

Simple Linear Regression

① Intuition

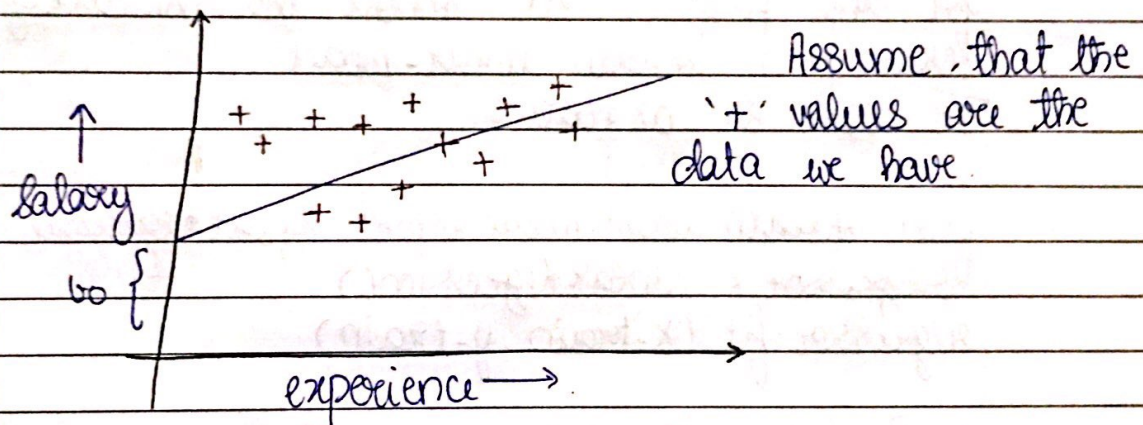
In simple linear regression, we use one independent variable to predict the value of a dependent variable.

Our data for this template contains two columns, salary & experience; and since, here we have to predict salary based on experience:

independent variable \rightarrow experience

dependent variable \rightarrow salary.

Thus, for a simple linear regression,



We try to predict the best possible line of the form

$$y = b_0 + b_1x,$$

where y is dependent variable & x is independent variable.

② Method to find best possible line

Let us assume a point \hat{y}_i on our line corresponding to a value y_i in our database, for same x .

Then, we project lines between certain values such that $\sum (y_i - \hat{y}_i)^2$ is minimum is our solution!

③ Writing it in python

First, copy the data preprocessing template and adjust it according to the data we have. We will not need to use feature scaling as the library we use takes care of that for us. Next.

(i) We fit simple linear regression to the training set

for this purpose, we import the `LinearRegression` class from `sklearn.linear-model`. The code is as follows:

```
from sklearn.linear-model import LinearRegression  
regressor = LinearRegression()  
regressor.fit(X_train, y_train)
```

(ii) Predicting the test set results

To do this, we will create a vector (`y_pred`) which will contain all values of `y-test` as predicted by the machine.

It's simple,
the code:

```
y_pred = regressor.predict(X_test)
```


(iii) Visualising the training set results

We use the matplotlib.pyplot library on the graph.

X axis :- ~~is~~ experience

Y axis :- salary

For this we use plt.scatter() which takes parameters as:

X axis values, y axis values, colour.

This will give us a scatter plot.

Then, we will use plt.plot to plot a line graph.

The code:

```
plt.scatter(X-test, y-test, color='red')
```

```
plt.plot(X-train, regressor.predict(X-train), color='blue')
```

and then the title labelling.