**WEEEK 4 COM624 LAB ACTIVITIES: LOGISTIC REGRESSION**

**Part 3: Logistic Regression**

**Scenario**

You are working for an e-commerce company. Your task is to predict whether a customer will **purchase a product** based on their browsing behaviour, time spent on site, and other features.

**Dataset Download**

We will use the **Social\_Network\_Ads.csv** dataset from GitHub. It contains features like Age, EstimatedSalary, and a binary target Purchased.

Download Social\_Network\_Ads.csv ➡ Save the file as social\_ads.csv in your project folder.

**Step-by-Step Instructions**

**Step 1: Set Up Your Environment**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report, confusion\_matrix

**Step 2: Load the Dataset**

df = pd.read\_csv('social\_ads.csv')

print(df.head())

print(df.info())

**Step 3: Prepare the Data**

# Select features and target

X = df[['Age', 'EstimatedSalary']]

y = df['Purchased']

# Split into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

# Feature scaling for better performance

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

**Step 4: Build the Logistic Regression Model**

model = LogisticRegression()

model.fit(X\_train, y\_train)

**Step 5: Make Predictions**

y\_pred = model.predict(X\_test)

print("Predictions:", y\_pred)

**Step 6: Evaluate the Model**

# Confusion matrix

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

# Classification report

print(classification\_report(y\_test, y\_pred))

**Step 7: Visualise Decision Boundary (Optional)**

# Only works well with 2 features

from matplotlib.colors import ListedColormap

X\_set, y\_set = X\_test, y\_test

X1, X2 = X\_set[:, 0], X\_set[:, 1]

X1\_grid, X2\_grid = np.meshgrid(

np.arange(start=X1.min()-1, stop=X1.max()+1, step=0.01),

np.arange(start=X2.min()-1, stop=X2.max()+1, step=0.01)

)

plt.contourf(X1\_grid, X2\_grid,

model.predict(np.array([X1\_grid.ravel(), X2\_grid.ravel()]).T).reshape(X1\_grid.shape),

alpha=0.75, cmap=ListedColormap(('red', 'green')))

plt.scatter(X1, X2, c=y\_set, edgecolors='k', cmap=ListedColormap(('red', 'green')))

plt.title("Logistic Regression Decision Boundary")

plt.xlabel("Age (scaled)")

plt.ylabel("Estimated Salary (scaled)")

plt.show()

**THE END**

**Final Complete Code**

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from sklearn.linear\_model import LogisticRegression

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import classification\_report, confusion\_matrix

from matplotlib.colors import ListedColormap

df = pd.read\_csv('social\_ads.csv')

print(df.head())

print(df.info())

X = df[['Age', 'EstimatedSalary']]

y = df['Purchased']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

scaler = StandardScaler()

X\_train = scaler.fit\_transform(X\_train)

X\_test = scaler.transform(X\_test)

model = LogisticRegression()

model.fit(X\_train, y\_train)

y\_pred = model.predict(X\_test)

print("Predictions:", y\_pred)

cm = confusion\_matrix(y\_test, y\_pred)

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')

plt.title("Confusion Matrix")

plt.xlabel("Predicted")

plt.ylabel("Actual")

plt.show()

print(classification\_report(y\_test, y\_pred))

# Decision boundary

X\_set, y\_set = X\_test, y\_test

X1, X2 = X\_set[:, 0], X\_set[:, 1]

X1\_grid, X2\_grid = np.meshgrid(

np.arange(start=X1.min()-1, stop=X1.max()+1, step=0.01),

np.arange(start=X2.min()-1, stop=X2.max()+1, step=0.01)

)

plt.contourf(X1\_grid, X2\_grid,

model.predict(np.array([X1\_grid.ravel(), X2\_grid.ravel()]).T).reshape(X1\_grid.shape),

alpha=0.75, cmap=ListedColormap(('red', 'green')))

plt.scatter(X1, X2, c=y\_set, edgecolors='k', cmap=ListedColormap(('red', 'green')))

plt.title("Logistic Regression Decision Boundary")

plt.xlabel("Age (scaled)")

plt.ylabel("Estimated Salary (scaled)")

plt.show()