Week 11

1. Write a shell script to check if a file exists and display an appropriate message.

```
echo "Enter the filename (with path if not in current directory):"
read filename
if [ -e "$filename" ]; then
   echo "The file '$filename' exists."
else
   echo "The file '$filename' does not exist."
fi
```

```
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$ ls
Q1.sh  Q2.sh  Q3.sh  Q4.sh
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$ ./Q1.sh
Enter the filename (with path if not in current directory):
Q3.sh
The file 'Q3.sh' exists.
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$
```

2. Write a shell script to find the factorial of a number using loops.

```
echo "Enter a number:"
read num
factorial=1
for (( i=1; i<=num; i++ ))
do
   factorial=$((factorial * i))
done
echo "The factorial of $num is $factorial."
```

```
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$ ./Q2.sh
Enter a number:
6
The factorial of 6 is 720.
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$
```

3. Write a shell script to demonstrate the use of conditionals (if-else statements).

```
greet() {
   echo "Hello, $1! Welcome to the shell script tutorial."
}
echo "Enter your name:"
read name
greet "$name"
```

```
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$ ./Q3.sh
Enter your name:
hammad
Hello, hammad! Welcome to the shell script tutorial.
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$
```

4. Write a shell script to create and use a simple function.

```
echo "Enter a number:"
read num
if [ "$num" -gt 0 ]; then
   echo "The number $num is positive."
elif [ "$num" -lt 0 ]; then
   echo "The number $num is negative."
else
   echo "The number is zero."
fi
```

```
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$ ./Q4.sh
Enter a number:
5
The number 5 is positive.
hammadxjaved@INBook-X1:/mnt/e/linux-week/Week-11$
```

5. You have been given a dataset demo.csv having independent features as x1, x2, x3, x4, x5, x6, x7 and dependent feature as y with value either 0 or 1. All independent features are continuous data except x1 and x2, which are having nominal data. Now write python program for the following:

a. Clean independent features

b. Add one more feature x7 having values between 0 and 1.

c. Perform scaling

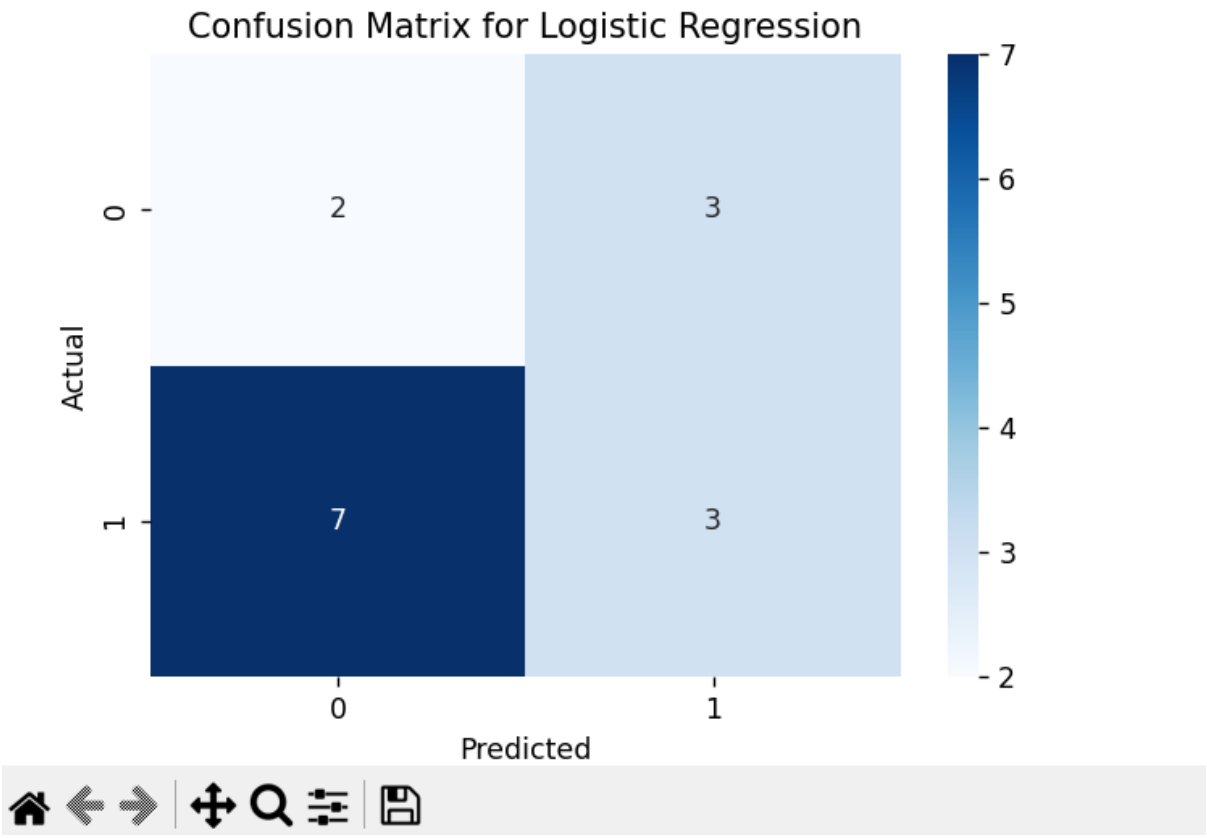
d. Train this dataset using Logistic regression, Decision Tree and Random Forest. Compare the performance of all the models based on accuracy and F1 score.

e. Draw confusion matrix of each model

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix
import matplotlib.pyplot as plt
import seaborn as sns

data = pd.read_csv('Week-11/demo.csv')

label_encoder = LabelEncoder()
data['x1'] = label_encoder.fit_transform(data['x1'])
data['x2'] = label_encoder.fit_transform(data['x2'])
```

```
data.fillna(data.mean(), inplace=True)

data['x7'] = np.random.rand(len(data))

X = data.drop('y', axis=1)
y = data['y']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.3, random_state=42)

models = {
    'Logistic Regression': LogisticRegression(),
    'Decision Tree': DecisionTreeClassifier(),
    'Random Forest': RandomForestClassifier()
}

performance = {}

for model_name, model in models.items():
    model.fit(X_train, y_train)

    y_pred = model.predict(X_test)

    accuracy = accuracy_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)

    performance[model_name] = {'Accuracy': accuracy, 'F1 Score': f1}
```

```
    cm = confusion_matrix(y_test, y_pred)
    plt.figure(figsize=(6, 4))
    sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=[0, 1], yticklabels=[0, 1])
    plt.xlabel('Predicted')
    plt.ylabel('Actual')
    plt.title(f'Confusion Matrix for {model_name}')
    plt.show()

performance_df = pd.DataFrame(performance).T
print("Model Performance Comparison:")
print(performance_df)
```



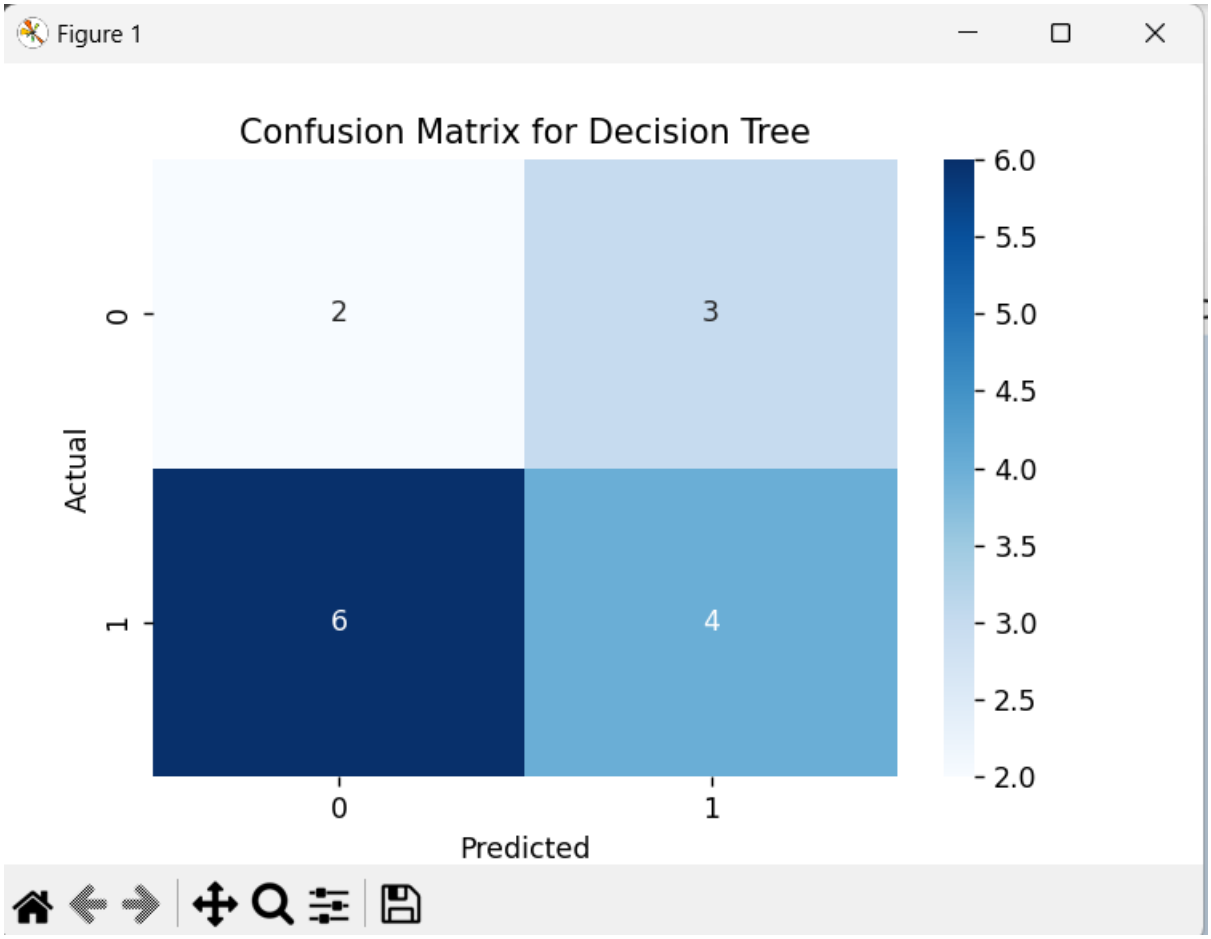
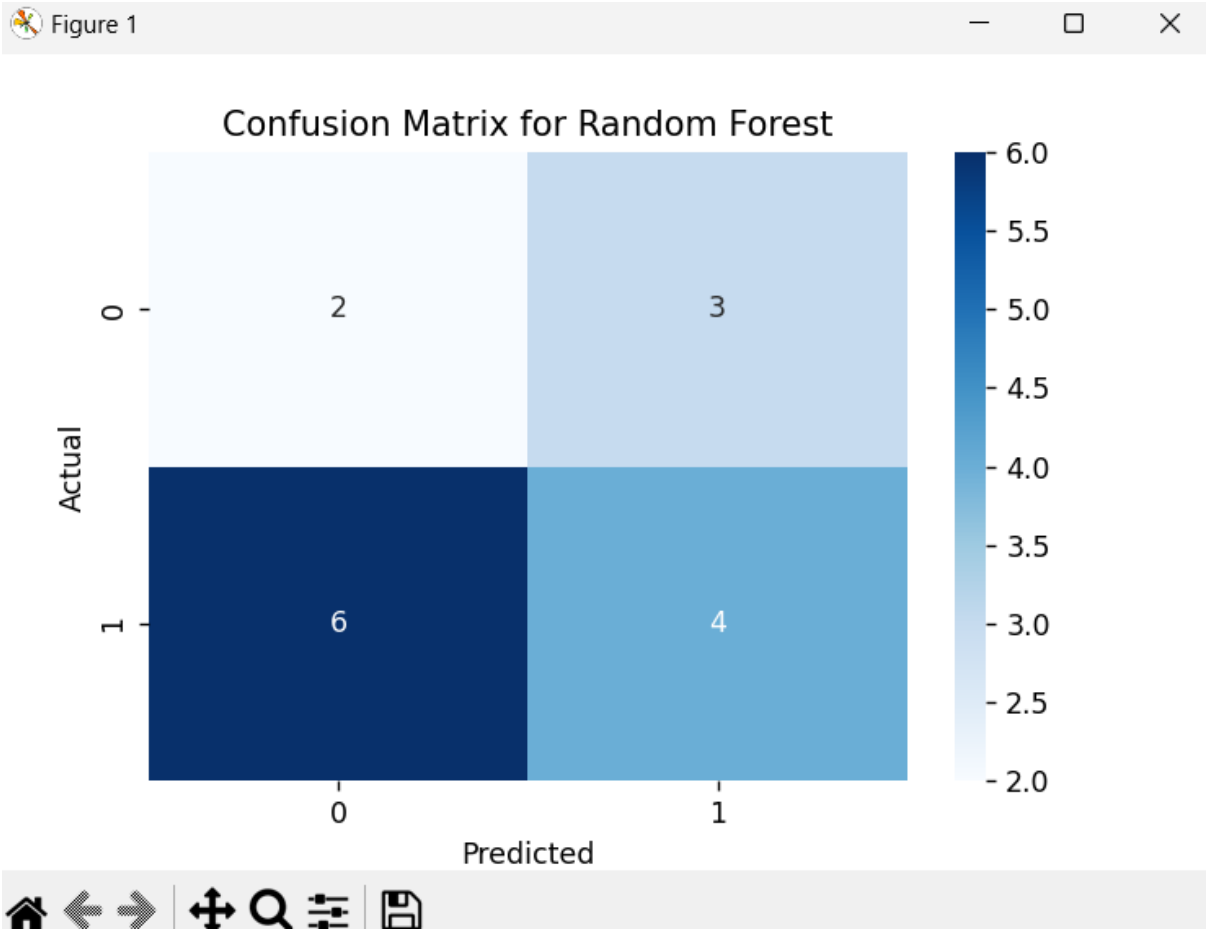


Figure 1
Confusion Matrix for Decision Tree
2
3
6
4
Actual
0
1
Predicted
0
1
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0




```
PS C:\Users\Hammad\OneDrive - myamu.ac.in\Desktop\MCA\MCA III\CAMS3P01 Laboratory Course-III (Min
i Project)\Weeks\MCA-III_LAB> & C:/Users/Hammad/AppData/Local/Microsoft/WindowsApps/python3.12.ex
e "c:/Users/Hammad/OneDrive - myamu.ac.in/Desktop/MCA/MCA III/CAMS3P01 Laboratory Course-III (Min
i Project)/Weeks/MCA-III_LAB/Week-11/Q5.py"
Model Performance Comparison:
                     Accuracy  F1 Score
Logistic Regression  0.333333  0.375000
Decision Tree        0.400000  0.470588
Random Forest        0.400000  0.470588
PS C:\Users\Hammad\OneDrive - myamu.ac.in\Desktop\MCA\MCA III\CAMS3P01 Laboratory Course-III (Min
```