In [1]:
```
#pandas is a library and it contains the functions related to the database
import pandas as pd
```

In [20]:
```
# copy .csv file data intot 'data' variable
dataset=pd.read_csv("Salary_Data.csv")
```

In [3]:
```
dataset #see the content of dataset, i.e., Salary Data
```

Out[3]:

	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
5	2.9	56642.0
6	3.0	60150.0
7	3.2	54445.0
8	3.2	64445.0
9	3.7	57189.0
10	3.9	63218.0
11	4.0	55794.0
12	4.0	56957.0
13	4.1	57081.0
14	4.5	61111.0
15	4.9	67938.0
16	5.1	66029.0
17	5.3	83088.0
18	5.9	81363.0
19	6.0	93940.0
20	6.8	91738.0
21	7.1	98273.0
22	7.9	101302.0
23	8.2	113812.0
24	8.7	109431.0
25	9.0	105582.0
26	9.5	116969.0
27	9.6	112635.0
28	10.3	122391.0
29	10.5	121872.0

In [4]:
```
#copy all input column data into independant variable
independent=dataset[["YearsExperience"]]
independent
```

Out[4]:

	YearsExperience
0	1.1
1	1.3
2	1.5
3	2.0
4	2.2
5	2.9
6	3.0
7	3.2
8	3.2
9	3.7
10	3.9
11	4.0
12	4.0
13	4.1
14	4.5
15	4.9
16	5.1
17	5.3
18	5.9
19	6.0
20	6.8
21	7.1
22	7.9
23	8.2
24	8.7
25	9.0
26	9.5
27	9.6
28	10.3
29	10.5

In [6]:
```
# copy all output column data into dependant variable
dependent=dataset[["Salary"]]
dependent
```

Out[6]:

	Salary
0	39343.0
1	46205.0
2	37731.0
3	43525.0
4	39891.0
5	56642.0
6	60150.0
7	54445.0
8	64445.0
9	57189.0
10	63218.0
11	55794.0
12	56957.0
13	57081.0
14	61111.0
15	67938.0
16	66029.0
17	83088.0
18	81363.0
19	93940.0
20	91738.0
21	98273.0
22	101302.0
23	113812.0
24	109431.0
25	105582.0
26	116969.0
27	112635.0
28	122391.0
29	121872.0

In [7]:
```
#import train_test_split model fucntion from sklearn.model_selection to splite trai
from sklearn.model_selection import train_test_split
# train adn test data split done for both X and Y axis
X_train,X_test,y_train,y_test=train_test_split(independent, dependent, test_size=0.
```

In [8]:
```
# import library of LinearRegression from linear_model
from sklearn.linear_model import LinearRegression
#assign all function of LinearRegression to regressor.
regressor=LinearRegression()
# to create the model using train dataset
regressor.fit(X_train,y_train)
```

Out[8]:
```
LinearRegression
LinearRegression()
```

In [9]:
```
#assign the weight of the linear regression into weight variable
weight=regressor.coef_
weight
```

Out[9]:
```
array([[9360.26128619]])
```

In [10]:
```
# assign the calcuated bias value from the created model into variable bias
bais=regressor.intercept_
bais
```

Out[10]:
```
array([26777.3913412])
```

In [11]:
```
#predict the value for all splited test data
y_pred=regressor.predict(X_test)
```

In [12]:
```
# import the r2_score library to check r2 score by passing actual data & predicted
from sklearn.metrics import r2_score
r_score=r2_score(y_test,y_pred)
```

In [13]:
```
# read the score value & if it is around 1, then it is a good model.
r_score
```

Out[13]:
```
0.9740993407213511
```

In [14]:
```
# import pickle library to copy the created model
import pickle
# create the file name to take copy
filename="finalized_model_linear.sav"
```

In [15]:
```
#save the model into to the 'finalized_model_linear.sav' file.
pickle.dump(regressor,open(filename,'wb'))
```

In [18]:
```
#load the saved model into variable
loaded_model=pickle.load(open("finalized_model_linear.sav",'rb'))
```

```
# predit/run the model with user input 18
result=loaded_model.predict([[18]])
```

```
C:\Users\HopeAI\anaconda\Lib\site-packages\sklearn\base.py:493: UserWarning: X does
not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

In [19]:
```
# print the predicted value
result
```

Out[19]:
```
array([[195262.09449268]])
```

In [ ]:
```

```