Step-by-Step Roadmap to Solve Simple Linear Regression Problems

Step 1: Understand the Problem Statement

Before starting, clearly understand:

- What is the **target variable** (dependent variable)?
- What is the **feature** (independent variable)?
- Is the problem regression-based?

Example Problem Statement:
"Predict a person's salary based on years of experience."

Step 2: Collect & Load the Dataset

You can get the dataset from a CSV file, database, or an online source.

Example Code: Load Dataset

```
import pandas as pd


# Load dataset

df = pd.read_csv("salary_data.csv")


# Display first few rows

print(df.head())
```

Step 3: Perform Exploratory Data Analysis (EDA)

EDA helps understand patterns, detect missing values, and analyze correlations.

Check Dataset Information

```
# Check dataset structure

print(df.info())


# Check for missing values

print(df.isnull().sum())


# Summary statistics

print(df.describe())
```

Visualizing the Relationship

```
import matplotlib.pyplot as plt


# Scatter plot
plt.scatter(df['YearsExperience'], df['Salary'], color='blue')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Years of Experience vs Salary')
plt.show()
```

Step 4: Split Data into Train & Test Sets

```
from sklearn.model_selection import train_test_split
# Define independent variable (X) and dependent variable (y)
X = df[['YearsExperience']]
y = df['Salary']
# Split data (80% train, 20% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Step 5: Train the Simple Linear Regression Model

```
from sklearn.linear_model import LinearRegression


# Initialize model
model = LinearRegression()


# Train the model
model.fit(X_train, y_train)
```

Step 6: Make Predictions

```
# Predict on test data
y_pred = model.predict(X_test)
```

Step 7: Evaluate Model Performance

Use evaluation metrics to check how well the model performs.

Metrics to Check

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

```
# Calculate metrics
mse = mean_squared_error(y_test, y_pred)
mae = mean_absolute_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Print results
print(f"Mean Squared Error: {mse:.2f}")
print(f"Mean Absolute Error: {mae:.2f}")
print(f"R² Score: {r2:.2f}")
```

Step 8: Visualize the Regression Line

```
# Plot actual vs predicted values
plt.scatter(X_test, y_test, color='blue', label='Actual Data')
plt.plot(X_test, y_pred, color='red', label='Regression Line')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.title('Simple Linear Regression')
plt.legend()
plt.show()
```

Step 9: Save & Deploy the Model

Once satisfied, save the model for future use.

```
import joblib

# Save model
joblib.dump(model, "salary_prediction_model.pkl")

# Load model
loaded_model = joblib.load("salary_prediction_model.pkl")
```