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

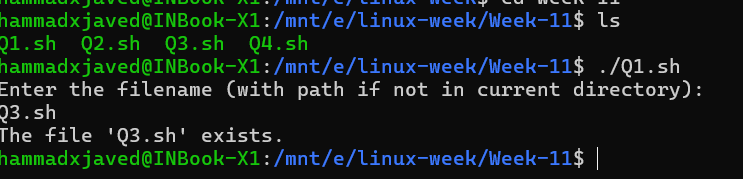
if [ -e "$filename" ]; then

    echo "The file '$filename' exists."

else

    echo "The file '$filename' does not exist."

fi



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

echo "Enter a number:"

read num

factorial=1

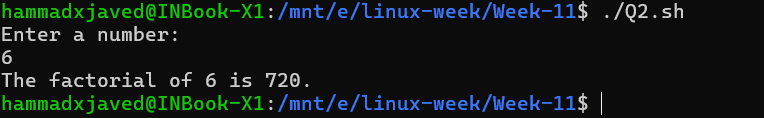
for (( i=1; i<=num; i++ ))

do

    factorial=$((factorial \* i))

done

echo "The factorial of $num is $factorial."



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

greet() {

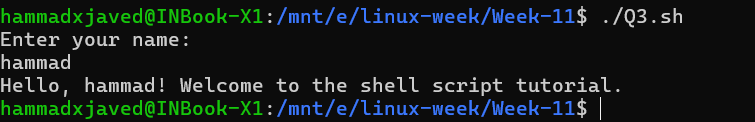
    echo "Hello, $1! Welcome to the shell script tutorial."

}

echo "Enter your name:"

read name

greet "$name"



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."

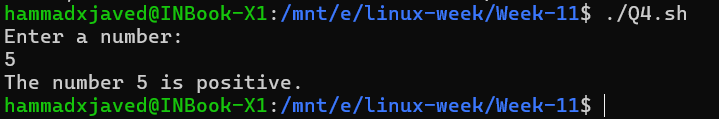
elif [ "$num" -lt 0 ]; then

    echo "The number $num is negative."

else

    echo "The number is zero."

fi



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)

