Experiment - 6

Objective:
Regression: Performing Simple Linear Regression over a salary dataset and predict salaries according to their experience years

Linear Regression

1. Loading data

Firstly, we will use the Python Pandas library to read our CSV data.

Code :

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import linear_model
import sklearn.metrics as sm
import matplotlib.pyplot as plt

data = pd.read_csv(r"C:\Users\KUNAL
KUNDU\OneDrive\Desktop\GHRCEM - Third Year\MLA-
Practicals\Dataset/Salary_Data.csv")
X = data['YearsExperience']
y = data['Salary']
```

2. Splitting data into a training set and a test set

Prior to building a linear model, we need to prepare a training set and a test set (part of standard procedure in a machine learning workflow). The training set will be used to train the model whereas the test set will be used to assess the performance of the trained model in predicting the result from unseen data.
We can use the *Python scikit-learn* ***train_test_split*** function to randomly split our data into a training and test set.

Code :

```
X_train, X_test, y_train, y_test = train_test_split(X,y,
test_size=0.3, random_state=42)
```

3. Data Transformation

Python scikit-learn only accepts the training and test data in a 2-dimensional array format. We have to perform data transformation on our training set and test set.

Code :

```
X_train = np.array(X_train).reshape((len(X_train),1))
y_train = np.array(y_train).reshape((len(y_train),1))

X_test = np.array(X_test).reshape(len(X_test), 1)
y_test = np.array(y_test).reshape(len(y_test), 1)
```

4. Training Model
Now we are ready to train our linear model.

Code:

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

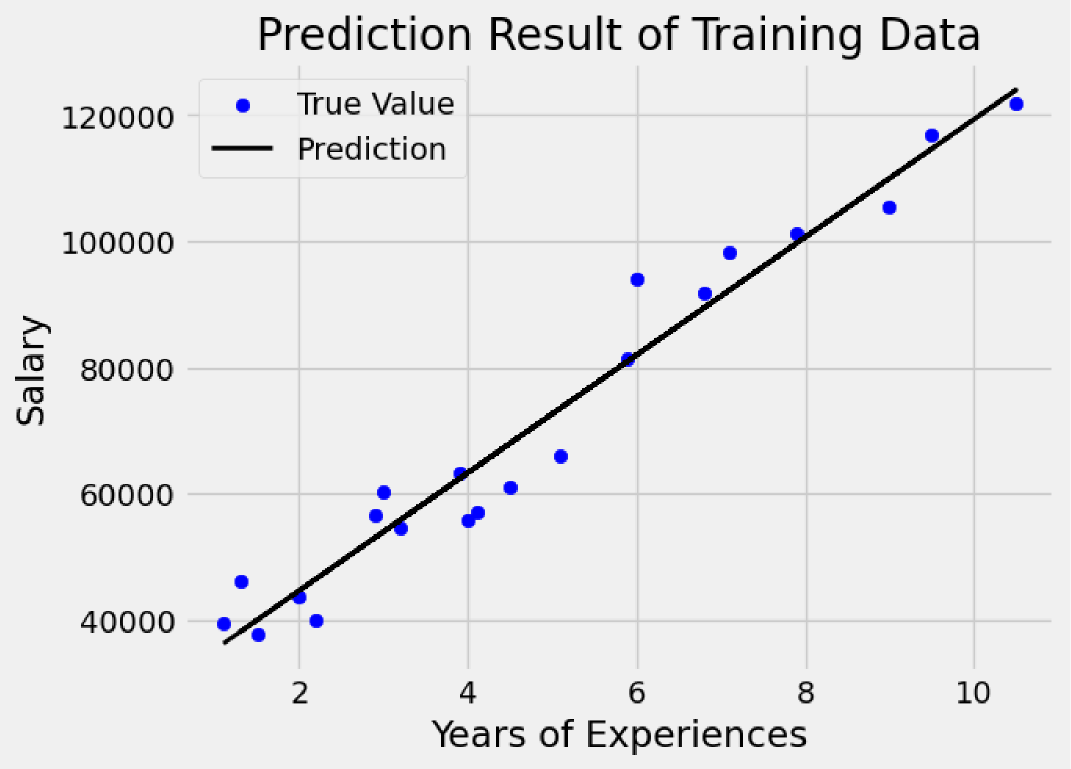
5. Predicting Salary using Linear Model
At this stage, we have trained a linear model and we first use it to predict the salary on our training set to see how well it fit on the data.

Code:

```
y_train_pred = model.predict(X_train)

plt.figure()
plt.scatter(X_train, y_train, color='blue', label="True
Value")
plt.plot(X_train, y_train_pred, color='black', linewidth=2,
label="Prediction")
plt.xlabel("Years of Experiences")
plt.ylabel("Salary")
plt.title('Prediction Result of Training Data')
plt.legend()
plt.show()
```

Output:



In general, the linear model fits well on the training data. This shows a linear relationship between the salary and years of experience.

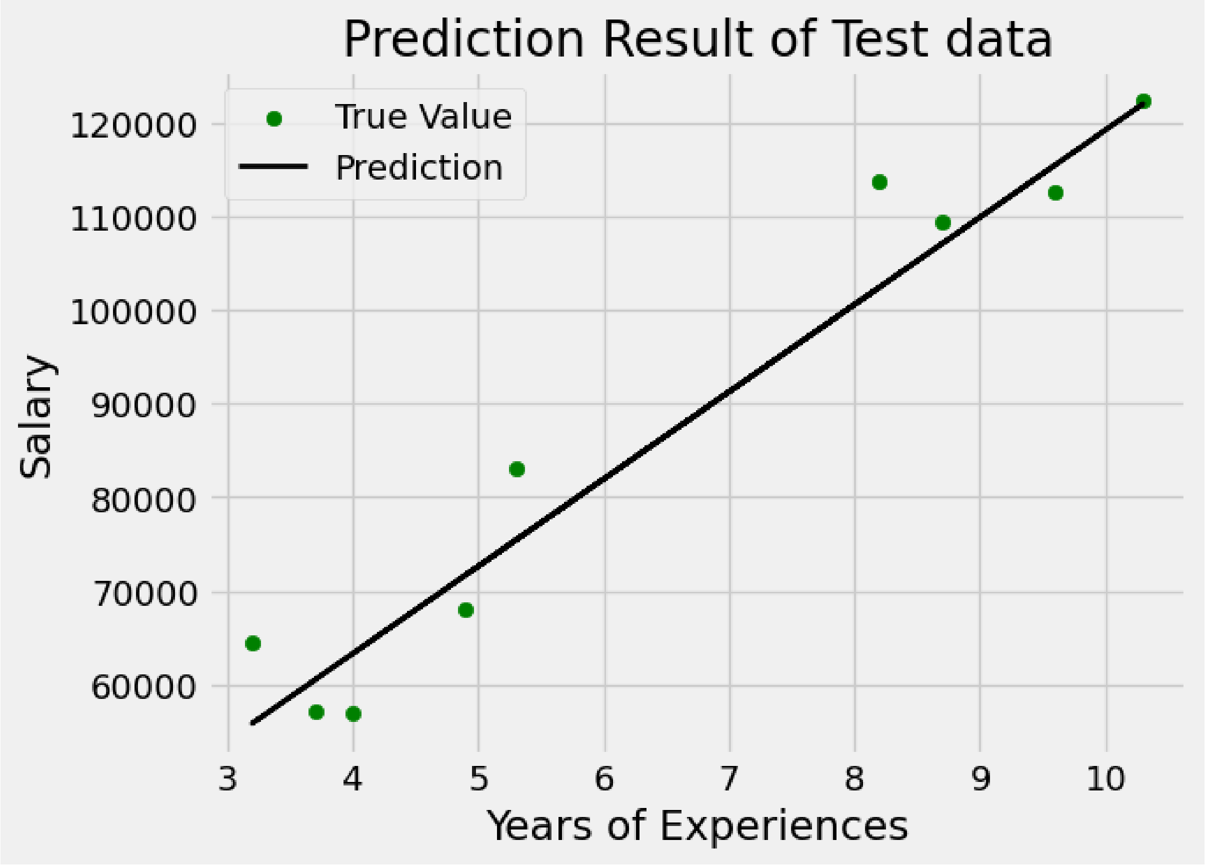
Now, we need to check if the linear model can perform well on our test set (unknown data).

Code:

```
y_test_pred = model.predict(X_test)

plt.figure()
plt.scatter(X_test, y_test, color='green', label='True Value')
plt.plot(X_test, y_test_pred, color='black', linewidth=2,
label='Prediction')
plt.xlabel("Years of Experiences")
plt.ylabel("Salary")
plt.title('Prediction Result of Test data')
plt.legend()
plt.show()
```

Output:



The graph shows that our linear model can fit quite well on the test set. We can observe a linear pattern of how the amount of salary is increased by the years of experience.

6. Model Evaluation

The previous section uses a graphical approach to evaluate the performance of our linear model which can sometimes be quite subjective in our judgment. Here we will use some quantitative methods to obtain a more precise performance evaluation of our linear model.

We will use three types of quantitative metrics:

- Mean Square Error — The average of the squares of the difference between the true values and the predicted values. The lower the difference the better the performance of the model. This is a common metric used for regression analysis.
- Explained Variance Score — A measurement to examine how well a model can handle the variation of values in the dataset. A score of 1.0 is the perfect score.
- R2 Score — A measurement to examine how well our model can predict values based on the test set (unknown samples). The perfect score is 1.0.

Code:

```
print("Mean squared error =",
round(sm.mean_squared_error(y_test, y_test_pred), 2))
print("Explain variance score =",
round(sm.explained_variance_score(y_test, y_test_pred), 2))
print("R2 score =", round(sm.r2_score(y_test, y_test_pred),
2))
```

Output:

```
Mean squared error = 37784662.47
Explain variance score = 0.95
R2 score = 0.94
```

Conclusions

We have managed to build a simple linear model to predict salary based on years of working experience. Based on our linear model, we can conclude that our salary is grown with our years of working experience and there is a linear relationship between them. We can use our linear model to predict the salary by giving input of years of experience.