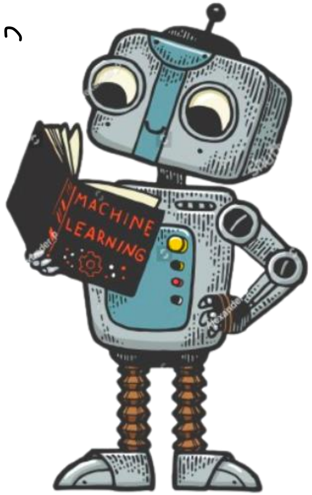
ML: Linear Regression-2

Agenda → Implement Linear-Regression from scratch



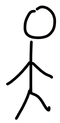
Summary

CARS 24

Linear Regression

Data → Dependent var. then 'Supervised ML Algo'

target / Dependent var. → continuous → Regression algo



features: age, # of kms driven, model, Insurance rating → Computer prog → Expected price

x_1 age	x_2 # of kms driven	x_3 Model	x_4 I.R	y Price

c_1, c_2, c_3, ...

"Historical Data"

Independent variable

target

→ ML-Algo (Linear Reg) → model $\bar{W}, w_0$ → predictions

$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$$

Task (objective) → predict the price of the car given the features of the used cars

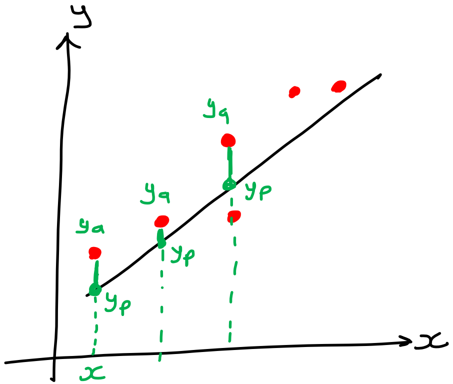
How weights are computed

Linear-Regression

OLS Algo (ordinary Least Squares)

out of all possible values of $\bar{w}$, w_0 the algo will choose those $\bar{w}$, w_0 for which the corresponding RSS is minimum

$$\mathcal{L}(\bar{w}, w_0) = \underset{\bar{w},\, w_0}{\operatorname{argmin}} \sum_{i=1}^{n} \Big(\underbrace{y_i}_{\text{y-actual}} - \underbrace{(\bar{w}^T \bar{x} + w_0)}_{\text{y-predicted}} \Big)^2$$

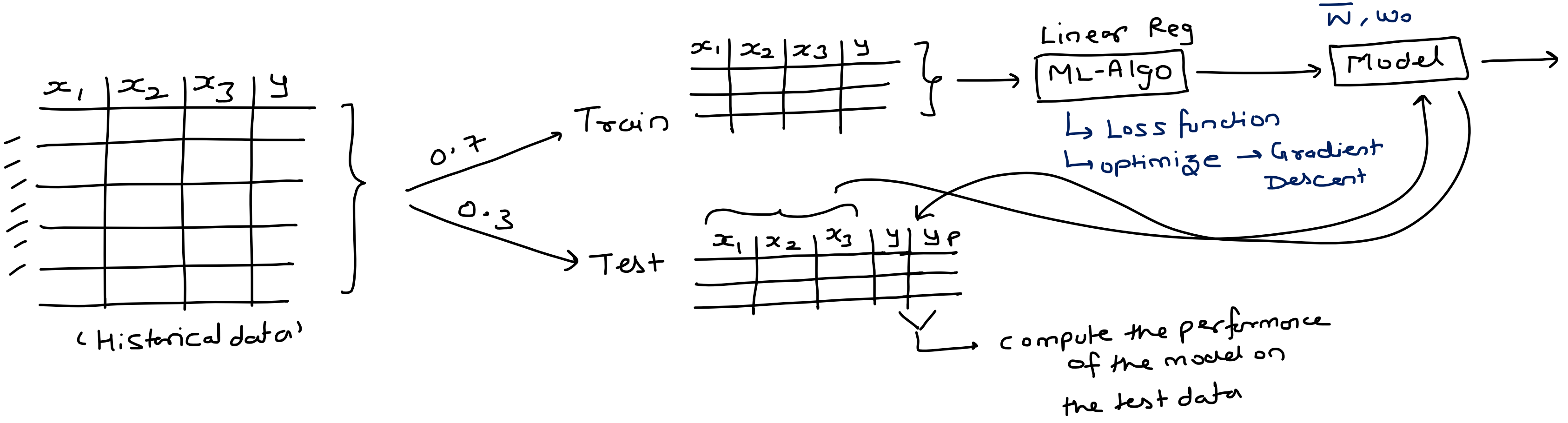


$R^2 \longrightarrow$ How much better your (Regression) model is from the avg-model

$$R^2 = \left[1 - \left(\frac{RSS}{TSS} \right) \right] \qquad TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

$R^2 = 0.85$ is good

pipeline



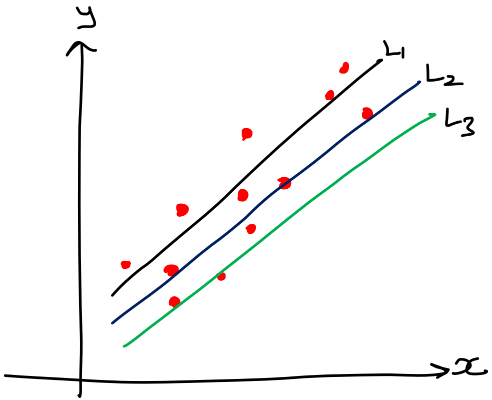
x_1 | x_2 | x_3 | y
(Historical data)
0.7
0.3
Train
Test
x_1 | x_2 | x_3 | y
x_1 | x_2 | x_3 | y | y_p
Linear Reg
ML-Algo
↳ Loss function
↳ optimize → Gradient Descent
$\overline{W}, w_0$
Model
compute the performance of the model on the test data

Loss function } cost function

Linear - Regression

$$\mathcal{L}(\bar{w}, w_0) = \underset{\bar{w},\, w_0}{\text{argmin}} \sum_{i=1}^{n} \left(y_i - (\bar{w}^T \bar{x} + w_0) \right)^2$$

$$\mathcal{L}(\bar{w}, w_0) = \underset{\bar{w},\, w_0}{\text{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (\bar{w}^T \bar{x} + w_0) \right)^2$$



$n = 100$

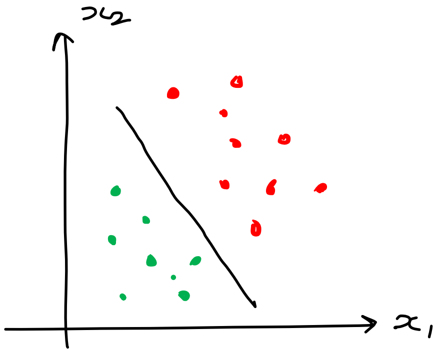
$L_1 : y = w_0 + w x \longrightarrow RSS = 200 \longrightarrow \text{Avg-RSS} = 200/100 = 2$

$L_2 : y = w_0' + w' x \longrightarrow RSS = 100 \longrightarrow \text{Avg-RSS} = 100/100 = 1$

$L_3 : y = w_0'' + w'' x \longrightarrow RSS = 500 \longrightarrow \text{Avg-RSS} = 500/100 = 5$

Advantage $\rightarrow$ High computational improvement

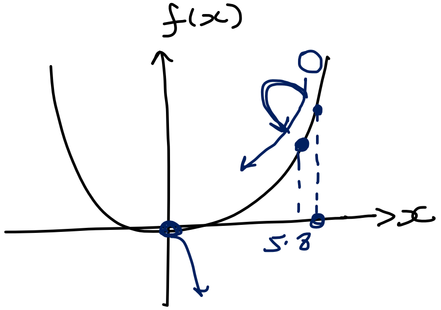
Rough work



$$\mathcal{L}(\bar{w}, w_0) = \underset{\bar{w},\, w_0}{\operatorname{argmin}} - \sum_{i=1}^{n} \underbrace{\left(\frac{\bar{w}^T \bar{x} + w_0}{\|\bar{w}\|}\right)}_{\text{distance}} \cdot y_i$$

$$f(x) = x^2 \longrightarrow f'(x) = 2x$$

optimisation task → find that value of x in which $f(x)$ is minimum



0.0000000

$[x = 6$ (initialisation)

$$x^n = x^o - \underbrace{\eta}_{\text{stepsize}} (\text{slope of } f(x))$$

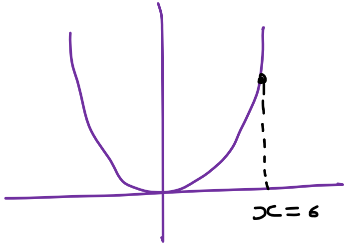
$SS = 0.01$

$$= 6 - (0.01 \times 2 \times 6)$$

$$= 5.8$$

$$\mathcal{L}(\bar{w}, w_0) = \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (\bar{w}^T \bar{x} + w_0)\right)^2$$

$f(x) = x^2$



optimisation problem

minimize loss function to find the value of $\bar{w}, w_0$

Gradient Descent

Update-Rule

$$\begin{cases} \bar{w} = \bar{w} - \eta \left(\nabla_{\bar{w}} \mathcal{L}(\bar{w}, w_0) \right) \\ w_0 = w_0 - \eta \left(\nabla_{w_0} \mathcal{L}(\bar{w}, w_0) \right) \end{cases}$$

$x = x - \eta(\text{slope of } x^2)$

$$\boxed{x = x - \eta(2x)}$$

$$\bar{w} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$w_0 = 1$

find $\nabla_{\bar{w}} \mathcal{L}(\bar{w}, w_0)$

$$\mathcal{L}(\bar{w}, w_0) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$\rightarrow \frac{\partial}{\partial \bar{w}} \frac{1}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$\rightarrow \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \bar{w}} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2(y_i - (\bar{w}^T \bar{x} + w_0)) \frac{\partial}{\partial \bar{w}} (y_i - \bar{w}^T \bar{x} + w_0)$$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0)) \left[\cancel{\frac{\partial}{\partial \bar{w}} y_i}^{\,0} - \frac{\partial}{\partial \bar{w}} \bar{w}^T \bar{x} - \cancel{\frac{\partial}{\partial \bar{w}} w_0}^{\,0} \right]$$

$$\hat{y} = \bar{w}^T \bar{x} + w_0$$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0))(-\bar{x})$$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y})(-\bar{x})$$

$$= \boxed{-\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y})(\bar{x})}$$

find $\nabla_{w_0} \mathcal{L}(\bar{w}, w_0)$

$$\mathcal{L}(\bar{w}, w_0) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$\rightarrow \frac{\partial}{\partial w_0} \frac{1}{n} \sum_{i=1}^{n} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w_0} (y_i - (\bar{w}^T \bar{x} + w_0))^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} 2(y_i - (\bar{w}^T \bar{x} + w_0)) \frac{\partial}{\partial w_0} (y_i - \bar{w}^T \bar{x} - w_0)$$

$\hat{y} = \bar{w}^T \bar{x} + w_0$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}) \left[\cancel{\frac{\partial}{\partial w_0} y_i}^{\,0} - \cancel{\frac{\partial}{\partial w_0} \bar{w}^T \bar{x}}^{\,0} - \frac{\partial}{\partial w_0} w_0 \right]$$

$$= \frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y})(-1)$$

$$\boxed{= \frac{-2}{n} \sum_{i=1}^{n} (y_i - \hat{y})}$$

OLS - Algo

$[x_1 \;\; x_2] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$

$x_1 \times x_1 + x_2 \times x_2$

$$\mathcal{L}(\bar{w}, w_0) = \underset{\bar{w},\, w_0}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \left(y_i - (\bar{w}^T \bar{x} + w_0)\right)^2$$

optimize this loss function [find the value of $\bar{w}, w_0$ where this function is minimum]

Gradient Descent (Sequentially reduce the input value to reach minima of the function)

update - Rule

$(y_i - \hat{y}) \cdot dot(\bar{x})$

update m

$$\begin{cases} \bar{w} = \bar{w} - \eta \left(-\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y})(\bar{x}) \right) \\ w_0 = w_0 - \eta \left(-\frac{2}{n} \sum_{i=1}^{n} (y_i - \hat{y}) \right) \end{cases}$$

$\{ \hat{y} = \bar{w}^T \bar{x} + w_0 \}$

Initialise $\bar{w}, w_0$ $\quad \bar{w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \; w_0 = 0 \} \longrightarrow$ Run for loop for 1000 times to update $\bar{w}, w_0$ $\longrightarrow$ Best $\bar{w}, w_0$

final model

Data pre-processing

1. Convert all categorical data into numbers/numerical

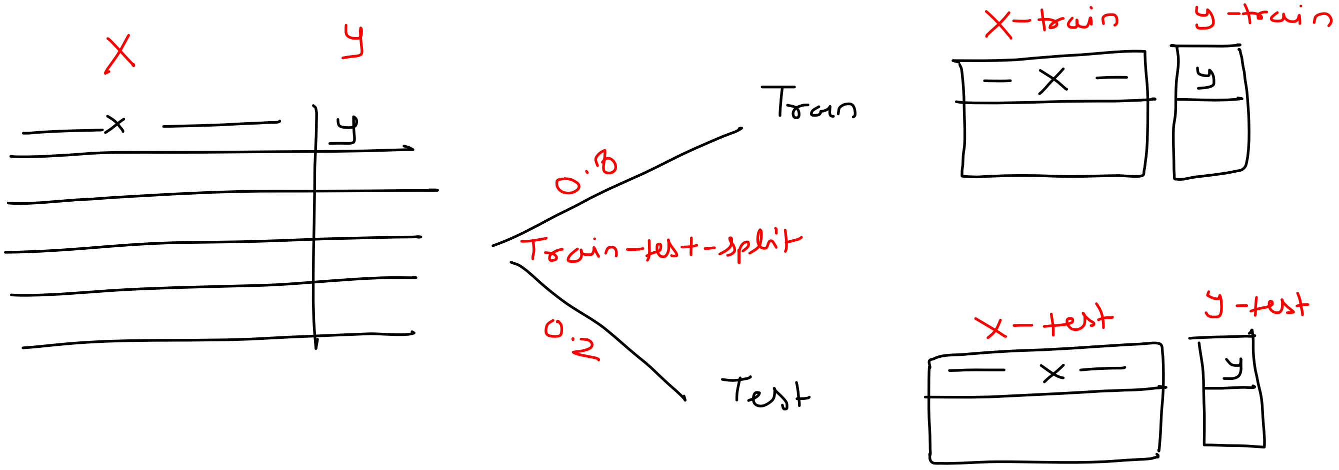
Nominal

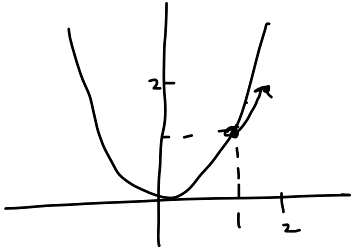
Label encoding (X)

City	
Kanpur	1
chennai	2
Bangalore	3
mumbai	4

→ algo assume that 4 > 3 > 2 > 1 ✗ wrong

No Specific order in the categories





$$f(x,y) = x^2 + y^2$$

$$\frac{\partial f(x,y)}{\partial x} = 2x \quad (\text{slope})$$

$$\frac{\partial f(x,y)}{\partial y} = 2y \quad (\text{slope})$$

$x = 1, y = 1$

$(2, 2)$

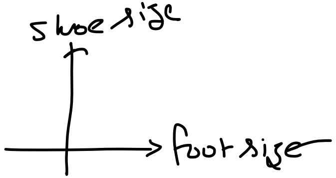
$$\begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Section	y
A	1
B	0
A	1
C	0
A	1
A	0

$$P(1|A) = \frac{3}{4}$$

$$P(1|B) = \frac{0}{1}$$

$$P(1|C) = \frac{0}{1}$$



Indian → model
African → model
American → model

Earth → universal

age	# of km	I	y
		1	
		1	
		1	

→ ML-algo → model w, w_b

Data

1000 → ML-algo → model

test

Higher the data | Lower the bias

age	# of km	I	y
		1	
		0	
		0	

→ ML-algo → model

w, w_b →

Age	No. Of KMS Driven	Model	Price	Sedan	SUV	Hatchback	Model
5	15K	1 Sedan	4.5	1	0	0	3.9
2	11K	2 SUV	6	0	1	0	6.4
1	9K	2 SUV	6.8	0	1	0	6.4
6	21K	3 Hatchback	3	0	0	1	3
2	10K	1 Sedan	3.3	1	0	0	3.9

target

(target encoding)

Model	Average_Price
Sedan	3.9
SUV	6.4
Hatchback	3

One-hot encoding

# of new columns = # of categories

Target encoding

Normalisation

Age	$\left\{ \frac{x - min}{max - min} \right\}$	Age_Scaled
5	$\frac{5-1}{8-1} = 0.57$	0.57
2	$\frac{2-1}{8-1} = 0.14$	0.14
1	$\frac{1-1}{8-1} = 0$	0
3	$\frac{3-1}{8-1} = 0.28$	0.28
8	$\frac{8-1}{8-1} = 1$	1

min 1 max 8

0 1

$$\text{normalisation} = \frac{(x - min)}{(max - min)}$$

fit — min, max

transform =

Normalisation → (0 − 1)

Standardisation → mean = 0, Std dev = 1