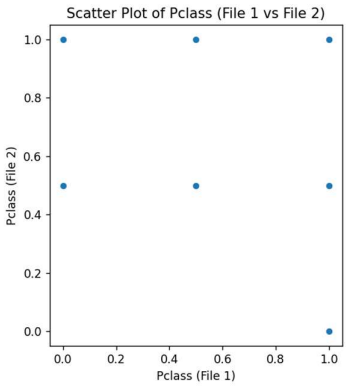

Histogram of Survived (File 2)
Count

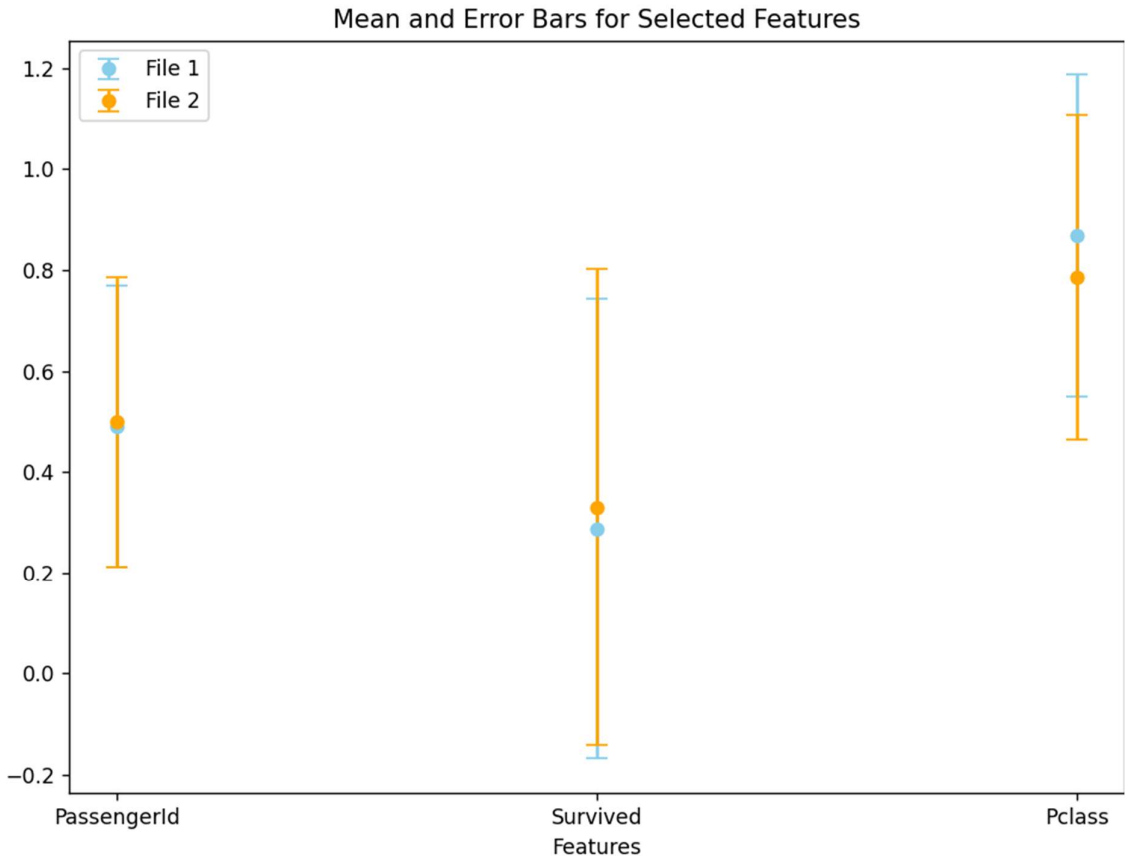

Histogram of Pclass (File 1)
Count
Pclass


Histogram of Pclass (File 2)
Count
Pclass


Scatter Plot of PassengerId (File 1 vs File 2)
PassengerId (File 2)
PassengerId (File 1)


Scatter Plot of Survived (File 1 vs File 2)
Survived (File 2)
Survived (File 1)


Scatter Plot of Pclass (File 1 vs File 2)
Pclass (File 2)
Pclass (File 1)


Mean and Error Bars for Selected Features
File 1
File 2
1.2
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
PassengerId
Survived
Pclass
Features