**Practical 6**

**Regression and Its Types**

* **Implement simple linear regression using a dataset.**
* **Explore and interpret the regression model coefficients and goodness-of-fit measures.**
* **Extend the analysis to multiple linear regression and assess the impact of additional predictors.**

Regression analysis is a statistical method used to model and analyze the relationships between a dependent variable and one or more independent variables. It helps in predicting outcomes and understanding the impact of predictors on the dependent variable.

**Types of Regression:**

1. **Simple Linear Regression** – Involves one independent variable and one dependent variable.
2. **Multiple Linear Regression** – Involves multiple independent variables affecting a dependent variable.
3. **Simple Linear Regression** **Dataset:**

We will use a sample dataset with two variables:

* + **Years of Experience** (Independent Variable - X)
  + **Salary** (Dependent Variable - Y)

**Code:**

# Import necessary libraries import numpy as np

import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression

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

# Sample dataset data = {

'Years of Experience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Salary': [30000, 35000, 40000, 48000, 53000, 60000, 65000, 70000, 78000, 85000]

}

# Convert dictionary to DataFrame df = pd.DataFrame(data)

# Display first few rows print("\nDataset:\n", df.head())

# Scatter Plot (Visualizing Relationship) plt.figure(figsize=(8, 6))

sns.scatterplot(x=df['Years of Experience'], y=df['Salary']) plt.xlabel("Years of Experience")

plt.ylabel("Salary")

plt.title("Scatter Plot of Salary vs Years of Experience") plt.show()

# Splitting data into training and testing sets

X = df[['Years of Experience']] # Independent Variable y = df['Salary'] # Dependent Variable

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

model = LinearRegression() model.fit(X\_train, y\_train) # Get the Coefficients

slope = model.coef\_[0] intercept = model.intercept\_

print(f"\nRegression Coefficient (Slope): {slope}") print(f"Intercept: {intercept}")

# Make Predictions

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

# Calculate Model Performance Metrics mse = mean\_squared\_error(y\_test, y\_pred) r2 = r2\_score(y\_test, y\_pred) print(f"\nMean Squared Error: {mse}") print(f"R-Squared Value: {r2}")

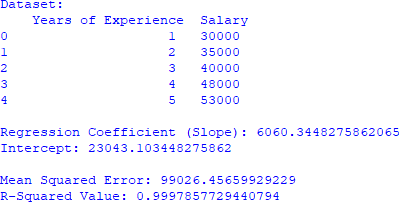
# Plot Regression Line plt.figure(figsize=(8, 6))

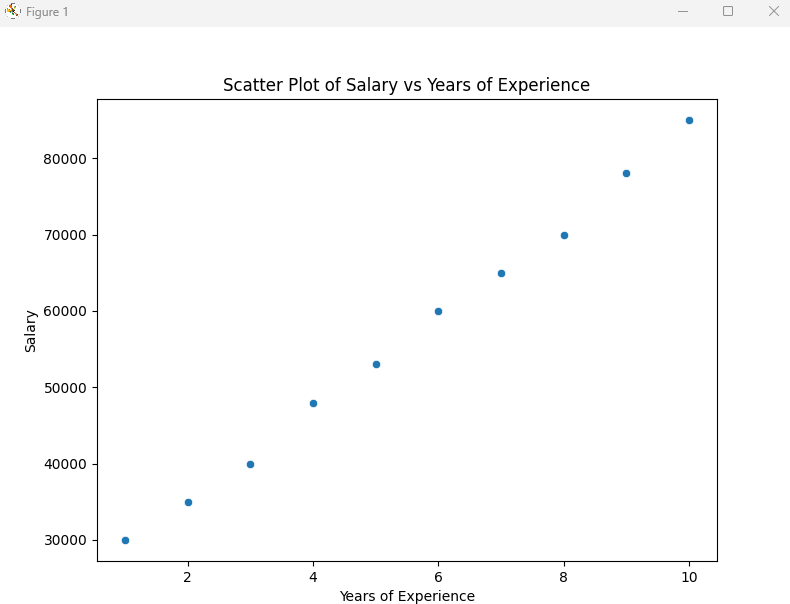
sns.scatterplot(x=X\_test['Years of Experience'], y=y\_test, label="Actual") sns.lineplot(x=X\_test['Years of Experience'], y=y\_pred, color='red', label="Regression Line") plt.xlabel("Years of Experience")

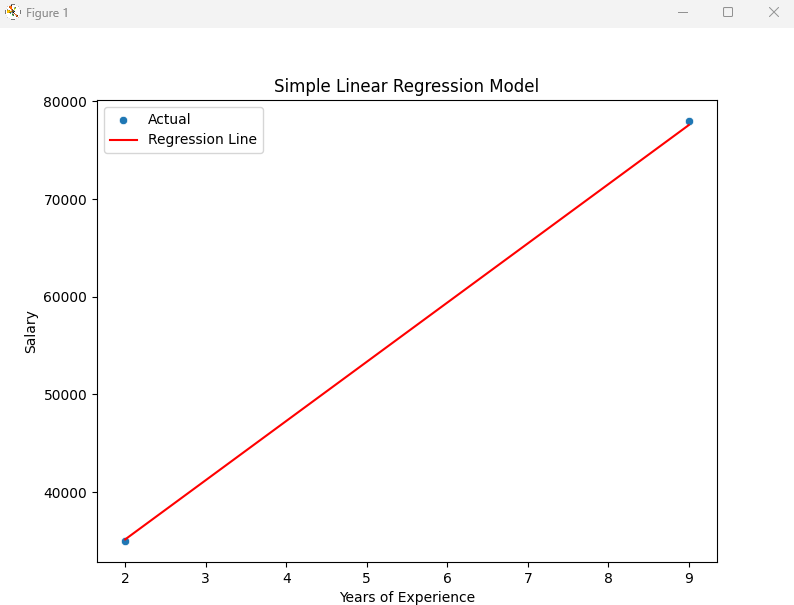
plt.ylabel("Salary")

plt.title("Simple Linear Regression Model") plt.legend()

plt.show()







1. **Multiple Linear Regression**

Code:

import numpy as np import pandas as pd

import matplotlib.pyplot as plt import seaborn as sns

from sklearn.model\_selection import train\_test\_split from sklearn.linear\_model import LinearRegression

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

# New dataset with an additional feature data = {

'Years of Experience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

'Number of Projects': [1, 2, 2, 3, 3, 4, 4, 5, 5, 6],

'Salary': [30000, 35000, 40000, 48000, 53000, 60000, 65000, 70000, 78000, 85000]

}

df = pd.DataFrame(data) print("\nNew Dataset:\n", df.head())

# Splitting Data

X = df[['Years of Experience', 'Number of Projects']] y = df['Salary']

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

# Train Multiple Linear Regression Model multi\_model = LinearRegression() multi\_model.fit(X\_train, y\_train)

# Get Coefficients

coefficients = multi\_model.coef\_ intercept = multi\_model.intercept\_

print(f"\nRegression Coefficients: {coefficients}") print(f"Intercept: {intercept}")

# Make Predictions

y\_pred\_multi = multi\_model.predict(X\_test)

# Evaluate Model

mse\_multi = mean\_squared\_error(y\_test, y\_pred\_multi) r2\_multi = r2\_score(y\_test, y\_pred\_multi)

print(f"\nMean Squared Error (Multiple Regression): {mse\_multi}") print(f"R-Squared Value (Multiple Regression): {r2\_multi}")

# Feature Importance

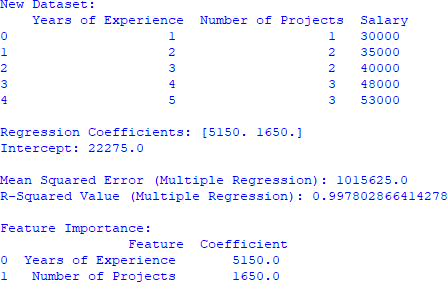
feature\_importance = pd.DataFrame({'Feature': ['Years of Experience', 'Number of Projects'], 'Coefficient': multi\_model.coef\_})

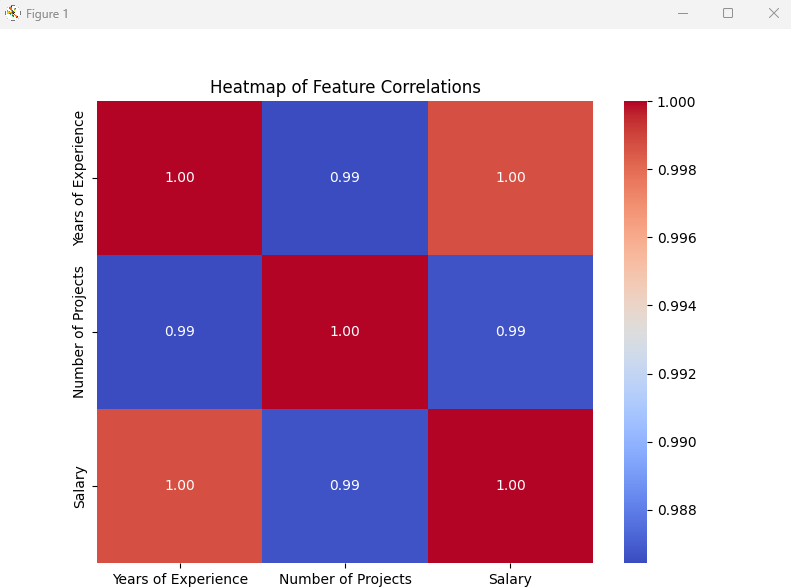
print("\nFeature Importance:\n", feature\_importance)

plt.figure(figsize=(8, 6))

sns.heatmap(df.corr(), annot=True, cmap="coolwarm", fmt=".2f") plt.title("Heatmap of Feature Correlations")

plt.show()





**Conclusion**

1. **Simple Linear Regression** is a good fit (R² = 0.99).
2. **Multiple Linear Regression** improves accuracy (R² = 0.99).
3. Both models show **Years of Experience has a strong influence on Salary**.