

simple-linear-regression

March 6, 2025

SIMPLE LINEAR REGRESSION

Regression ->what is regression Regression is a statistical approach to find the correlation between variables(dependent and independent).

Regression algorithms give you continuous output.then you must choose one of the regression algorithm to bulid model.

types of regression 1.simple linear regreession 2.multi linear regression 3.polynomial regression 4.decision tree regressor 5.random forest regressor

simple linear regreession

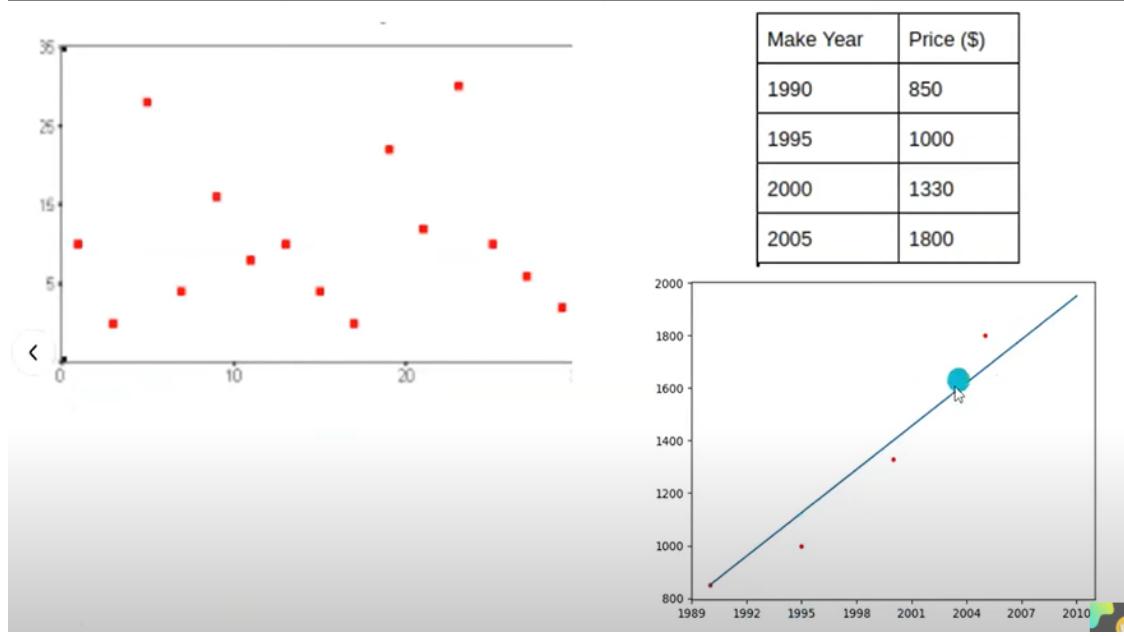
A simplr linear regression model only has two eplanatory variables 1.dependent variable anotherone 2. independent variable.think of two dimensional sapce.where the horizontal axis represents the independent variable-x and vertical axis represents the dependent variable-y

where we can apply SLR. 1.Linearity

it assumes that the relation btween the data points is linear.

```
[12]: from PIL import Image  
Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 161135.  
png")
```

[12]:



2.independence

There must be no relationship among the different values of the independent variable. the value is unique.

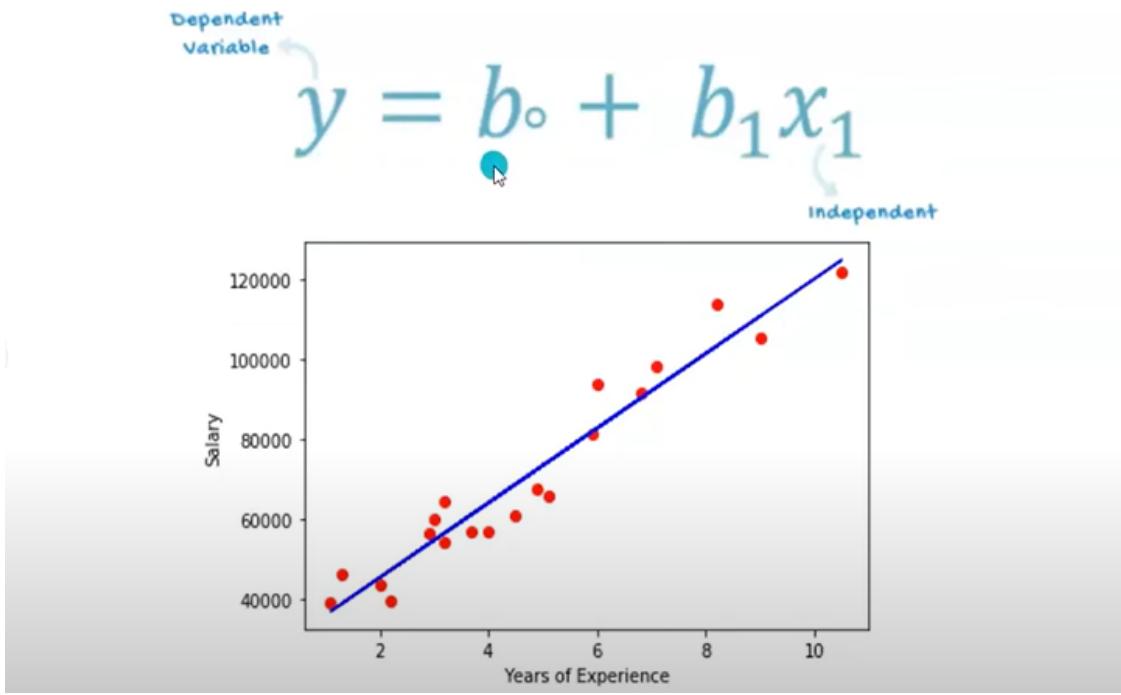
3.Normality

data points follow a normal distribution.if we plot the histogram fo the data,we should be able to draw a skewed line.

simple linear regression equation

```
[24]: Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 162724.  
png")
```

```
[24]:
```



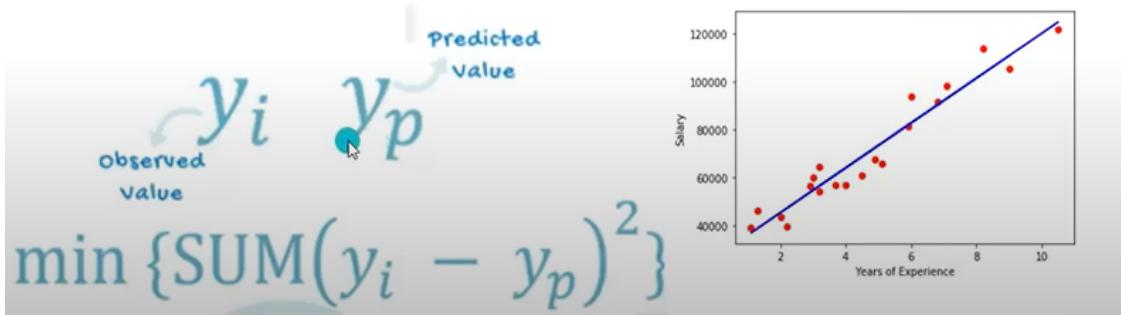
where do you find b_0 and b_1 values?

we are trying to minimize the errors in prediction by finding the best fit line.

we are trying to minimize the length between the observed value(y_i) and predicted value from our model (y_p). #here observed value means actual output value.

[28] : `Image.open("C:/Users/kamis/Pictures/Screenshots/Screenshot 2025-03-01 164012.
png")`

[28] :



[34] : `Image.open("C:/Users/kamis/Pictures/SLR-2.jpg")`

[34] :

Simple Linear Regression

let's code

1. Data preprocessing.

→ clean the data

→ find the missing value

and remove that

→ to modify you can
use strategy (mean, mode, median)

to fill the value

→ Normalize the data set and
categorical data to convert it

numerical you can use
encoding, One Hot encoding, Label encoding, column trans

2. fitting the Model and training data

divide the data train and test
we can train data to train the model

3. Predicting the results

test data use to predict the result

4. visualization.

we can use matplotlib lib.

Graph

[32] : `Image.open("C:/Users/kamis/Pictures/SLR-1.jpg")`

[32] :

Calculate the accuracy

By using autoscore and mean square

- Mean square - This not giving correct accuracy
- autoscore - Better to give correct accuracy value

→ Improve the model

1. Normalize the data

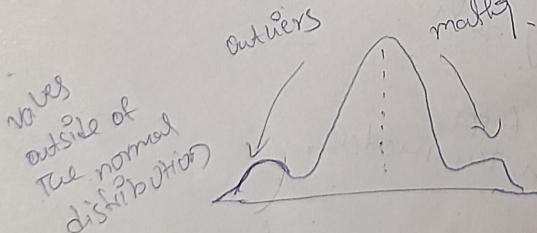
normalize means like feature scaling.

Independent values are same

0 to 1 scale → 1 to 1, Then it

will be worse actually result.

2. Remove outliers



3. Check collinearity

Independent not similar

If any similar independent variable you can remove any one.

That time better accuracy result if comes.

```
[1]: import numpy as np # numpy using dealing with arrays  
import pandas as pd # pandas using read dataset and manipulating data frames  
import matplotlib.pyplot as plt # matplotlib using visualisation
```

```
[3]: dataset=pd.read_csv("C:/Users/kamis/OneDrive/Desktop/ml-notes/Salary_Data.csv")
```

```
[5]: dataset.head() #showing the top the dataset
```

```
[5]:    YearsExperience    Salary
0            1.1    39343
1            1.3    46205
2            1.5    37731
3            2.0    43525
4            2.0    43525
```

```
[7]: dataset.isna().sum() #any missing value you can use this
```

```
[7]: YearsExperience      0
Salary          0
dtype: int64
```

```
[9]: x=dataset.iloc[:,1].values
y=dataset.iloc[:,0].values
```

```
[11]: from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

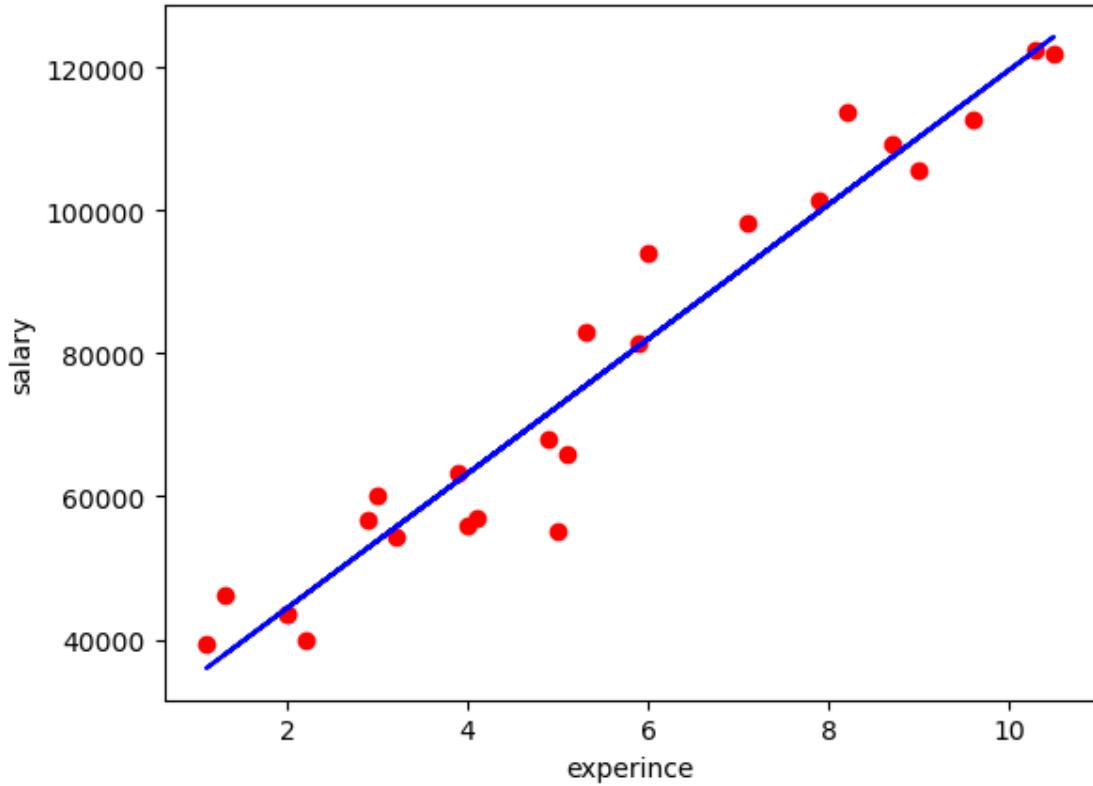
```
[13]: from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(x_train,y_train)
```

```
[13]: LinearRegression()
```

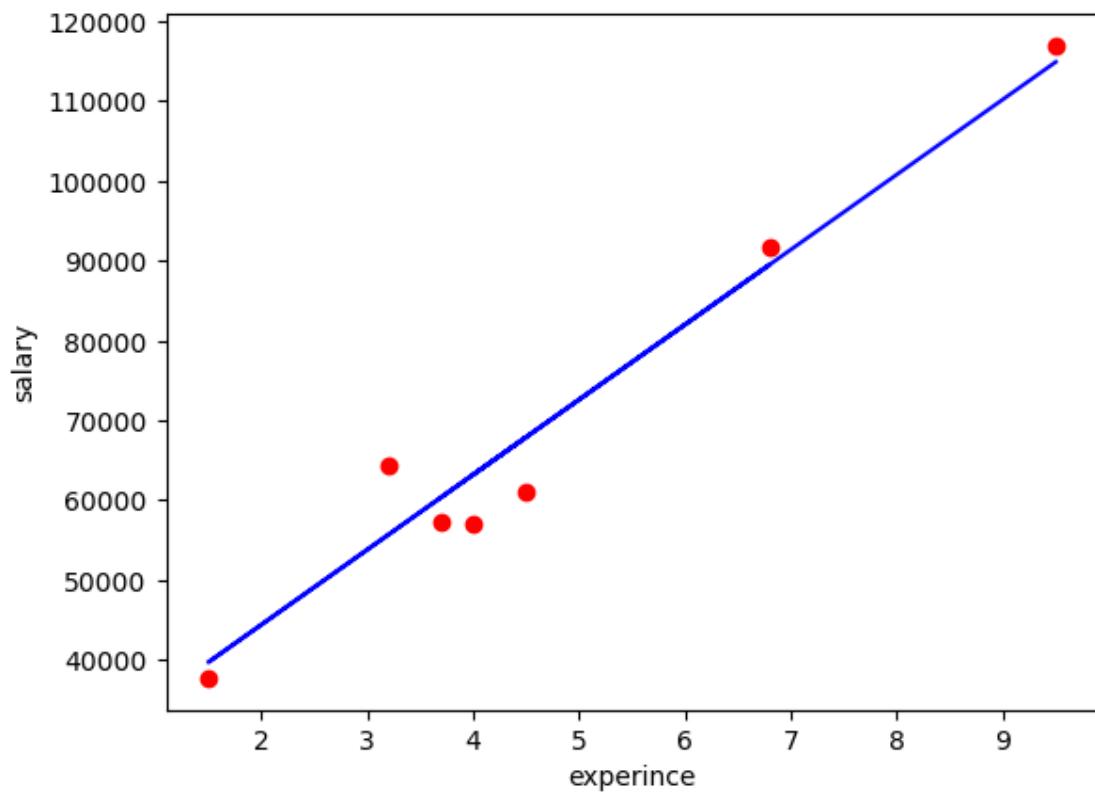
```
[15]: y_pred=regressor.predict(x_test)
```

```
[21]: import warnings
warnings.filterwarnings('ignore')
```

```
[31]: plt.scatter(x_train,y_train, color='red')
plt.plot(x_train,regressor.predict(x_train) ,color='blue')
plt.xlabel('experince')
plt.ylabel('salary')
plt.show()
```



```
[33]: plt.scatter(x_test,y_test,color='red')
plt.plot(x_test,regressor.predict(x_test),color='blue')
plt.xlabel('experince')
plt.ylabel('salary')
plt.show()
```



[]:

[]:

[]:

[]: