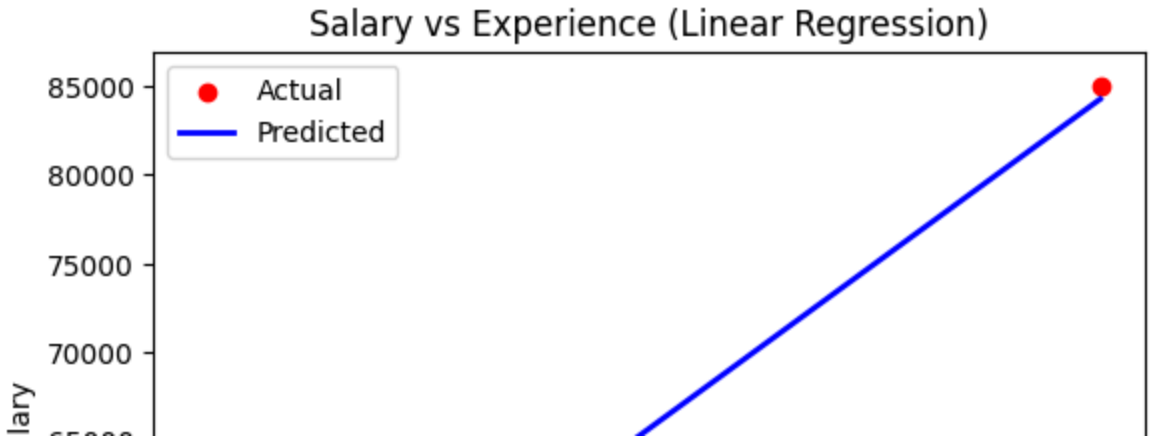
Name: Kshitij Jadhav Class: TY-3 Batch: B Roll No.: 24 Expt: 6

Implementation of Prediction algorithm (linear regression)

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

data = {
    'YearsExperience': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Salary': [35000, 40000, 50000, 55000, 60000, 65000, 70000, 80000, 85000, 90000]
}
df = pd.DataFrame(data)

X = df[['YearsExperience']]  # 2D array
y = df['Salary']             # 1D array

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

model = LinearRegression()

model.fit(X_train, y_train)
y_pred = model.predict(X_test)

print("Slope (Coefficient):", model.coef_[0])
print("Intercept:", model.intercept_)

print("Mean Squared Error:", mean_squared_error(y_test, y_pred))
print("R-squared Score:", r2_score(y_test, y_pred))

plt.scatter(X_test, y_test, color='red', label='Actual')
plt.plot(X_test, y_pred, color='blue', linewidth=2, label='Predicted')
plt.title("Salary vs Experience (Linear Regression)")
plt.xlabel("Years of Experience")
plt.ylabel("Salary")
plt.legend()
plt.show()
```

```
Slope (Coefficient): 6174.16829745597
Intercept: 28688.845401174163
Mean Squared Error: 4164783.759253391
R-squared Score: 0.9864007060922338
```



Review Questions: Q.1) What are the key steps involved in implementing a simple linear regression model using python and scikit-learn?

Ans: Steps:

1.Import libraries import pandas as pd, from sklearn.linear_model import LinearRegression, etc.

2.Prepare data Load or create dataset → separate into X (input) and y (output).

3.Split data Use train_test_split() to divide data into training and testing sets.

4.Train model Create a LinearRegression() object and call .fit(X_train, y_train).

5.Predict Use .predict(X_test) to get predictions.

6.Evaluate Use mean_squared_error() and r2_score() to check model performance.

7.Visualize Plot actual vs. predicted values using matplotlib.

Q.2) How can you evaluate the performance of the linear regression model in Python? List and explain at least two metrics.

Ans: We can evaluate a linear regression model using these two key metrics: 1.Mean Squared Error (MSE):

Measures the average squared difference between actual and predicted values.

Lower MSE = better fit.

2.R-squared Score (R^2): Represents how well the model explains the variability of the output. $R^2 = 1$ is perfect, $R^2 = 0$ means no explanatory power.

Q.3) What is the role of the train_test_spilt() function in building a linear regression model and why is it important?

Ans: The train_test_split() function splits the dataset into training and testing sets.

Role: Train the model on one part (training set). Test its performance on unseen data (testing set).

Importance: It helps check if the model generalizes well and avoids overfitting.

Github:https://github.com/yugsgithub/DWM-Expts