05-02-2021

KANAV BANSAL

LECTURE - 44

As we know that, To solve the OPTIMISATION EQUATION of Linear Regression we use GRADIENT DESCENT.

* The values of learning Rate i.e., η ranges between 0 and 1.

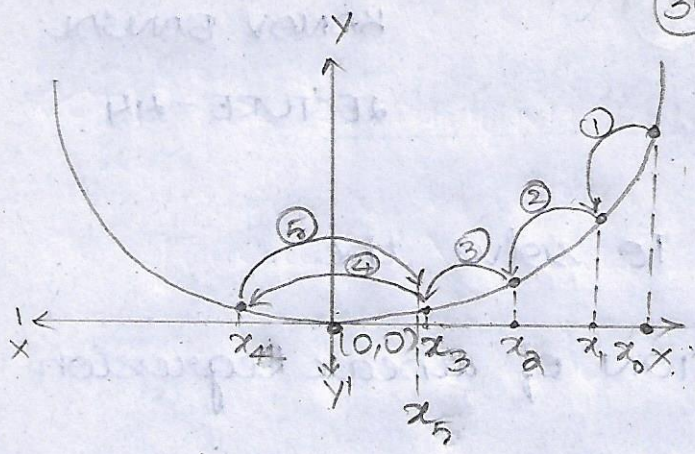
* We takes η values as 1, 0.1, 0.01, 0.001, 0.0001 frequently. in order to choose which η value fits as the best for the model.

↳ If η - small → more number of parameters are computed.

↳ If η - largere → less number of parameters are computed.

NOTE:

η value by default is taken as 0.001.



$\Rightarrow x_1 = x_0 - \eta\left[\frac{df}{dx}\right]$

$\Rightarrow x_1 = x_0 - \eta$ [-ve value]

$\Rightarrow x_1 = x_0 + \eta$ [value]

↳ (A)

↳ From the above graph, we know that the values of 'x' are changing i.e.,

x_0 to x_1 ; x_1 to x_2 ..... ; x_4 to x_5

* ↳ From x_3 to x_4, we know that the "sign" of the value remains same i.e.,

$\Rightarrow x_4 = x_3 - \eta\left[\frac{df}{dx}\right]$

* ↳ But from x_4 to x_5 we observe that the "sign" of the value changes from "-ve" to "+ve"

$\Rightarrow x_5 = x_4 - \eta\left[\frac{-df}{dx}\right]$

$\Rightarrow x_5 = x_4 + \eta\left[\frac{df}{dx}\right]$ → (B)

This is because, the x_4 values needs to go at minima point (0,0).

→ Due to the long move, it moved towards the x' (-ve x-axis) i.e., left side of minima.

→ So, now the x_4 needs to go back and first move to minima and then to the left side of minima.

→ In this process the value of x_4 is "-ve" as it tends towards the right side of minima i.e., x_5 is obtained and the value of x_5 is "+ve".

$$\Rightarrow (B) \rightarrow x_5 = x_4 + \eta \left[\frac{df}{dx}\right].$$

CODE FOR GRADIENT DESCENT:

```
def gradient(X, y, learning_rate=0.001,
             iters=1000, m_curr=0, c_curr=0):
    N = float(len(y))
    grad = pd.DataFrame(columns=['slope',
                                 'intercept', 'mse'])
    for i in range(iters):
        y_pred = (m_curr*X) + c_curr
        mse = sum([error**2 for error in
                   (y-y_pred)])/N
# derivative w.r.t 'm'
        derivative_m = -(2/N)*sum(X*(y-y_pred)
# derivative w.r.t 'c'
        derivative_c = -(2/N)*sum(y-y_pred)
# Updating 'm'
        m_curr = m_curr-(learning_rate*
                         derivative_m)
# Updating 'c'
        c_curr = c_curr-(learning_rate*
                         derivative_c)
```

```
grad.loc[i] = [m_curr, c_curr, mse]
return(grad)
```

From the above code,

N → length of 'y' parameter.

mse → Mean squared error. $= \frac{\Sigma (Y_i - Y_{pred})^2}{N}$

m_curr → current value of 'm'

c_curr → current value of 'c'

iters → iterations.

WHY DO WE DIVIDE WITH N ?

→ The reason why we divide with N is we are trying to minimise $\Sigma (Y_{act} - Y_{pred})$ which should be equal to min. the average i.e., $\min. \frac{1}{N} \Sigma (Y_{act} - Y_{pred})^2$

$\Rightarrow \Sigma (Y_{act} - Y_{pred}) \rightarrow \min. \frac{1}{N} \Sigma (Y_{act} - Y_{pred})^2$

For Example,

→ we have values as 10, 100, 20, 30 then

SUM = 10+100+20+30 = 160

AVERAGE = $\frac{\text{SUM}}{\text{NO. OF OBSERVATION}}$

$\Rightarrow \frac{160}{4} = 40$

Wardy

→ we have values as 50, 100, 20, 30 then

SUM = 50+100+20+30 = 200

AVERAGE = $\frac{200}{4}$ = 50

We observe that,

→ As sum increases, the average increases.

→ As reducing the sum, the average also tends to reduce.

$\Rightarrow$ SUM $\propto$ AVERAGE

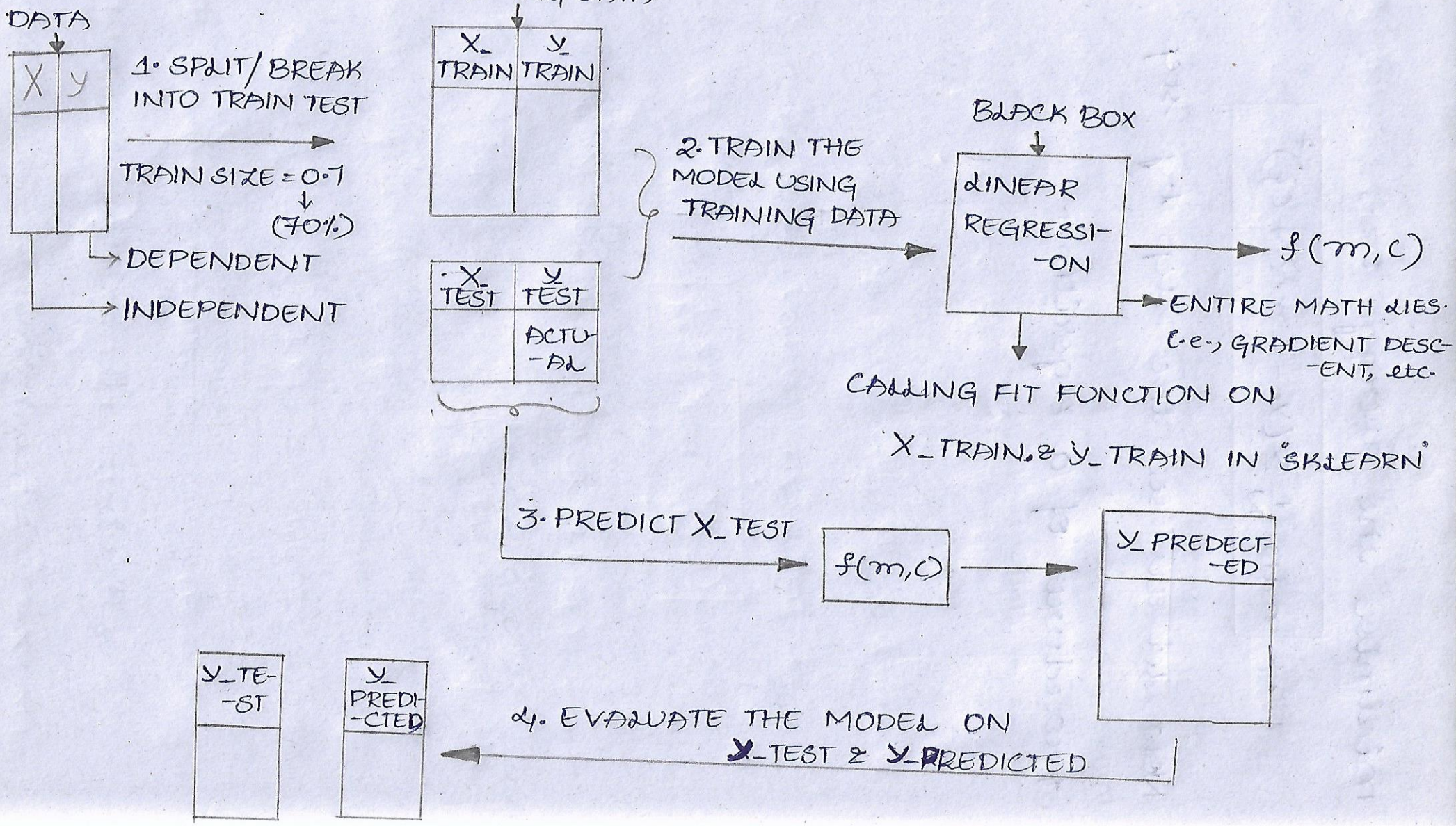
∴ We are taking the derivative - instead of minimizing sum, we

minimise the average i.e.,

$$\Rightarrow \boxed{\min \cdot \frac{1}{N} \Sigma (y_i - \{mx_i + c\})^2}$$

Now let us see the step by step procedure of a Algorithm.

STEPS :


DATA
X
y
1. SPLIT/ BREAK INTO TRAIN TEST
TRAIN SIZE = 0.7
(70%)
DEPENDENT
INDEPENDENT
TRAINING DATA
X_ TRAIN
y_ TRAIN
X_ TEST
y_ TEST
ACTU-AL
2. TRAIN THE MODEL USING TRAINING DATA
BLACK BOX
LINEAR REGRESSI-ON
f(m, c)
ENTIRE MATH LIES
i.e., GRADIENT DESC-ENT, etc.
CALLING FIT FONCTION ON
X_TRAIN & y_TRAIN IN "SKLEARN"
3. PREDICT X_TEST
f(m,c)
y_ PREDECT-ED
y_TE-ST
y_ PREDI-CTED
4. EVALUATE THE MODEL ON
y_TEST & y_PREDICTED

↳ As each value in X changes, we will get the values of y_{pred} in "array" form.

↳ Using y_{test} & y_{pred}, we evaluate the model.

* ↳ The above steps are same for all the algorithms.

* TO evaluate the models, we use

→ MSE
→ RMSE
→ MAE
→ R^2 SCORE

These methods show how the evaluation of Linear Regression performs.

MEAN SQUARED ERROR (MSE):

The mean squared error of an estimator (of a procedure for estimating an unobserved quantity) measures the average of the squares of the errors i.e., the average squared difference between the estimated values and what is estimated. $\Rightarrow \boxed{\frac{1}{N}\Sigma(y_{test} - y_{pred})^2}$

ROOT MEAN SQUARED ERROR (RMSE):

It is defined as the square root of the mean squared error.

→ Also called as RMS (or) QUADRATIC MEAN.

→ It is the measure of how well a regression line fits the data points.

→ It can also be constructed as 'STANDARD

DEVIATION IN THE RESIDUALS":

$$\Rightarrow \sqrt{\frac{1}{N}\Sigma(y_{test} - y_{pred})^2}$$

MEAN ABSOLUTE ERROR (MAE):

It is defined as the arithematic average of the absolute errors i.e., the actual value and the predicted value.

→ This model evaluation metric is used with regression models.

$$\Rightarrow \frac{1}{N}\Sigma|y_{test} - y_{pred}|$$

R^2 SCORE :

the proportion of the variance in the dependent variable that is predictable from the independent variable.

(or)

Defined as the total variance explained by the model to its total variance.

↳ So, if it is 100%, the two variables are perfectly correlated i.e., with no variance.

→ R^2 measures the goodness of fit of a regression model.

→ The ideal value for R^2 is 1.

→ The closer the value of R^2 to 1, the better is the model fitted.

→ If R^2 is zero, it is a normal mean.

NOTE:

In SVM, the kind of evaluation changes.

SVM → CLASSIFICATION - CHANGES
SVM → REGRESSION - REMAINS SAME
↓
(MSE, MAE, RMSE, R^2 SCORE)

NORMALIZATION:

It is a technique often applied as a part of data preparation for ML.

→ It is used to change the values of numeric columns in the dataset to use a

common scale, without distorting the differences in the range of values or losing information.

STANDARDIZATION:

It is the process of rescaling one or more attributes so that they have a mean value of "zero" and standard deviation of "1".

→ It assumes that your data has a Gaussian (BELL CURVE) distribution.

RANDOM STATE:

It ensures that, the splits that we generate are reproducible.

→ The random state that we provide is used as a seed to the random number

generator.

→ This ensures that the random numbers are generated in the same order.

WHY DO WE USUALLY GO FOR STANDARDIZATION?

→ We make sure that the data is internally consistent. i.e., each data type has the same content and format.

→ These values are useful for tracking the data that isn't easy to compare otherwise.