#Modeling in machine learning

(A) simple linear regression

say a line → a simple linear equation

$y = b_0 + b_1 x_1$

- b_0 → y intercept
- y → dependent variable
- x_1 → Independent variable

In Regression on basis of Independent variable and some other info we will try to predict the dependent variable

so say y = amount of potatos
x = amount of Fertilizer so

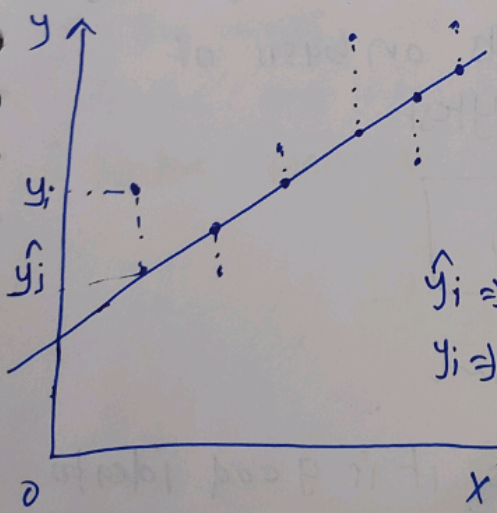
potatoes [In ton] = b_0 + b_1 (slope) × Fertilizer in kg

So If we have value of b_0 & b_1 we can able to plot it in graph which can help us to predict value of dependent variable using a linear equation

⇒ And we can have multiple different slope lines which contains some dataset so in order to find out correct one we can use ordinary ~~square~~ least square

(*) Ordinary least square

It is a method used to find most optional linear regression line for a dataset.



In least square for each line we find one which has least ordinary least square value from all point and that line will be optimal one to predict dependent value.

$\hat{y}_i$ ⇒ Predicted
y_i ⇒ actual

⇒ we have goal to minimize difference betn predicted and actual value

i.e $\text{Sum}(y_i - \hat{y}_i)^2$ is minimized

for $\hat{y} = b_0 + b_1 x$.

difference

residual $\epsilon_i = y_i - \hat{y}_i$

do preprocessing of dataset first & then go for modelling

* (A) Training dataset and model

You can build a model from scratch or other you can do is to ~~create~~ use prebuild models.

from scikit we are going to use linear Regression class and then we can create object for it

class

Code:

```
from sklearn.linearmodel import LinearRegression
regressor = linearRegression()
```

↳ Here you don't have to insert any parameters

this will create simple Regration model

the fit function is one used to train dataset

Fit method always used to train your model

```
regressor.fit( xtrain, ytrain);
```

we have to insert data in some format

↳ this will train "regressor" model on xtrain & ytrain that is training dataset

now predicting test set result and evaluating results

use predict function to predict results on basis of xtest and check outputs with ytest

```
regressor.predict(xtest)
```

and output will be predicted value

↳ this will return a array so it is good idea to store it to visualise

```
result : regressor.predict(xtest)
```

result is predicted (y)

ypred

so now we have to plot ytest and y_Predicted Results

#Visualise train set

plt.scatter(~~X~~ Xtrain, ytrain, color = "red")

& plot Regression line builded model button)

plt.plot(Xtrain, ~~yPred~~);

plt.plot(Xtrain, regressor.Predict(X_train), color = "blue")

In order to label gran

you can use title(), xlabel(), ylabel() &

at last

plt.show() to show output

learn plt function

multiple regression model has multiple feature

Simple linear model

has only one feature

in matrix of feature

so it is so bad for multiple feature dataset like heart disease prediction

⇒ The simple linear Regression is good only for the single feature and it is not good idea to use it for multiple feature containing dataset.