

## # Name:SHEIK PAREETH

## # Predicted fresher salary

In [1]:

```
#Import the libraries and put nicknames  
import pandas as pd
```

In [2]:

```
#create new variables  
#assign LHS=RHS  
dataset=pd.read_csv("Salary.csv")
```

In [3]:

```
#input and output split  
#input split  
indep=dataset[["YearsExperience"]]
```

In [4]:

```
#input and output split  
#output split  
dep=dataset[["Salary"]]
```

In [5]:

```
#Draw the graph for independent and dependent  
import matplotlib.pyplot as plt
```

In [6]:

```
#split train and test  
#take 30% of sample  
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(indep,dep,test_size=0.3,random_state=0)
```

In [7]:

```
#model creation  
#formul loaded this libraries  
from sklearn.linear_model import LinearRegression
```

In [8]:

```
#create new variables  
#formula loaded  
regressor=LinearRegression()
```

In [9]:

```
#formula substitution  
regressor.fit(X_train,y_train)
```

Out[9]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [10]:

```
#find the weight value  
weight=regressor.coef_
```

In [11]:

```
#find the bias value  
bias=regressor.intercept_
```

In [12]:

```
#check above the model is good  
#Evaluation process  
y_pred=regressor.predict(X_test)
```

In [14]:

```
#predict actual value from pandas range 0-9  
#form the table  
predActual=pd.DataFrame(index=range(0,10))
```

In [18]:

```
#cross checking process  
predActual["Actualvalue"]=y_test
```

In [19]:

```
#predict salary upto 9 values  
y_test.index=range(0,9)
```

In [20]:

```
#predict actual value and change the column name salary to pred  
y_pred_table=pd.DataFrame(y_pred,columns=["pred"])
```

In [22]:

```
#above the column is predicted this table  
predActual["predvalue"]=y_pred_table
```

In [23]:

```
#evaluation metrics process  
#find the r2 value  
#best model  
from sklearn.metrics import r2_score  
r=r2_score(y_test,y_pred)
```

In [24]:

```
#save the best model  
#this technique are used  
import pickle  
filename = "finalModel.sav"  
pickle.dump(regressor, open(filename, "wb"))
```

In [29]:

```
#create new variable read the above variables  
load_model=pickle.load(open("finalModel.sav", "rb"))
```

In [30]:

```
#predict 0 year experinced person salary  
load_model.predict([[0]])
```

Out[30]:

```
array([[26777.3913412]])
```