Day 8 (AI&ML Journey) Linear Regression

Required Libraries Numpy sklearn.linear_model sklearn.model_selection (train_test_split)

Step 1: Import libraries

In [4]:
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
```

Loading Data (1-d)

In [5]:
```
X = np.array([[1], [2], [3], [4]])
y = np.array([2, 4, 5, 8])
model = LinearRegression()
model.fit(X, y)
prediction = model.predict([[5]])
print(prediction)
```

```
[9.5]
```

Loading another form of data

In [6]:
```
# Step 2: Sample dataset (Years of Experience vs Salary)
X = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9]]).reshape(-1, 1)
y = np.array([35000, 37000, 40000, 43000, 50000, 52000, 55000, 58000, 60000])
```

In [7]:
```
# Step 3: Train-test split (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [8]:
```
# Step 4: Create and train the model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[8]:

```
LinearRegression()
```

In [9]:

```
# Step 5: Make predictions
y_pred = model.predict(X_test)
```

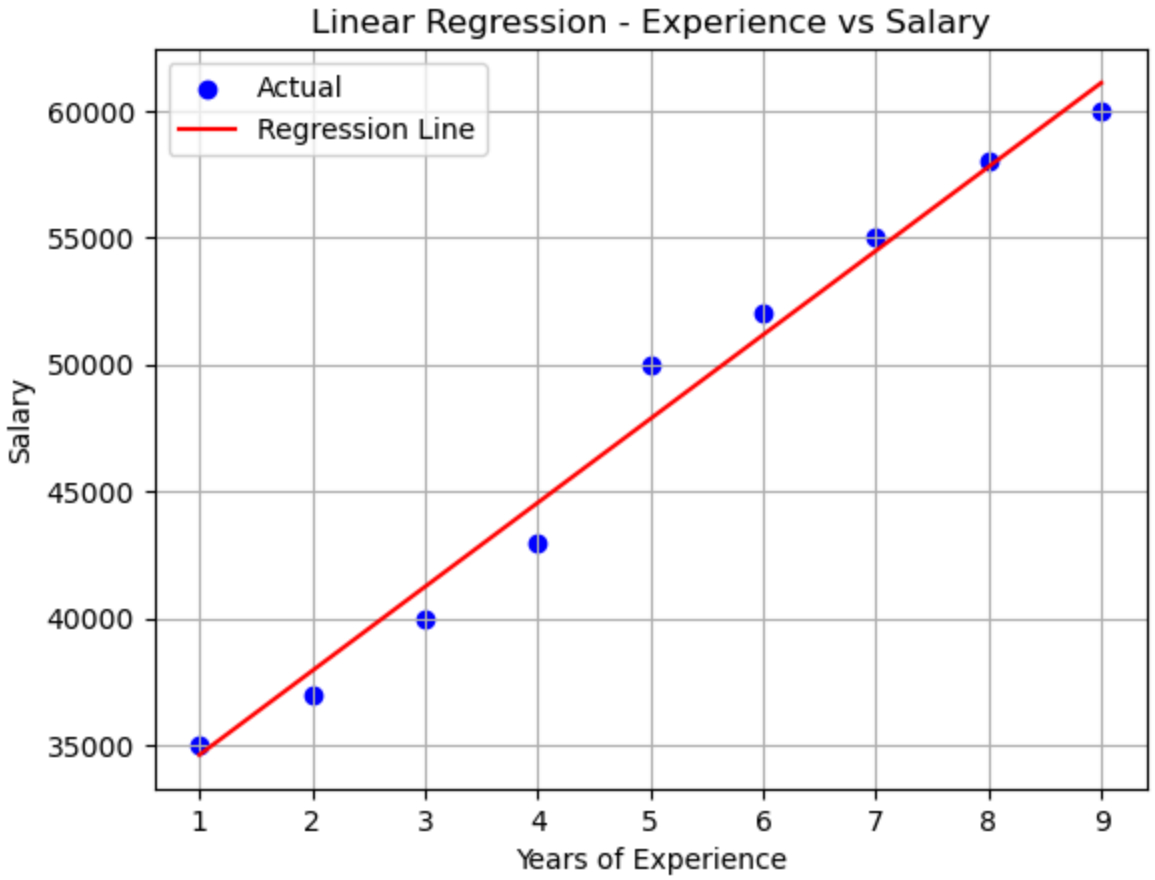
In [11…

```
# Step 6: Evaluate the model
print("R² Score:", model.score(X_test, y_test))
print("Slope (m):", model.coef_[0])
print("Intercept (b):", model.intercept_)
```

```
R² Score: 0.9958813457355731
Slope (m): 3309.5238095238083
Intercept (b): 31309.523809523813
```

In [12…

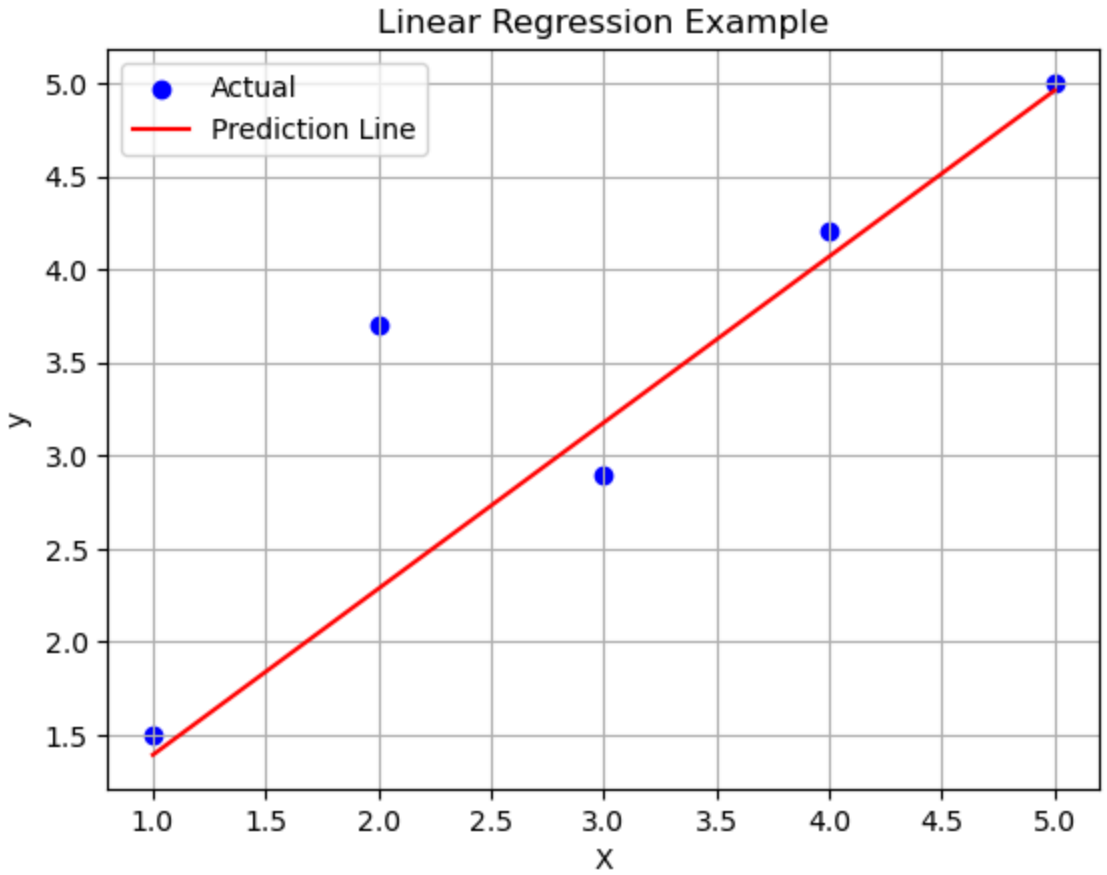
```
# Step 7: Visualize
plt.scatter(X, y, color='blue', label='Actual')
plt.plot(X, model.predict(X), color='red', label='Regression Line')
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.title("Linear Regression - Experience vs Salary")
plt.legend()
plt.grid(True)
plt.show()
```



In [3]:
```
# Sample data
X = np.array([[1], [2], [3], [4], [5]])  # Feature (2D)
y = np.array([1.5, 3.7, 2.9, 4.2, 5.0])  # Target (1D)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict
```

```
y_pred = model.predict(X_test)

# Evaluate
score = model.score(X_test, y_test)
print("R² Score:", score)
print("Slope:", model.coef_)
print("Intercept:", model.intercept_)

# Plot
plt.scatter(X, y, color='blue', label='Actual')
plt.plot(X, model.predict(X), color='red', label='Prediction Line')
plt.legend()
plt.xlabel("X")
plt.ylabel("y")
plt.title("Linear Regression Example")
plt.grid(True)
plt.show()
```

```
C:\Users\DELL\anaconda3\Lib\site-packages\sklearn\metrics\_regression.py:1211: UndefinedMetricWarning:
R^2 score is not well-defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)
R² Score: nan
Slope: [0.89142857]
Intercept: 0.502857142857144
```



In [14…

```python
#sample project
from sklearn.linear_model import LinearRegression
from sklearn.datasets import fetch_california_housing
data = fetch_california_housing()

X = data.data
y = data.target

model = LinearRegression()
model.fit(X, y)

print("Score:", model.score(X, y))
```

Score: 0.6062326851998051