**WEEK 4 COM624 LAB ACTIVITIES PART 1: LINEAR REGRESSION**

**Regression Analysis Lab Guide**

**Learning Outcomes**

By the end of this lab, students will be able to:

* Understand and apply different regression models
* Load and prepare real-world datasets
* Build and evaluate regression models using scikit-learn
* Interpret coefficients and performance metrics
* Visualise regression lines and residuals

**Models Covered**

1. **Linear Regression**
2. **Multiple Linear Regression**
3. **Logistic Regression**

**Part 1: Linear Regression**

**Scenario**

You are working for a tech startup that wants to predict **employee salaries** based on **years of experience**.

**Download**

Salary\_Data.csv on Kaggle ➡ Rename the file to salary\_data.csv and place it 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 LinearRegression

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

from sklearn.metrics import mean\_squared\_error, r2\_score

**Step 2: Load the Dataset**

df = pd.read\_csv('salary\_data.csv')

print(df.head())

**Step 3: Visualise the Data**

sns.scatterplot(x='YearsExperience', y='Salary', data=df)

plt.title("Years of Experience vs Salary")

plt.xlabel("Years of Experience")

plt.ylabel("Salary (£)")

plt.show()

**Step 4: Prepare the Data**

X = df[['YearsExperience']]

y = df['Salary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

**Step 5: Build the Model**

model = LinearRegression()

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

print("Intercept:", model.intercept\_)

print("Slope:", model.coef\_[0])

**Step 6: Make Predictions**

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

comparison = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

print(comparison)

**Step 7: Evaluate the Model**

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error:", mse)

print("R-squared:", r2)

**Step 8: Visualise the Regression Line**

plt.scatter(X\_train, y\_train, color='blue', label='Training data')

plt.plot(X\_train, model.predict(X\_train), color='red', label='Regression line')

plt.title("Linear Regression Fit")

plt.xlabel("Years of Experience")

plt.ylabel("Salary (£)")

plt.legend()

plt.show()

**Step 9: Residual Analysis**

residuals = y\_test - y\_pred

sns.histplot(residuals, kde=True)

plt.title("Residuals Distribution")

plt.xlabel("Residuals")

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 LinearRegression

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

from sklearn.metrics import mean\_squared\_error, r2\_score

df = pd.read\_csv('salary\_data.csv')

sns.scatterplot(x='YearsExperience', y='Salary', data=df)

plt.title("Years of Experience vs Salary")

plt.xlabel("Years of Experience")

plt.ylabel("Salary (£)")

plt.show()

X = df[['YearsExperience']]

y = df['Salary']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

model = LinearRegression()

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

print("Intercept:", model.intercept\_)

print("Slope:", model.coef\_[0])

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

comparison = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred})

print(comparison)

mse = mean\_squared\_error(y\_test, y\_pred)

r2 = r2\_score(y\_test, y\_pred)

print("Mean Squared Error:", mse)

print("R-squared:", r2)

plt.scatter(X\_train, y\_train, color='blue', label='Training data')

plt.plot(X\_train, model.predict(X\_train), color='red', label='Regression line')

plt.title("Linear Regression Fit")

plt.xlabel("Years of Experience")

plt.ylabel("Salary (£)")

plt.legend()

plt.show()

residuals = y\_test - y\_pred

sns.histplot(residuals, kde=True)

plt.title("Residuals Distribution")

plt.xlabel("Residuals")

plt.show()