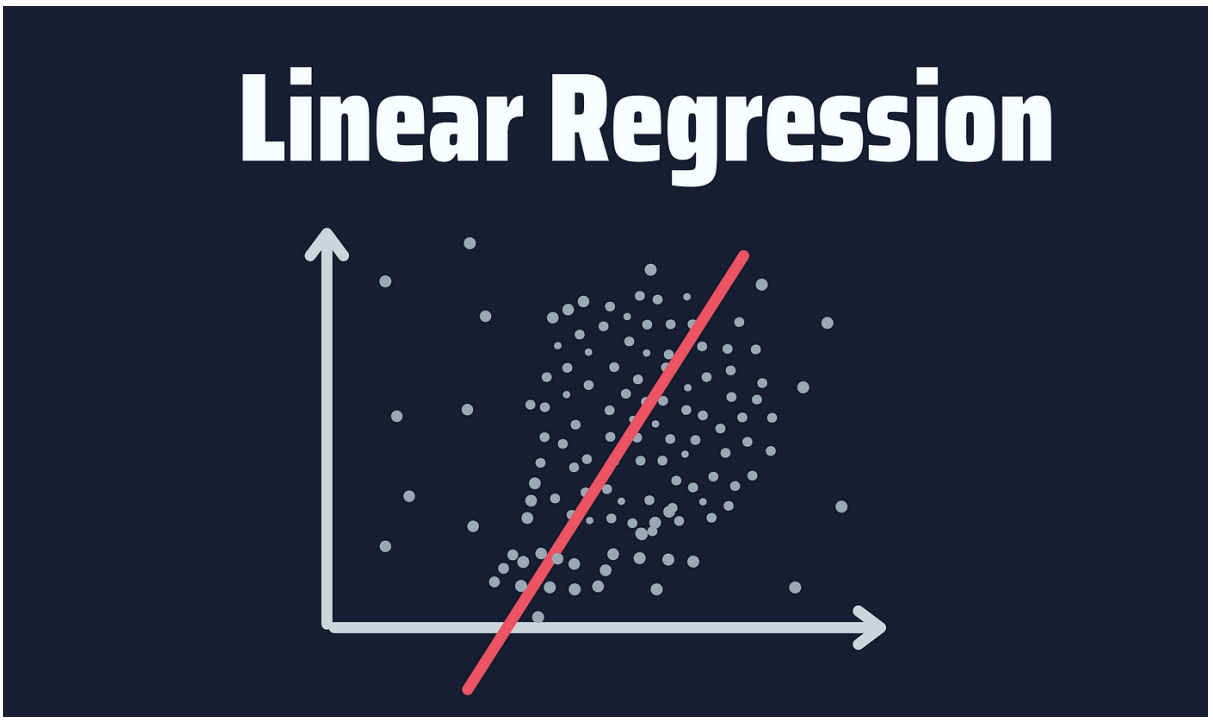


Table of Contents

Understanding the algorithms

① Simple Linear Regression

Linear model in Regression (supervised learning)

Simple linear regression assumes that a linear relationship exists b/w the response variable & the explanatory variable, It models this relationship with a linear surface called a "hyperplane"

→ This will be clear with an eg.
Let's take the height & weight sample of few people

Height	weight
5.6	60
5.7	62
5.8	63
5.9	62
6.1	64
6.2	75

$H \propto W$

$W \propto H$

Now, I want to know what would be the weight of a person with 6.0 Height?

→ By observation it seems like

$H \propto W$ → to remove proportionality we can multiply a constant

$W = mH$ → But, it may also differ by some value, so, $\pm b$

$W = mH \pm b$ → Now, this is the relation of height & weight.

constants

$w = mH + b$ → Now, here if I know the value of 'm' & 'b'. then I can predict the weight.

given height

Now, main aim is to find out value of 'm' & 'b'.

→ Let's take 2 sample, (5.6, 60), (5.7, 62),

$60 = m(5.6) + b$ — (1)

$62 = m(5.7) + b$ — (2)

→ By this we get $m = 20$, $b = -52$.

So, our equation $w = 20H - 52$

Let's try with a known height 5.9.

$w = 20(5.9) - 52 = 66$ kg.

Here we got '66', but we are expecting 62.

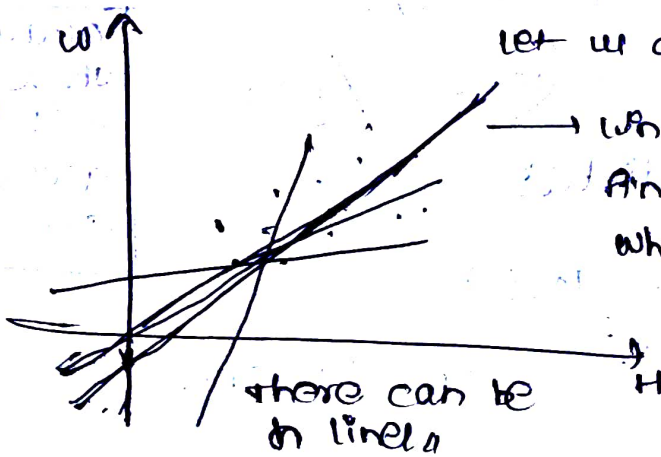
(*) So, may be if we change ['m' & 'b'] then we could get a exact value (approx.)

Now, Here we get the definition of ML as **establishing a mathematical relationship** of data. So, that it can predict a new data.

→ So, when we say a ML model,

(*) It is nothing but a mathematical relationship. which can predict new data.

Plotting 'H' & 'w'



let us assume we have a best fit line.

→ Linear regression always tried to find the linear relation b/w variable which is a straight line.

Hence, $y = mx + c$

→ we could find n diff lines.

But that one line which can predict all the values approx. is the "Best fit line"

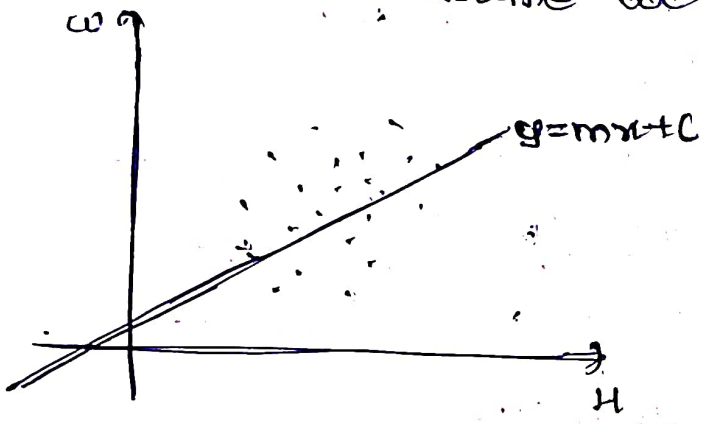
$y = mx + c$ → And this best fit line is called model

→ we have to find the 'm' & 'c'.

m-slope, c-Intercept

(*) Evaluating the fitness of the model with a cost function

→ First, let us assume we have a best fit line.



But, now we have doubts.

How did we opt this line?
why this line?
what about other lines?

→ Now, if it's the best fit line, then for future if I want the weight of '6.0'.

→ It is nothing but the value of y-coordinate $y = mx + c$ line with x coordinate '6.0'

→ If this line is $W = 20H - 52$ and at $x = 5.9$,

I got $\hat{w} = 66$, but actual $w = 62$.

So, I can see there is a loss / Error.

$$\text{loss} = |(y - \hat{y})|$$

(y known, $\hat{y}$ predicted)

Always there will be a slight error

Q.) why did we got this loss?

A may be best of 'm' & 'c' we taken.

Gradient Descent - 48

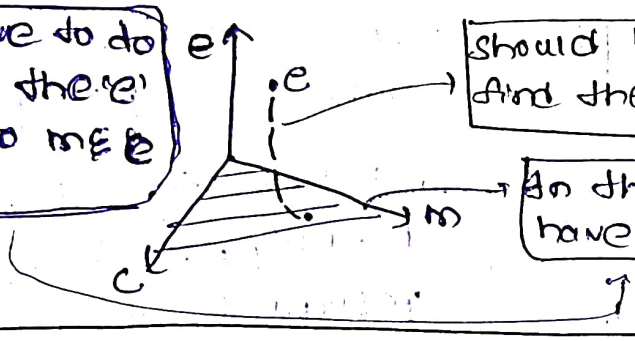
→ Now, to improvize my model strength I have to 'reduce the loss'.

and this loss depends on 'm' & 'c' taken.

→ So, I am supposed to shift m & c such that I end up reducing the loss.

we have loss (e), slope (m), constant (c).

So, I have to do is bring the 'e' down to m & e plane



should bring down & find the new coordinate

In this plane we have 'error'.

Slowly, by examining each input m & c gets changed and eventually loss decreased. this process is called learning.

So,

$$m_{new} = m_{old} - \eta \Delta E_m$$

$$c_{new} = c_{old} - \eta \Delta E_c$$

m_{new} - changed one, new

m_{old} - old m value.

η - Learning rate

ΔE_m - Error in m $\left(\frac{de}{dm}\right)$

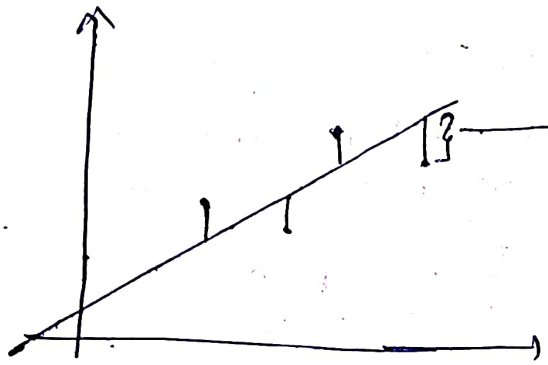
$\eta \rightarrow (0.0001 - 10)$ range.

→ So, here we are taking a new m.

By subtracting a fraction of the error rate from old m that fraction is learning rate.

So, now I am supposed to choose the m & c, which gives a least possible error then it will be the best fit line.

→ A cost function also called a loss func, is used to define & measure the error of a model. the diff. b/w the weights predicted by model & the observed weights in training set are called "residuals", or training error / loss.



→ this is the residual / loss. which is $|y-\hat{y}|$

→ we have to reduce this residual actually, we have to reduce residual of all data points.

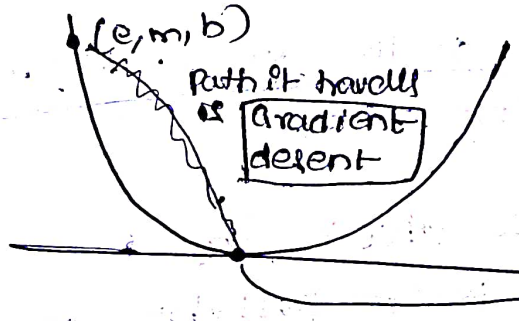
→ we can produce best weight predictor model by minimizing the 'sum of residuals'.

(Sum of residuals) $\sum_{i=1}^{n} r_i = \sum_{i=1}^{n} [y_i - (mx_i + c)]$ → $\hat{y}$

So, to cancel the negative value, we take square

(*) $R(x) = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} [y_i - (mx_i + c)]^2$ — sum of squares of residuals.

Now, it is a quadrant in 3D.



→ So, we are supposed to shift the (e, m, b) to the origin (0, m, b).

where $\frac{de}{dm} = 0$, $\frac{de}{db} = 0$.

As we can see residual is both a func. of m & b, so, differentiating partially w.r.t m & b will give us

$$\frac{\partial E}{\partial m} = \sum_{i=0}^{n} x_i (y_i - (mx_i + b))$$

$$\frac{\partial E}{\partial b} = \sum_{i=0}^{n} 2 (y_i - (mx_i + b))$$

so, we know for best line, residual should be min. minima of func. occurs where derivative = 0. i.e,

$$\sum_{i=0}^{n} x_i (y_i - (mx_i + c)) = 0$$

$$\sum_{i=0}^{n} 2 (y_i - (mx_i + b)) = 0$$

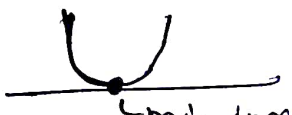
$$\sum_{i=0}^{n} x_i y_i - \sum_{i=0}^{n} m x_i^2 - \sum_{i=0}^{n} c x_i = 0$$

$$\sum_{i=0}^{n} y_i - \sum_{i=0}^{n} m x_i - \sum_{i=0}^{n} c = 0$$

the same eqn. can be written in matrix form as:

$$\begin{bmatrix} \sum_{i=0}^{n} x_i^2 & \sum_{i=0}^{n} x_i \\ \sum_{i=0}^{n} x_i & n \end{bmatrix} \begin{bmatrix} m \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=0}^{n} x_i y_i \\ \sum_{i=0}^{n} y_i \end{bmatrix}$$

ideally, if we'd have an eqn. of one dependent & one independent variable the minima would look like



dependent var $y = mx + c$ independent var

→ the new values for slope & intercept are calculated.

repeat until convergence()

{

$$m_{new} = m_{old} - \eta \left[\frac{1}{n} \sum_{i=1}^{n} (h_\theta \cdot x_i - y_i) x_i \right] \quad \frac{dE}{dm}$$

$h_\theta = mx$

$$c_{new} = c_{old} - \eta \left[\frac{1}{n} \sum_{i=1}^{n} (h_\theta \cdot x_i - y_i) \right] \quad \frac{dE}{dc}$$

}

Accuracy → this is how LR works.

(*) measuring model strength

RSS - Residual Sum of Squares
TSS - Total sum of Squares

The R^2-squared (R^2) statistic provides a measure of fit.

It takes the (form of proportion) - (proportion explained)

$$R^2 = 1 - \frac{RSS}{TSS}$$

$$RSS = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$ → Predicted

because changing

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$ → mean.

If $R^2 = 0.75$ it says that our model fits 75% of dataset.

Adjusted R^2 Statistic

→ As R^2 is just a linear eqn., as we increase the no. of independent variables in our eqn., the R^2 increases as well.

But, that doesn't mean that the new independent variables have any correlation with the output variable.

i.e; R^2 will increase, but is not necessary model yields better results.

→ To rectify this we use adjusted R^2 value which penalises excessive use of such features which don't correlate with the output data.

$$R^2_{adj} = 1 - \frac{(1-R^2)(N-1)}{N-P-1}$$

where,
P - no. of predictors
N - total sample size.

If $P=0$, $R^2_{adj} = R^2$.

Note:-

→ using training data to learn the values of the parameters for simple linear regression that produce the best fitting model is called ordinary least squares (OLS) (or) linear least square.

Another approach to find m & C.

(To solve OLS)

→ variance is a measure of how far a set of values are spread out.

→ covariance is a measure of how much two variables change together. If the variables increase together, their cov is positive. 0 - no relation, -ve → one increase & one decrease

$$var = \frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}$$

→ variance of explanatory variable.

$$cov = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{n-1}$$

→ covariance of the response & explanatory variables.

$$\therefore \beta = \frac{cov(x,y)}{var(x)}$$

$y = \beta x + \alpha$.

$\alpha = \bar{y} - \beta \bar{x}$ → we can get α

Notes

→ The independent variables are uncorrelated with the residual term, also known as "Exogeneity". This in layman term generalises that in no way should the error term be predicted given the value of independent variables.

the error terms have a constant variance.
i.e) Homoscedasticity.

→ the error terms are normaly distributed.

→ No multicollinearity, i.e; no independent variables should be correlated with each other or affect another.

model confidence

Q) Is linear regression a low bias / high variance model (or) a high bias / low variance model?

A) It's a "High bias / low variance" model

→ Even after repeated sampling, the best fit line will stay roughly in the same pos. (low variance)

→ But the avg. of the models created after repeated sampling won't do a great job in capturing the perfect relationship. (High bias)

→ Low variance is helpful when we don't have real training data.

```
import statsmodels.formula.api as smf
model_name = smf.ols(formula='y ~ x', data=df).fit()
model_name.conf_int()
```