

Regression Analysis and Plot Interpretation

✓ Implementation of Simple Linear Regression Algorithm

Problem Statement for Simple Linear Regression: Here we are taking a dataset that has two variables: salary (dependent variable) and experience (Independent variable). We have to predict Salary based on Experience.

✓ Import Library

```
import numpy as np
import matplotlib.pyplot as plt
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
```

✓ Import Dataset

```
dataset = pd.read_csv('/content/Experience-Salary.csv')
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, 1].values
```

✓ Splitting the dataset into the Training set and Test set

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 1/3, random_state :
```

✓ Fitting simple linear regression to the training set

```
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

```
▼ LinearRegression
LinearRegression()
```

✓ **Predicting the test set result **

```
y_pred = regressor.predict(X_test)
```

✓ Visualising the training set result

```
plt.scatter(X_train, y_train, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Training set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



TT B I <> 🔗 🖼️ ⌵ ⌶ ⌷ — Ψ 😊 📄

```
### **Visualising the test set result**
```

Visualising the test set result

```
plt.scatter(X_test, y_test, color = 'red')
plt.plot(X_train, regressor.predict(X_train), color = 'blue')
plt.title('Salary vs Experience (Test set)')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.show()
```



```
exp = int(input("enter years of experience to get amount of salary "))
sal = regressor.predict([[exp]])
sal
```

```
enter years of experience to get amount of salary 10
array([13.19333317])
```