Name : Muskan Kumari

Roll.No: 21MCA035

DS Case Study

Simple linear regression is a statistical method that we can use to find a relationship between two variables and make predictions. The two variables used are typically denoted as y and x. The independent variable, or the variable used to predict the dependent variable is denoted as x. The dependent variable, or the **outcome/output**, is denoted as y.

let's say we have a scatter plot showing how years of experience affect salaries. Imagine drawing a line to predict the trend.

Simple Linear Regression Using Python

we will be using salary data from Kaggle. The data consists of two columns, years of experience and the corresponding salary.

Coding

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


data = pd.read_csv('Salary_Data.csv')
x = data['YearsExperience']
y = data['Salary']
print(data.head())
```

output:

```
   YearsExperience   Salary
0              1.1  39343.0
1              1.3  46205.0
2              1.5  37731.0
3              2.0  43525.0
4              2.2  39891.0
```

Above is a scatter plot showing our data. We can see a positive linear relationship between Years of Experience and Salary, meaning that as a person gains more experience, they also get paid more.

Calculating the Regression Line

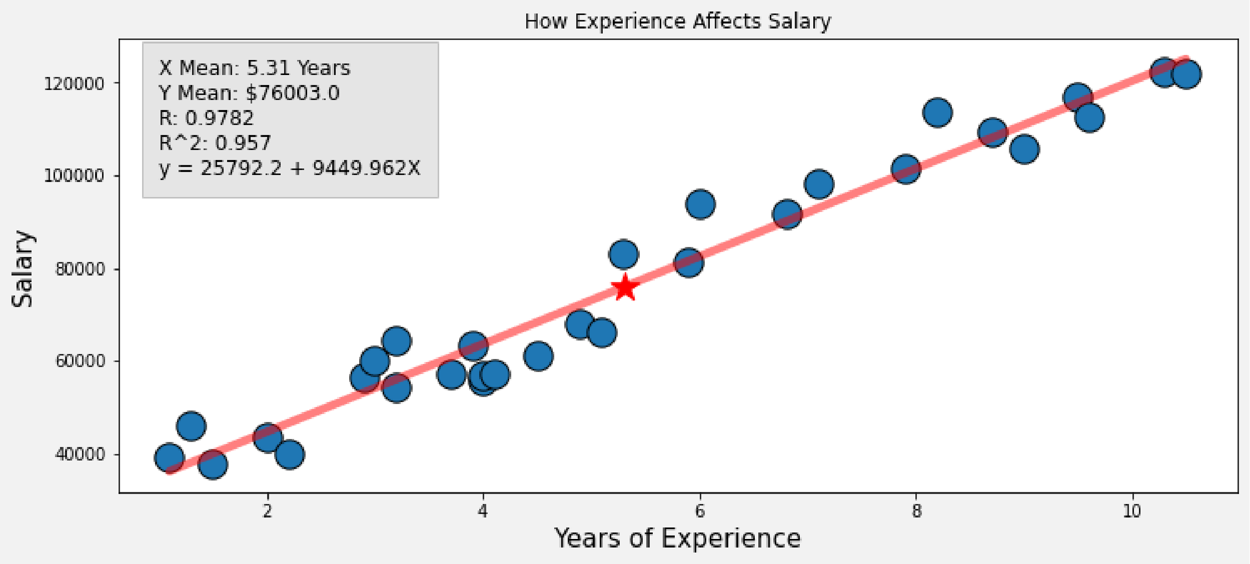
```
def linear_regression(x, y):
    N = len(x)
    x_mean = x.mean()
    y_mean = y.mean()

    B1_num = ((x - x_mean) * (y - y_mean)).sum()
    B1_den = ((x - x_mean)**2).sum()
    B1 = B1_num / B1_den

    B0 = y_mean - (B1*x_mean)

    reg_line = 'y = {} + {}β'.format(B0, round(B1, 3))

    return (B0, B1, reg_line)


N = len(x)
x_mean = x.mean()
y_mean = y.mean()

B1_num = ((x - x_mean) * (y - y_mean)).sum()
B1_den = ((x - x_mean)**2).sum()
B1 = B1_num / B1_den


B0 = y_mean - (B1 * x_mean)

def corr_coef(x, y):
    N = len(x)

    num = (N * (x*y).sum()) - (x.sum() * y.sum())
    den = np.sqrt((N * (x**2).sum() - x.sum()**2) * (N * (y**2).sum() -
y.sum()**2))
    R = num / den
    return R

B0, B1,
reg_line = linear_regression(x, y)
print('Regression Line: ', reg_line)
R = corr_coef(x, y)
print('Correlation Coef.: ', R)
print('"Goodness of Fit": ', R**2)
```

`output:`

```
Regression Line:  y = 25792.20019866869 + 9449.962β
Correlation Coef.:  0.97824161848876
"Goodness of Fit":  0.9569566641435087
```

Plotting the Regression Line

```
plt.figure(figsize=(12,5))
plt.scatter(x, y, s=300, linewidths=1, edgecolor='black')
text = '''X Mean: {} Years
Y Mean: ${}
R: {}
R^2: {}
y = {} + {}X'''.format(round(x.mean(), 2),
                       round(y.mean(), 2),
                       round(R, 4),
                       round(R**2, 4),
                       round(B0, 3),
                       round(B1, 3))
plt.text(x=1, y=100000, s=text, fontsize=12, bbox={'facecolor': 'grey',
'alpha': 0.2, 'pad': 10})
plt.title('How Experience Affects Salary')
plt.xlabel('Years of Experience', fontsize=15)
plt.ylabel('Salary', fontsize=15)
plt.plot(x, B0 + B1*x, c = 'r', linewidth=5, alpha=.5, solid_capstyle='round')
plt.scatter(x=x.mean(), y=y.mean(), marker='*', s=10**2.5, c='r') # average
point
def predict(B0, B1, new_x):
    y = B0 + B1 * new_x
    return y
```



Now we can use our calculations of the regression line to make predictions with new data that we come across. To create the `predict()` function, we just follow the formula for the simple linear regression line and plug in the values that we calculated as well as the new X value. This function will return the prediction y.