VU1F19200164
Tejas Shinde
BE Comps-A/A

ML Exp No. 1

Aim: To understand and implement the linear Regression algorithm

Theory:

Linear Regression is machine learning algorithm based on supervised learning.
A LR model predicts values based on independent variable it was initially trained on via a line of best fit that can be used to extrapolate new values based on dependant variables.

It is used for finding out relationships between variable and forecasting. Fits a line minimizing the sum of mean-squared error for each data point.

General form:

$$y = m1x1 + m2x2 + m3x3 + \dots + mnxn + c + e$$

where,
y = dependent variable
x_i = Independent variable
e = random / stochastic error term.

* Library used :-

- Pandas: It is derived from the word Panel data. It can perform five significant steps required for processing and analysis of data i.e load, manipulate, prepare, model and analyze.

- NumPy: It stands for 'Numerical Python'. It consist of multidimensional array objects and collection of routines for processing of array.

- Linear Regression: It uses relationship between data-points to draw a straight line through all them.

- Matplotlib: It uses to create 2D graphs and plots by using python scripts.

- Seaborn: It is used for data visualization and exploratory data analysis.

- Sklearn.metrics: It implements several loss, score, and utility function to measure classification performance.

* Dataset: 'Salary_Data.csv'

Conclusion: Hence, we successfully implemented linear regression algorithm.

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
dataset = pd.read_csv('Salary_Data.csv')
dataset.head()
```

Out[2]:

	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

In [3]:
```
dataset.shape
```

Out[3]: (30, 2)

In [4]:
```
X = dataset.iloc[:, :-1].values  #independent variable array
y = dataset.iloc[:,1].values  #dependent variable vector
```

In [5]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=1/3,random_state=0)
```

In [6]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train,y_train)
```

Out[6]:
```
LinearRegression()
```

In [7]:
```
y_pred = regressor.predict(X_test)
y_pred
```

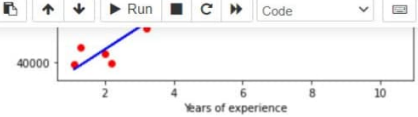
Out[7]:
```
array([ 40835.10590871, 123079.39940819,  65134.55626083,  63265.36777221,
       115602.64545369, 108125.8914992 , 116537.23969801,  64199.96201652,
        76349.68719258, 100649.1375447 ])
```

In [8]:
```
y_test
```

Out[8]:
```
array([ 37731., 122391.,  57081.,  63218., 116969., 109431., 112635.,
        55794.,  83088., 101302.])
```

In [9]:
```
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("Salaries")
plt.show()
```

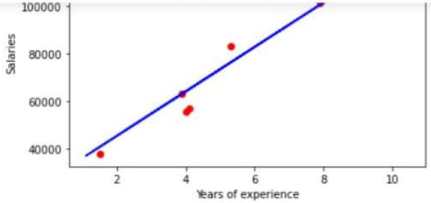


In [10]:
```
plt.scatter(X_test, y_test, color='red')
plt.plot(X_train, regressor.predict(X_train), color='blue')
plt.title("Salary vs Experience (Testing set)")

plt.xlabel("Years of experience")
plt.ylabel("Salaries")
plt.show()
```

In [11]:
```
print('Salary of employee who has experience of 12 years', regressor.predict([[12]]))
```

```
Salary of employee who has experience of 12 years [138967.5015615]
```



In [11]:
```
print('Salary of employee who has experience of 12 years', regressor.predict([[12]]))
```

```
Salary of employee who has experience of 12 years [138967.5015615]
```

In [12]:
```
from sklearn import metrics
mean_squ_error = metrics.mean_squared_error(y_test, y_pred)
mean_abs_error = metrics.mean_absolute_error(y_test, y_pred)
print("Mean Squared Error:", mean_squ_error)
print("Mean absolute Error:", mean_abs_error)
```

```
Mean Squared Error: 21026037.329511296
Mean absolute Error: 3426.4269374307123
```

In [ ]:

Type here to search
29°C ENG 20:30 01-08-2022