TECH FORTUNE

we stand for excellence

A PROJECT REPORT ON

SALARY PREDICTION OF EMPLOYEE BASED ON THEIR EXPERIENCE USING LINEAR REGRESSION

By

1MS17IS062 MANJUNATHA N

MS RAMAIAH INSTITUTE OF TECHNOLOGY BANGALORE-54

Under the guidance of

Mr Rama Krishna

Director Technical

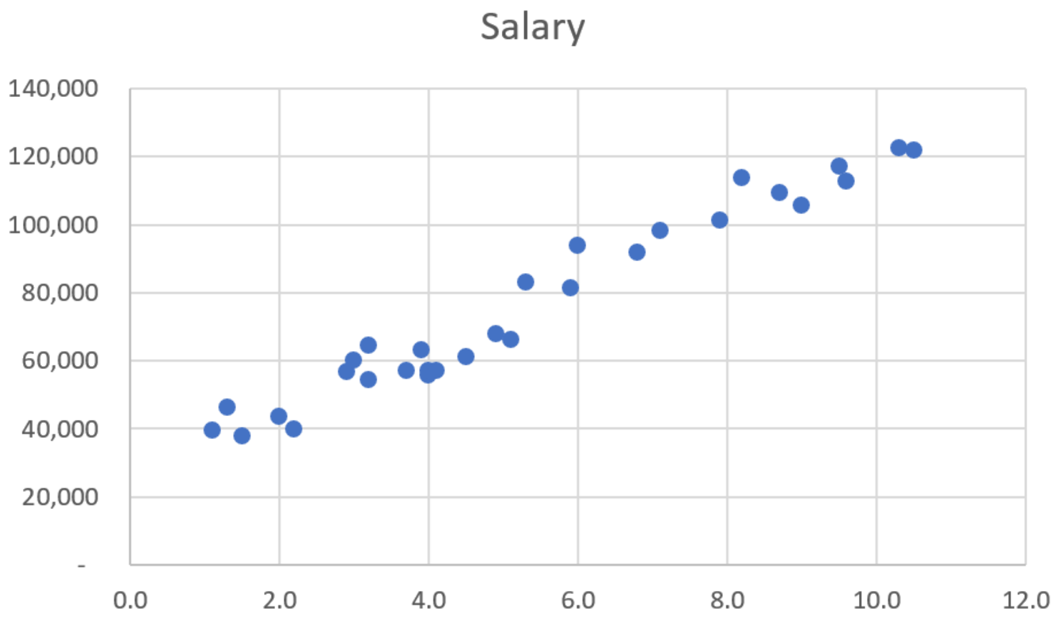
Tech Fortune Technologies Bangalore-40

PROBLEM STATEMENT

Predict Salary using Simple Linear Regression

"The scenario is you are a HR officer, you got a candidate with 5 years of experience. Then what is the best salary you should offer to him?"

Before deep dive into this problem, let's plot the data set into the plot first:

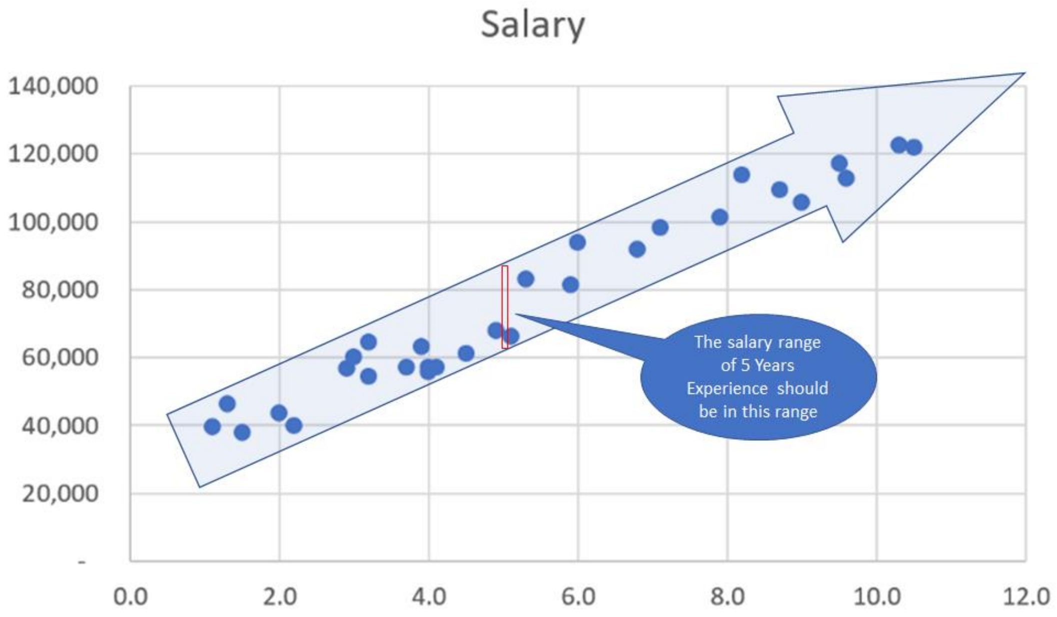


Please look at this chart carefully. Now we have a bad news: all the observations are not in a line. It means we cannot find out the equation to calculate the (y) value.

So what now? Don't worry, we have a good news for you!

Look at the Scatter Plot again before scrolling down. Do you see it?

All the points is not in a line BUT they are in a line-shape! **It's linear**



Based on our observation, we can guess that the salary range of 5 Years Experience should be in the red range. Of course, we can offer to our candidate any number in that red range. But how to pick the best number for him? It's time to use Machine Learning to predict the best salary for our candidate.

Dataset

First let’s look at the dataset. It is Salary_Data.csv and can be found https://github.com/Manjuchandan/Salary-Prediction-Using-Linear-Regression/blob/master/SalaryPrediction/salary_data.csv
It has 2 columns — “Years of Experience” and “Salary” for 30 employees in a company. So in this example, we will train a Simple Linear Regression model to learn the correlation between the number of years of experience of each employee and their respective salary. Once the model is trained, we will be able to do some sample predictions.

Project Objective

We have to predict the salary of an employee given how many years of experience they have.

VISUALISATION

Step 1: Load the Dataset

Below is the code snippet for loading the dataset.
We will be using the pandas data frame.
Here X is the independent variable which is the “Years of Experience”
and y is the dependent variable which is the “Salary”

So for X, we specify

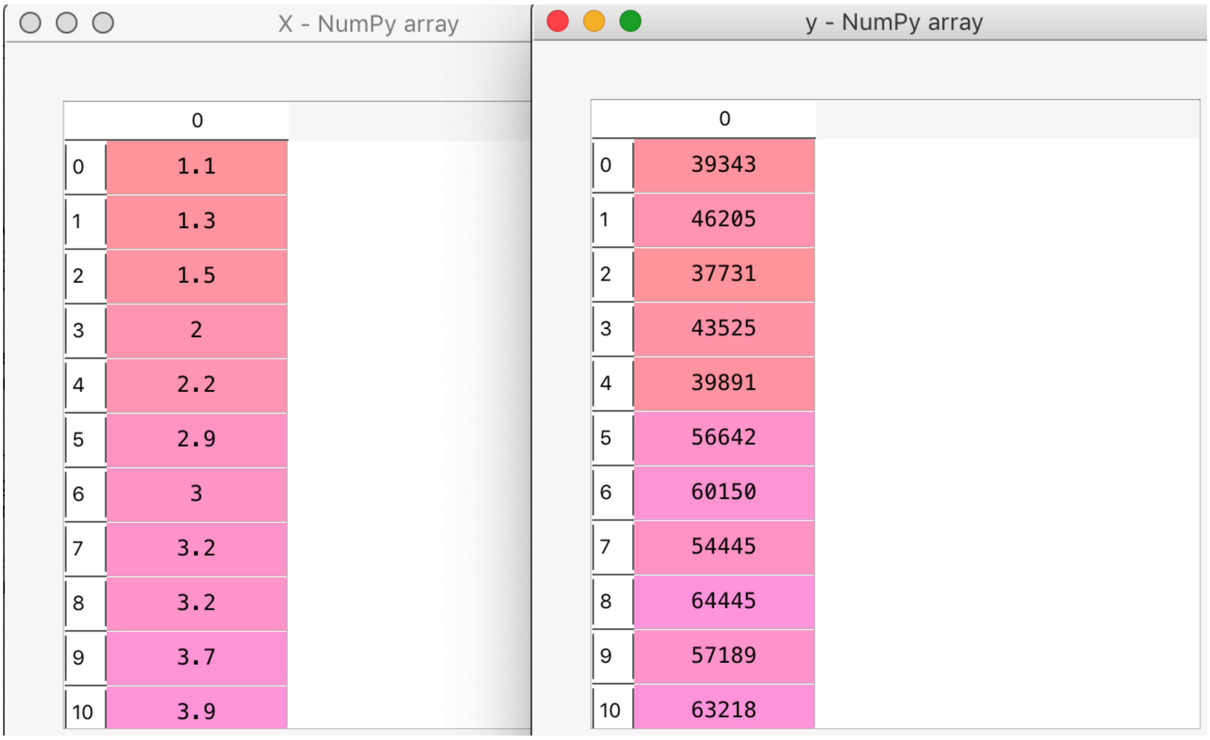
```
dataset.iloc[:, :-1].values
```

which simply means take all rows and all columns except last one

And for y, we specify

```
dataset.iloc[:, 1].values
```

which simply means take all rows and only columns with index 1 — In python indexes begin at 0 — so index 1 here is the second column which is Salary



	0
0	1.1
1	1.3
2	1.5
3	2
4	2.2
5	2.9
6	3
7	3.2
8	3.2
9	3.7
10	3.9

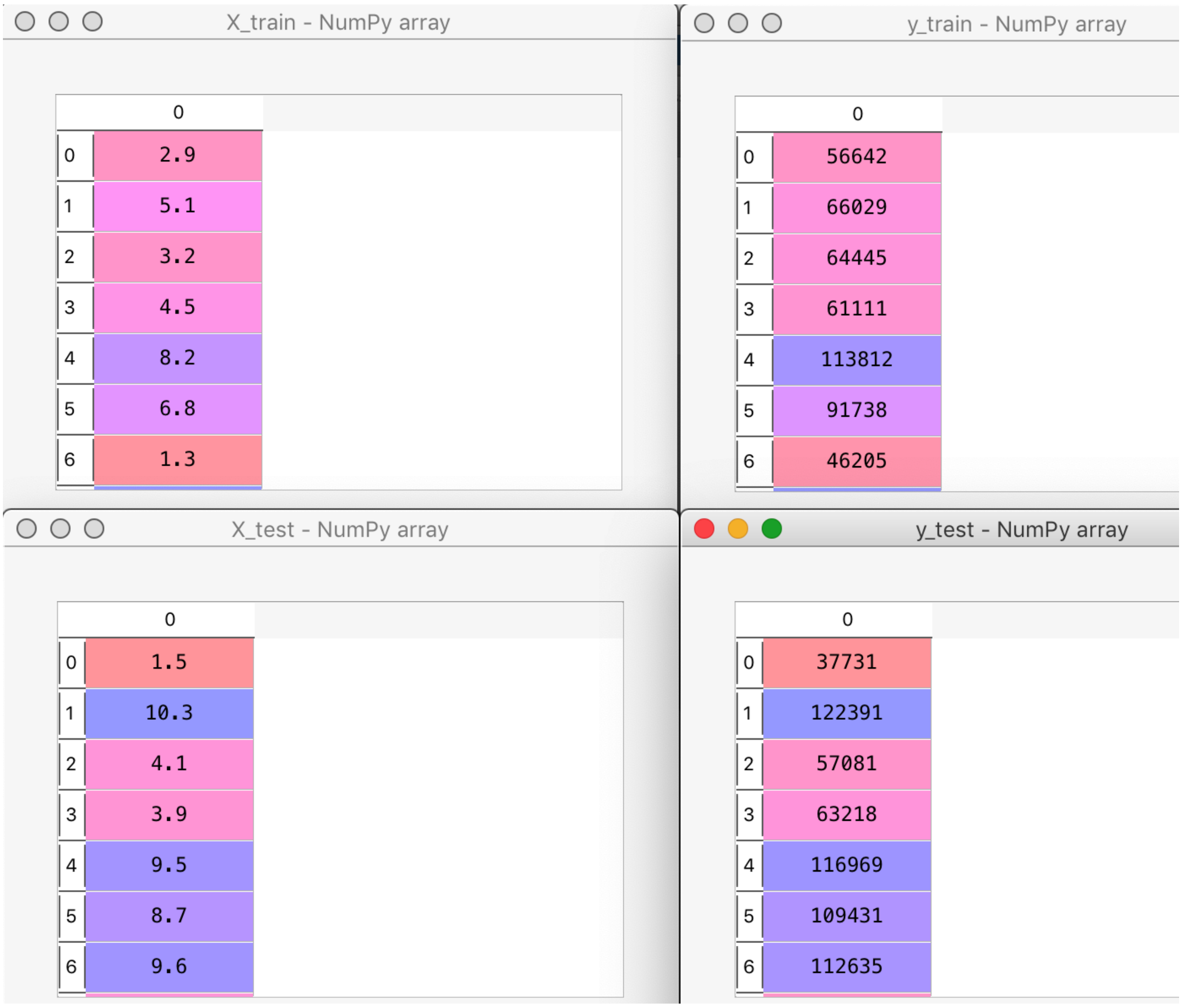
	0
0	39343
1	46205
2	37731
3	43525
4	39891
5	56642
6	60150
7	54445
8	64445
9	57189
10	63218

Step 2: Split dataset into training set and test set

Next we have to split the dataset into training and testing. We will use the training dataset for training the model and then check the performance of the model on the test dataset.

For this we will use the train test split method from library model selection
We are providing a test size of 1/3 which means test set will contain 10 observations and training set will contain 20 observations
The *random state=0* is required only if you want to compare your results with mine.

Below is the sample screenshot of X train, y train, X test and y test



X_train - NumPy array

	0
0	2.9
1	5.1
2	3.2
3	4.5
4	8.2
5	6.8
6	1.3

y_train - NumPy array

	0
0	56642
1	66029
2	64445
3	61111
4	113812
5	91738
6	46205

X_test - NumPy array

	0
0	1.5
1	10.3
2	4.1
3	3.9
4	9.5
5	8.7
6	9.6

y_test - NumPy array

	0
0	37731
1	122391
2	57081
3	63218
4	116969
5	109431
6	112635

Step 3: Fit Simple Linear Regression model to training set

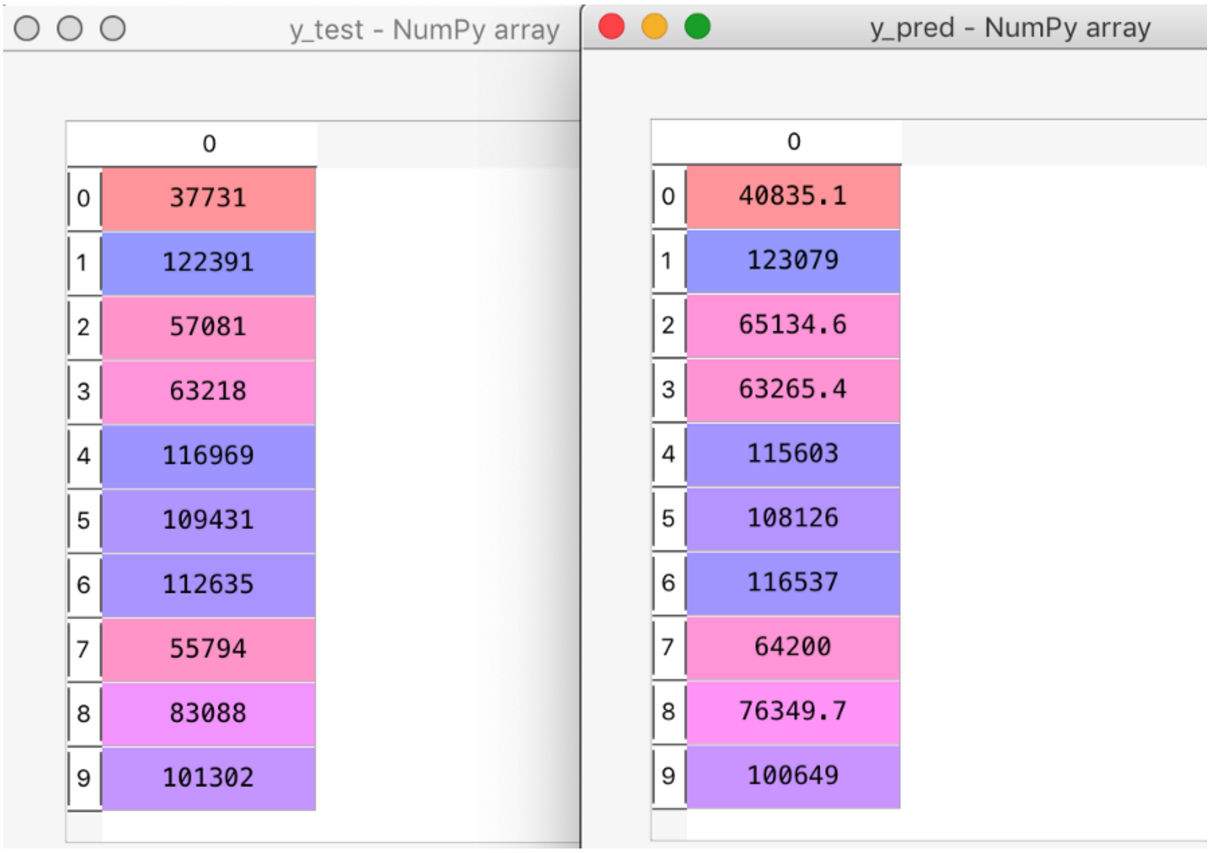
This is a very simple step. We will be using the Linear Regression class from the library sklearn linear model. First we create an object of the Linear Regression class and call the fit method passing the X train and y train.

Step 4: Predict the test set

Using the regressor we trained in the previous step, we will now use it to predict the results of the test set and compare the predicted values with the actual values

Now we have the y pred which are the predicted values from our Model and y test which are the actual values.

Let us compare are see how well our model did. As you can see from the screenshot below — our basic model did pretty well.



y_test - NumPy array

	0
0	37731
1	122391
2	57081
3	63218
4	116969
5	109431
6	112635
7	55794
8	83088
9	101302

y_pred - NumPy array

	0
0	40835.1
1	123079
2	65134.6
3	63265.4
4	115603
5	108126
6	116537
7	64200
8	76349.7
9	100649

If we take the first employee — the actual salary is 37731 and our model predicted 40835.1 — which is not too bad. There are some predictions that are off but some are pretty close.

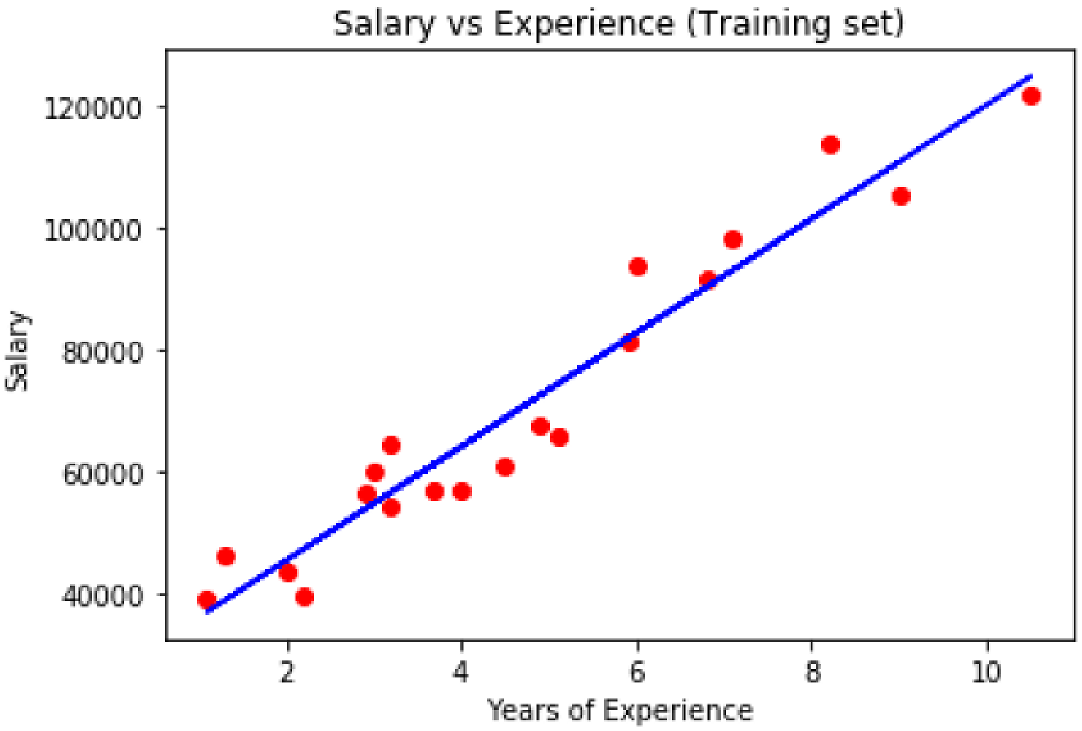
Step 5 — Visualizing the training set

Let’s visualize the results.
First we’ll plot the actual data points of training set — X train and y train

```
plt.scatter(X_train, y_train, color = ‘red’)
```

Next we’ll plot the regression line — which is the predicted values for the X train

```
plt.plot(X_train, regressor.predict(X_train), color=’blue’)
```



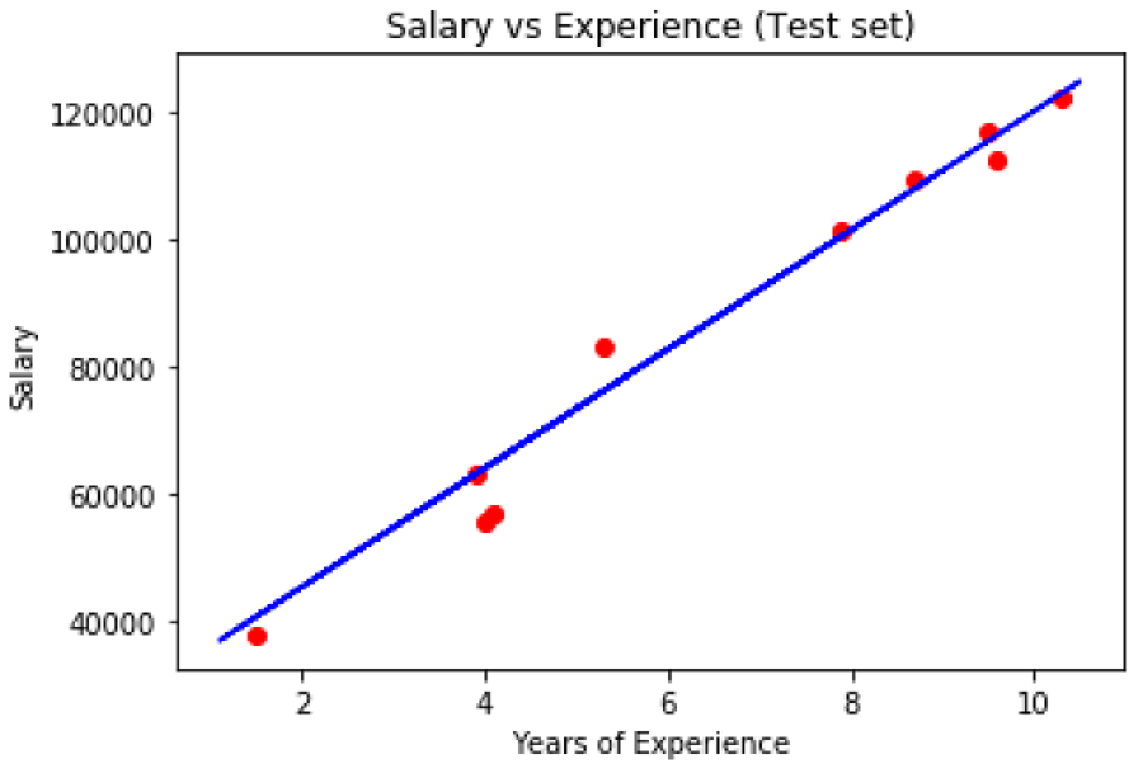
Step 6 — Visualizing the test set

Let’s visualize the test results.
First we’ll plot the actual data points of training set — X test and y test

```
plt.scatter(X_test, y_test, color = ‘red’)
```

Next we’ll plot the regression line — which is the same as above

```
plt.plot(X_train, regressor.predict(X_train), color=’blue’)
```



Step 7 — Make new predictions

We can also make brand new predictions for data points that do not exist in the dataset. Like for a person with 15 years experience

RESULT

The Predicted Salary Of A Person With 15 Years Of Experience is [167005 .32889087]

THANK YOU