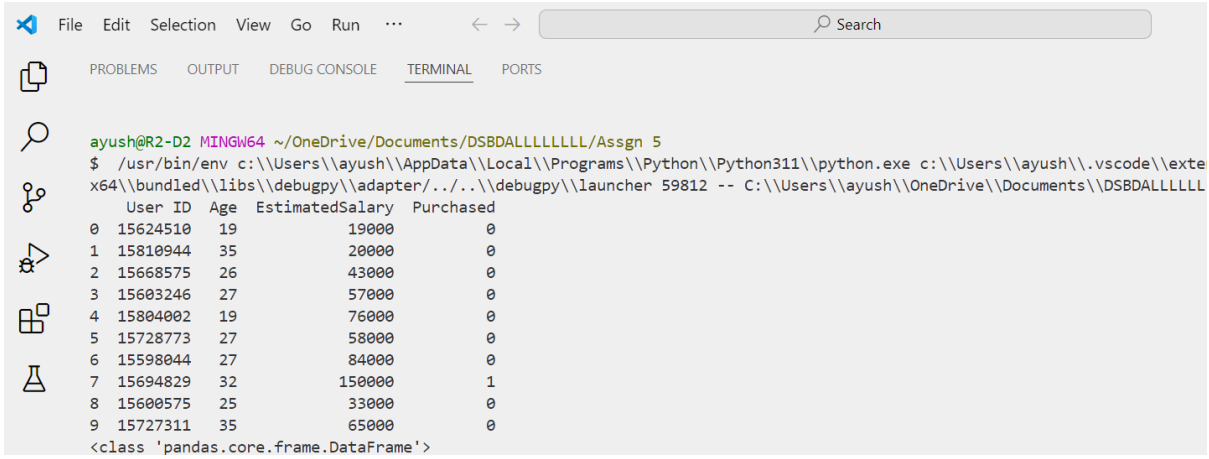
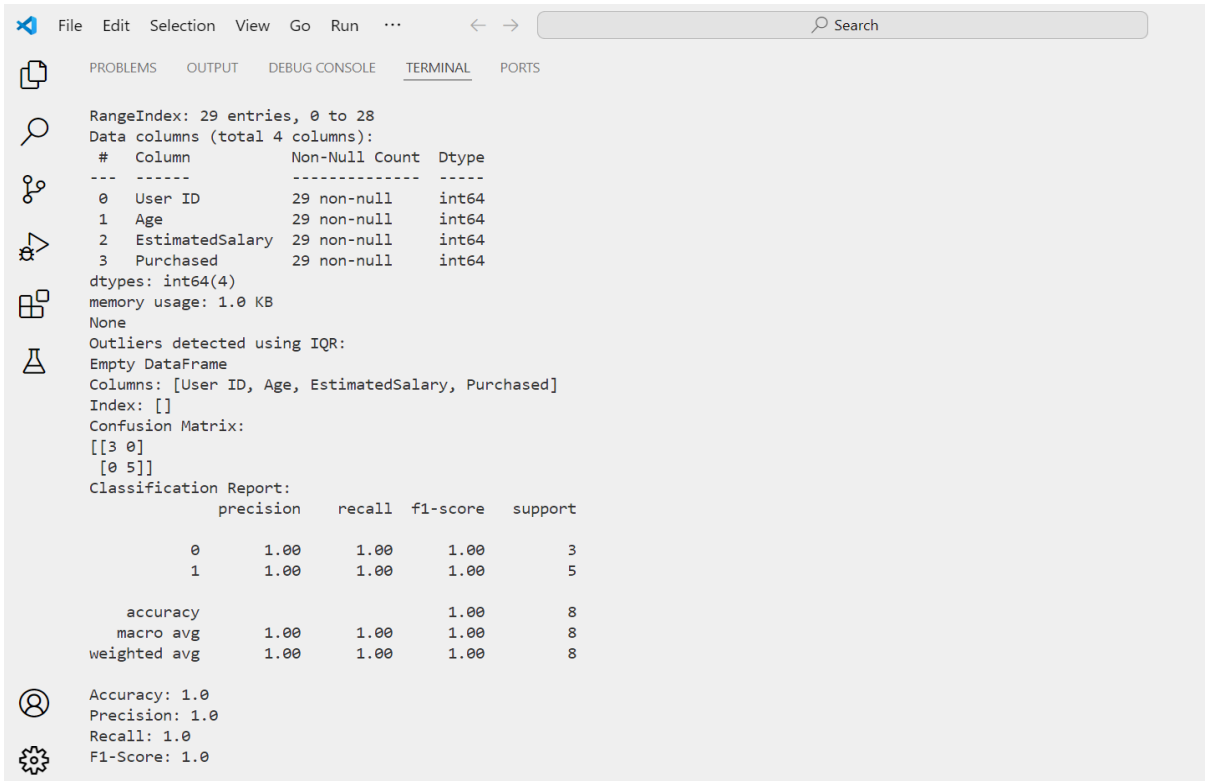
```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import confusion_matrix, classification_report

# Load the CSV file into a DataFrame
df = pd.read_csv(r"ass5.csv")

# Display first few rows and information about the DataFrame
print(df.head(10))
print(df.info())

# Detect outliers using IQR (assuming 'Age' column for this example)
Score = df['Age']
q1, q3 = Score.quantile(0.25), Score.quantile(0.75)
iqr = q3 - q1
lower_fence = q1 - (1.5 * iqr)
upper_fence = q3 + (1.5 * iqr)
outliers = df[(df['Age'] < lower_fence) | (df['Age'] > upper_fence)]
print("Outliers detected using IQR:")
print(outliers)

# Prepare features (x) and target (y) variables
x = df[['Age', 'EstimatedSalary']]  # Adjust column names as per your DataFrame
y = df['Purchased']                 # Adjust column name for target variable

# Split data into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=0)

# Feature scaling
sc = StandardScaler()
```

```
x_train = sc.fit_transform(x_train)
x_test = sc.transform(x_test)

# Train a Logistic Regression model
classifier = LogisticRegression(random_state=0)
classifier.fit(x_train, y_train)

# Make predictions
y_pred = classifier.predict(x_test)

# Evaluate the model
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix:")
print(cm)

c_report = classification_report(y_test, y_pred)
print("Classification Report:")
print(c_report)

# Calculate and print accuracy, precision, recall, and F1-score
tn, fp, fn, tp = cm.ravel()
accuracy = (tp + tn) / (tp + tn + fp + fn)
precision = tp / (tp + fp)
recall = tp / (tp + fn)
f1_score = 2 * (precision * recall) / (precision + recall)

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-Score:", f1_score)
```

OUTPUT –


File Edit Selection View Go Run ···
Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
ayush@R2-D2 MINGW64 ~/OneDrive/Documents/DSBDALLLLLLL/Assgn 5
$ /usr/bin/env c:\\Users\\ayush\\AppData\\Local\\Programs\\Python\\Python311\\python.exe c:\\Users\\ayush\\.vscode\\exte
x64\\bundled\\libs\\debugpy\\adapter/../..\\debugpy\\launcher 59812 -- C:\\Users\\ayush\\OneDrive\\Documents\\DSBDALLLLLL
User ID Age EstimatedSalary Purchased
0 15624510 19 19000 0
1 15810944 35 20000 0
2 15668575 26 43000 0
3 15603246 27 57000 0
4 15804002 19 76000 0
5 15728773 27 58000 0
6 15598044 27 84000 0
7 15694829 32 150000 1
8 15600575 25 33000 0
9 15727311 35 65000 0
<class 'pandas.core.frame.DataFrame'>


File Edit Selection View Go Run ···
Search
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
RangeIndex: 29 entries, 0 to 28
Data columns (total 4 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 User ID 29 non-null int64
1 Age 29 non-null int64
2 EstimatedSalary 29 non-null int64
3 Purchased 29 non-null int64
dtypes: int64(4)
memory usage: 1.0 KB
None
Outliers detected using IQR:
Empty DataFrame
Columns: [User ID, Age, EstimatedSalary, Purchased]
Index: []
Confusion Matrix:
[[3 0]
[0 5]]
Classification Report:
precision recall f1-score support
0 1.00 1.00 1.00 3
1 1.00 1.00 1.00 5
accuracy 1.00 8
macro avg 1.00 1.00 1.00 8
weighted avg 1.00 1.00 1.00 8
Accuracy: 1.0
Precision: 1.0
Recall: 1.0
F1-Score: 1.0

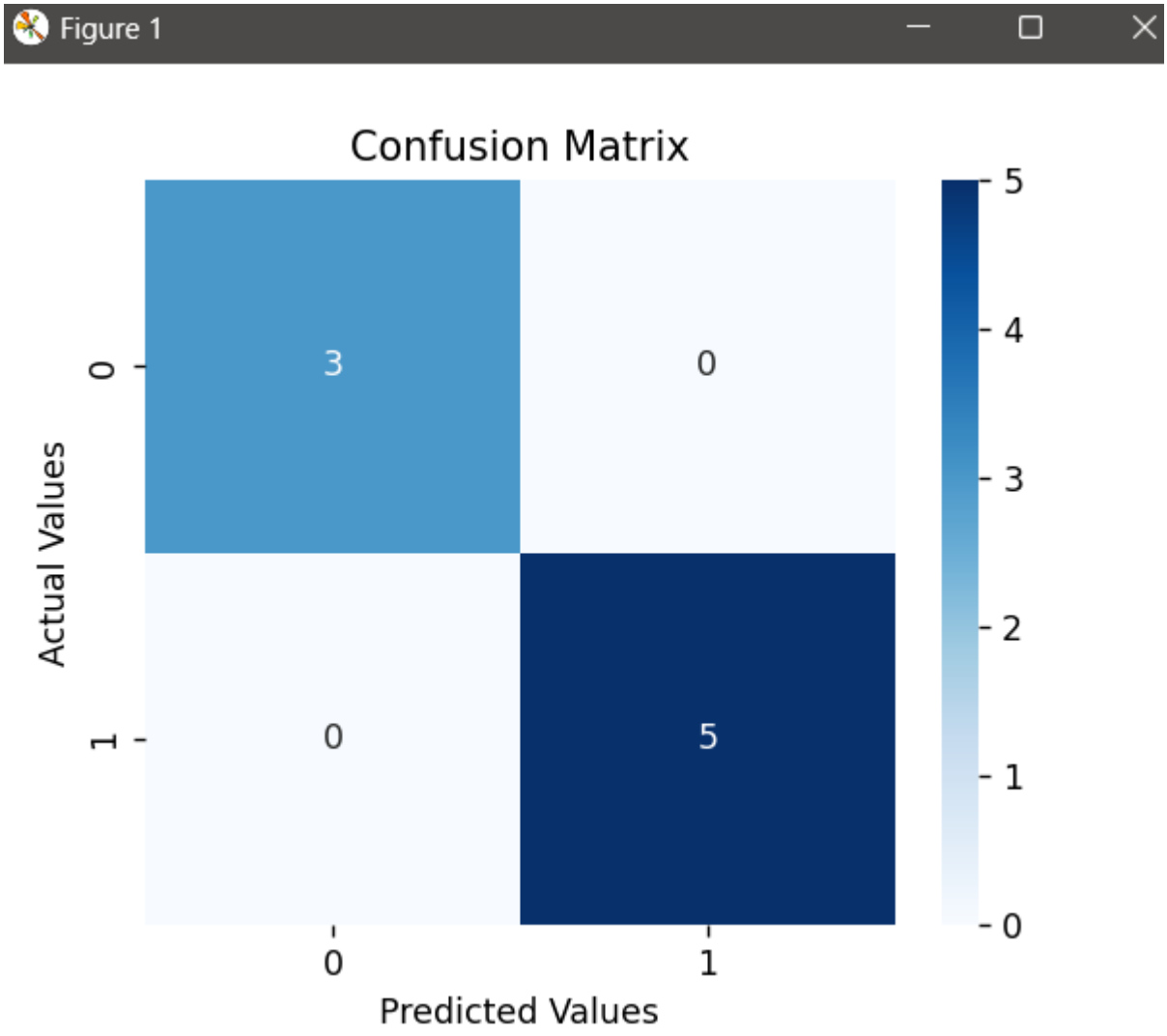


Figure 1
Confusion Matrix
3
0
0
5
0
1
Actual Values
Predicted Values
5
4
3
2
1
0