Task 3: Linear Regression Model

Problem Statement

Build a linear regression model to predict a target variable.

Steps Completed

1. **Dataset Selection**: The Salary dataset (`Salary_Data.csv`) was chosen for this task. This dataset contains 'YearsExperience' as the independent variable and 'Salary' as the dependent variable, making it suitable for simple linear regression.
2. **Data Splitting**: The dataset was split into training and testing sets using `train_test_split` from `sklearn.model_selection`. 80% of the data was used for training and 20% for testing.
3. **Model Training**: A `LinearRegression` model from `sklearn.linear_model` was initialized and trained on the training data (`X_train`, `y_train`).
4. **Prediction**: The trained model was used to make predictions on the test set (`X_test`).
5. **Model Evaluation**: The model's performance was evaluated using:
    - **Mean Squared Error (MSE)**: Measures the average squared difference between the estimated values and the actual value. Lower MSE indicates a better fit.
    - **R-squared (R2) Score**: Represents the proportion of the variance in the dependent variable that is predictable from the independent variable(s). A higher R2 score indicates a better fit.
6. **Visualization**: A scatter plot was generated to visualize the actual vs. predicted salaries, along with the regression line, to provide a clear understanding of the model's fit.

Code Implementation

The linear regression model implementation is in the `linear_regression.py` script.

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

```python
def linear_regression_model(input_filepath):
    df = pd.read_csv(input_filepath)

    X = df[["YearsExperience"]]
    y = df["Salary"]

    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.2, random_state=42)

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

    y_pred = model.predict(X_test)

    mse = mean_squared_error(y_test, y_pred)
    r2 = r2_score(y_test, y_pred)

    print(f"\n--- Model Performance ---")
    print(f"Mean Squared Error (MSE): {mse:.2f}")
    print(f"R-squared (R2): {r2:.2f}")

    plt.figure(figsize=(10, 6))
    plt.scatter(X_test, y_test, color="blue", label="Actual
Salary")
    plt.plot(X_test, y_pred, color="red", label="Predicted
Salary")
    plt.title("Salary vs. Years of Experience (Linear
Regression)")
    plt.xlabel("Years of Experience")
    plt.ylabel("Salary")
    plt.legend()
    plt.grid(True)
    plt.savefig("linear_regression_plot.png")
    plt.show()

if __name__ == "__main__":
    input_file = "../Salary_Data.csv"
    linear_regression_model(input_file)
```

Output

Upon execution, the script prints the Mean Squared Error and R-squared score of the model. It also generates a `linear_regression_plot.png` file, visualizing the actual vs. predicted salaries and the regression line.

Console output includes:

```
 --- Model Performance ---
Mean Squared Error (MSE): 49830096.86
R-squared (R2): 0.90
```

This output demonstrates the model's performance and its ability to predict salary based on years of experience.