Project Report: Simple Linear Regression on Salary Data

1. Project Title

Prediction of Salary Based on Years of Experience using Simple Linear Regression

2. Objective

The objective of this project is to build a Simple Linear Regression model that predicts an employee's salary based on their years of experience. The goal is to find the linear relationship between these two variables and evaluate the model's prediction accuracy.

3. Dataset Description

Dataset Name: Salary_Data.csv

Source: https://www.kaggle.com/datasets/karthickveerakumar/salary-data-simple-linear-regression

Attributes:

Column	Description
YearsExperience	Number of years a person has worked
Salary	Annual salary in dollars

Number of Records: 30 rows

Number of Columns: 2

4. Algorithm Used

Algorithm: Simple Linear Regression

It models the relationship between a dependent variable (Salary) and an independent variable (YearsExperience) using a straight line:

Salary = b0 + b1 × YearsExperience

Where:
- b0 = Intercept (Salary when experience = 0)
- b1 = Coefficient (change in salary for each additional year of experience)

5. Steps Followed

1. Data Loading using pandas.
2. Visualized the data to observe linear relationship.
3. Split the dataset into training and testing sets (80%-20%).
4. Trained the Linear Regression model using scikit-learn.

5. Predicted salary values for test data.
6. Evaluated model performance using MSE and R^2 score.
7. Visualized regression line with training and test data.

6. Python Libraries Used

- pandas
- numpy
- matplotlib
- scikit-learn

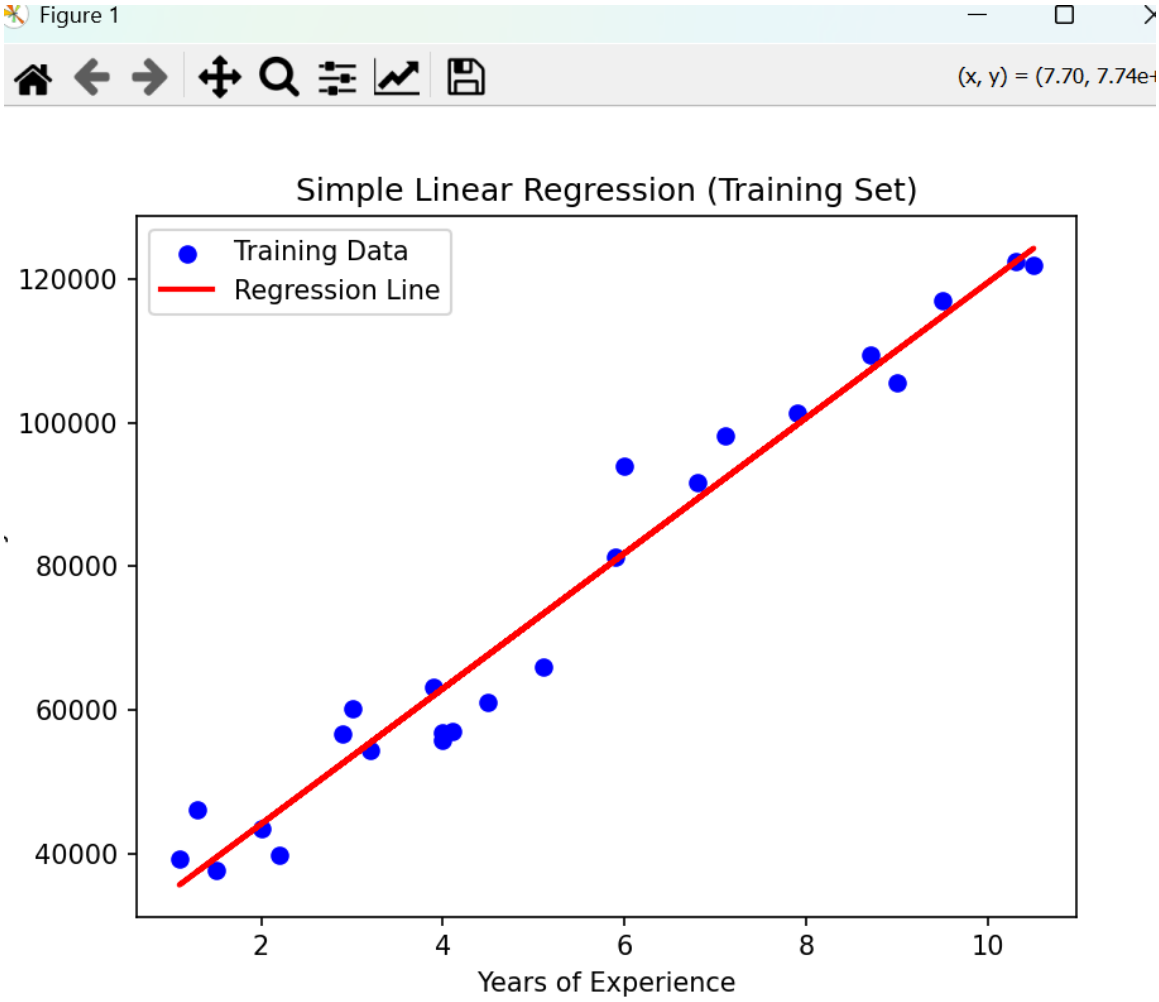
7. Results

Metric	Value (Approx.)
Intercept (b_0)	25792.20
Coefficient / Slope (b_1)	9449.96
Mean Squared Error	31270951.72
R^2 Score	0.98

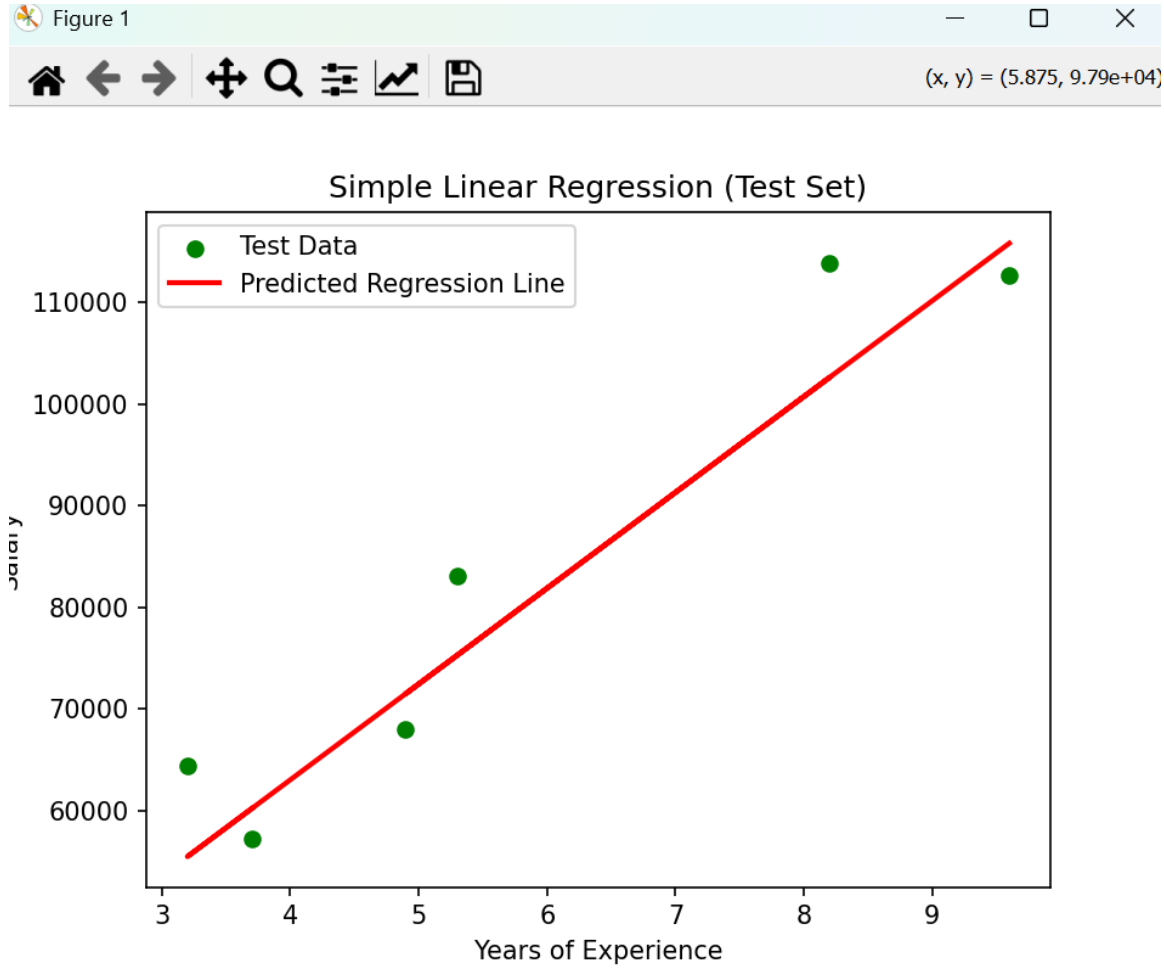
Interpretation: For every additional 1 year of experience, the salary increases by about $9,449.96. The R^2 score of 0.98 indicates the model explains 98% of the variance in salary.

8. Graphical Results

a) Training Set Visualization - Actual data (blue) and regression line (red).



b) Test Set Visualization - Predicted salaries (red line) over actual test points (green).



9. Conclusion

The Simple Linear Regression model effectively predicts salary based on years of experience. There is a strong positive linear relationship between experience and salary, as shown by the high R^2 score. This model can serve as a simple yet accurate predictor of salary.

10. Future Scope

- Add more features such as education or company size.
- Use Multiple Linear Regression for better accuracy.
- Apply Polynomial Regression for non-linear trends.
- Deploy as a web-based salary prediction application.

11.code

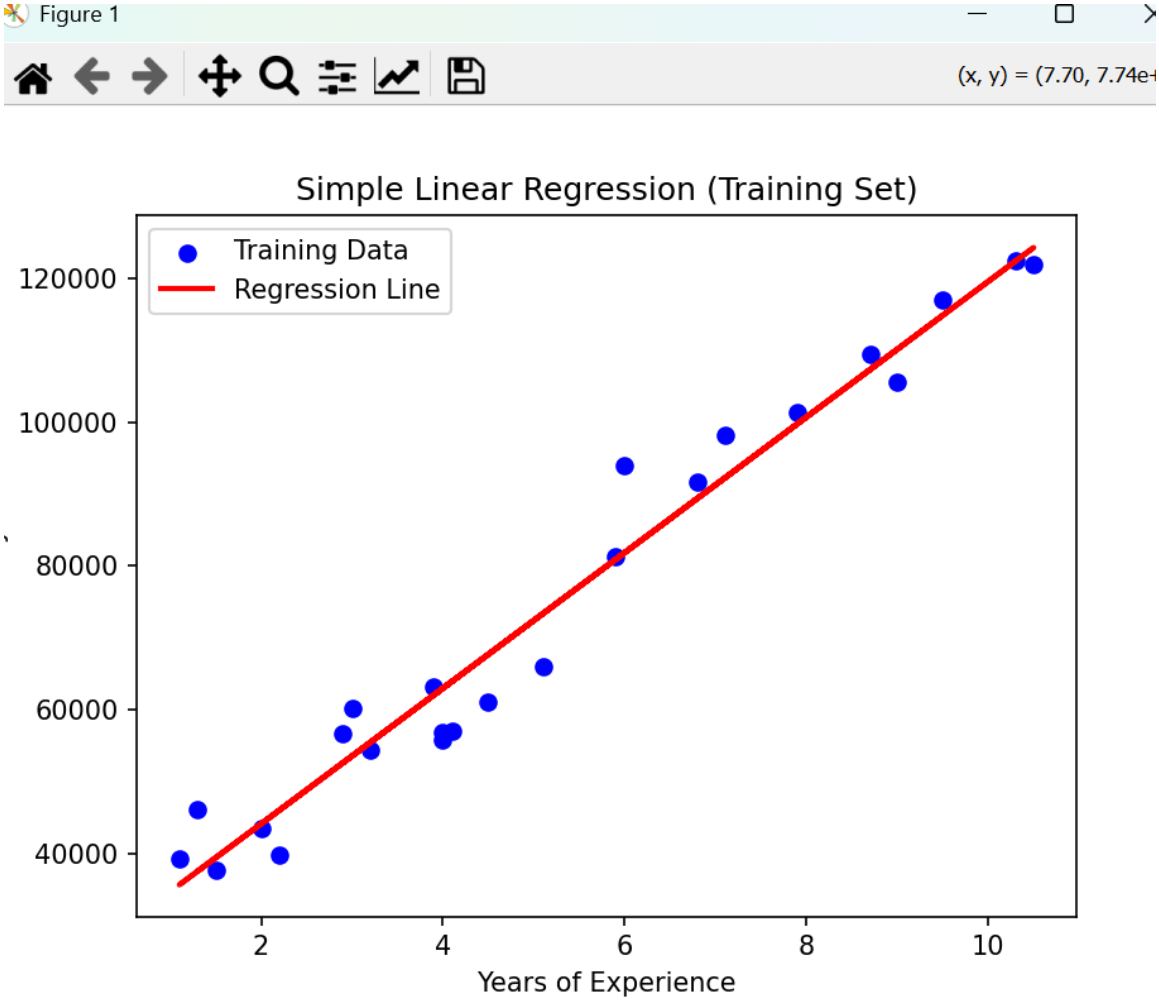
```
# ============================================

# Simple Linear Regression on Salary_Data.csv
```

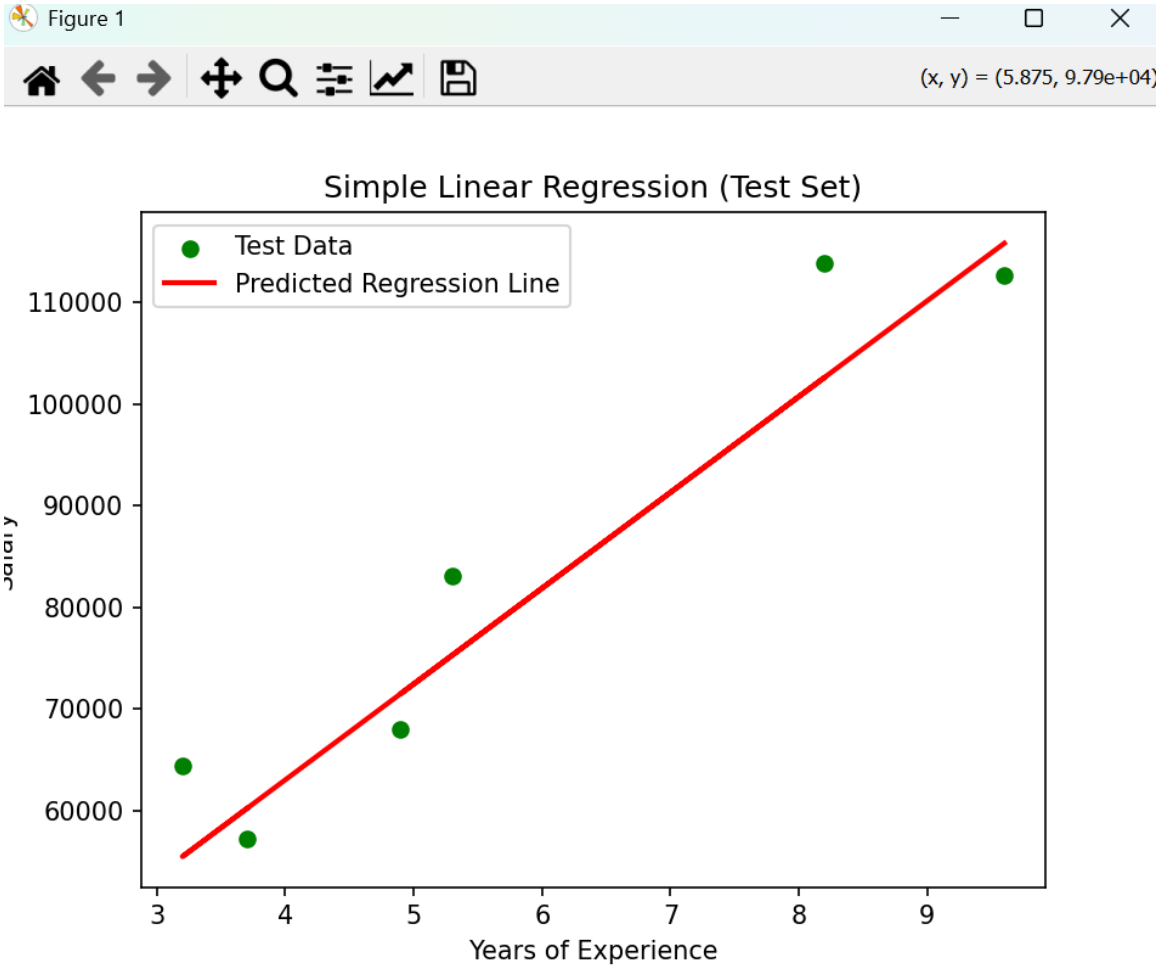
```
# ==========================================

# Step 1: Import required libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Step 2: Load the dataset
# Make sure Salary_Data.csv is in the same folder as this script
data = pd.read_csv("Salary_Data.csv")

# Display first few rows
print("First 5 rows of the dataset:")
print(data.head())

# Step 3: Separate features (X) and target (y)
X = data[['YearsExperience']]  # independent variable
y = data['Salary']             # dependent variable

# Step 4: Split data into training and testing sets (80% - 20%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Step 5: Create and train the Linear Regression model
```

```
model = LinearRegression()
model.fit(X_train, y_train)

# Step 6: Predict salaries for test data
y_pred = model.predict(X_test)

# Step 7: Evaluate the model
print("\nModel Performance:")
print(f"Intercept (b0): {model.intercept_:.2f}")
print(f"Coefficient / Slope (b1): {model.coef_[0]:.2f}")
print(f"Mean Squared Error: {mean_squared_error(y_test, y_pred):.2f}")
print(f"R² Score: {r2_score(y_test, y_pred):.2f}")

# Step 8: Visualize Training Data with Regression Line
plt.scatter(X_train, y_train, color='blue', label='Training Data')
plt.plot(X_train, model.predict(X_train), color='red', linewidth=2, label='Regression Line')
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Simple Linear Regression (Training Set)")
plt.legend()
plt.show()

# Step 9: Visualize Test Data with Regression Line
```

```
plt.scatter(X_test, y_test, color='green', label='Test Data')
plt.plot(X_test, y_pred, color='red', linewidth=2, label='Predicted Regression Line')
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Simple Linear Regression (Test Set)")
plt.legend()
plt.show()
```

12.output screenshots






```
[Running] python -u "c:\Users\DELL\Documents\mtechai courses\apr project\linearregression.py"
First 5 rows of the dataset:
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0

Model Performance:
Intercept (b0): 25321.58
Coefficient / Slope (b1): 9423.82
Mean Squared Error: 49830096.86
R Score: 0.90
```