Linear Regression-1

Car 24 ← DS

→ Car

2 year

Mil

Km

features

50L

↓

40L

Classification

40L 30L 20L

Regression

→ 40L.40K

→ 40L 40R+20

→ 40L+4R

[0 to 1]

0

0.1

0.001

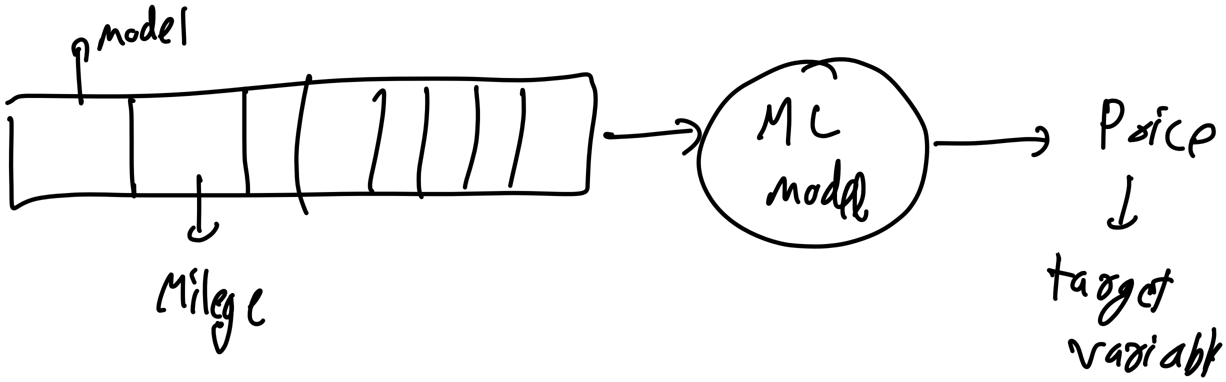
0.01

0.00001

0.000000

0.2

0.0 ......



One Hot Encoding

Age	Milage	Make	Cost
20	10	Maruti	20L
30	40	Ford	30L
—	—	Maruti	—
—	—	Maruti	—
—	—	Ford	—

2 Make + 2 Age + 4 Milege = cost

2 * Maruti

ordinal

Label encoding

School	Age	Milage	Make	Cost
High school	20	10	1	20L
Graduate	30	40	2	30L
Graduate	—	—	1	—
PostGrad	—	—	Maruti	—
	—	—	Ford	—

Maruti → 1

Ford → 2

Age	Milage		Cost	Make_Maruti	Make_Ford
20	10		20L	1	0
30	40		30L	0	1
—	—		—	1	0
—	—		—	1	0
—	—		—	0	1

Age	Milage	Make	Cost	Make_maruti	make_For	BMW	H
20	16	Maruti	20L	1	0	0	0
30	40	Ford	30L	0	1	0	0
—	—	BMW	—	0	0	1	0
—	—	Maruti	—	1	0	0	0
—	—	Ford	—	0	1	0	0
—	—	Hundai	—	0	0	0	1

Target Encoding

0 to 1

age

2 → 0

3 →

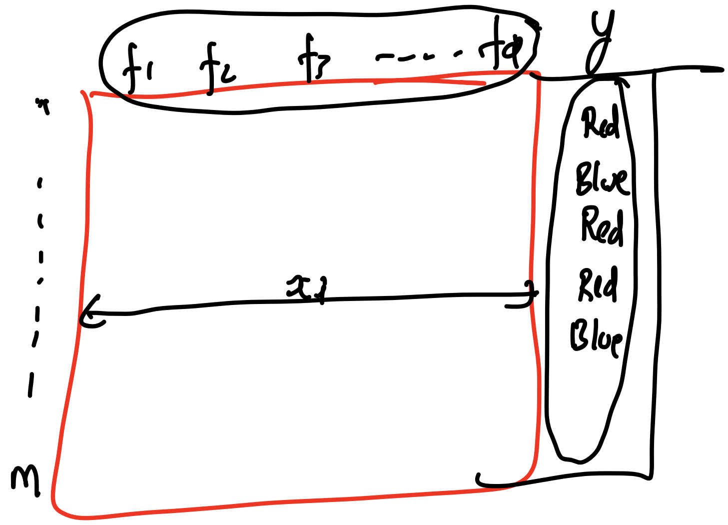
5 →

7 →

10 →

12 →

$$x_{scale} = \frac{X - X_{min}}{X_{max} - X_{min}}$$



d → # features

m → # sample

n → 2

m * d

$$x^i = [x_1^i, x_2^i, \dots x_d^i]$$

y_i $\hat{y}_i$

↓

predicted o/p

Break : 8:32

$$\{x^i, y^i\}^m$$

$$x^i \xrightarrow{\text{ML model}} \hat{y}_i$$

$$\hat{y}_i \approx y^i$$

$X^{i}_{New} \xrightarrow{\text{ML model}} \hat{y}_i$

	x-train	y-train
Training data	1	10
	2	21
	3	30
	5	50
test data	6	61
	7	70
	x-test	y-test

$y = 10x$

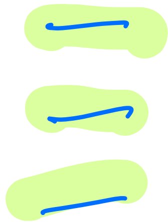
$= 10 * 6 = 60$

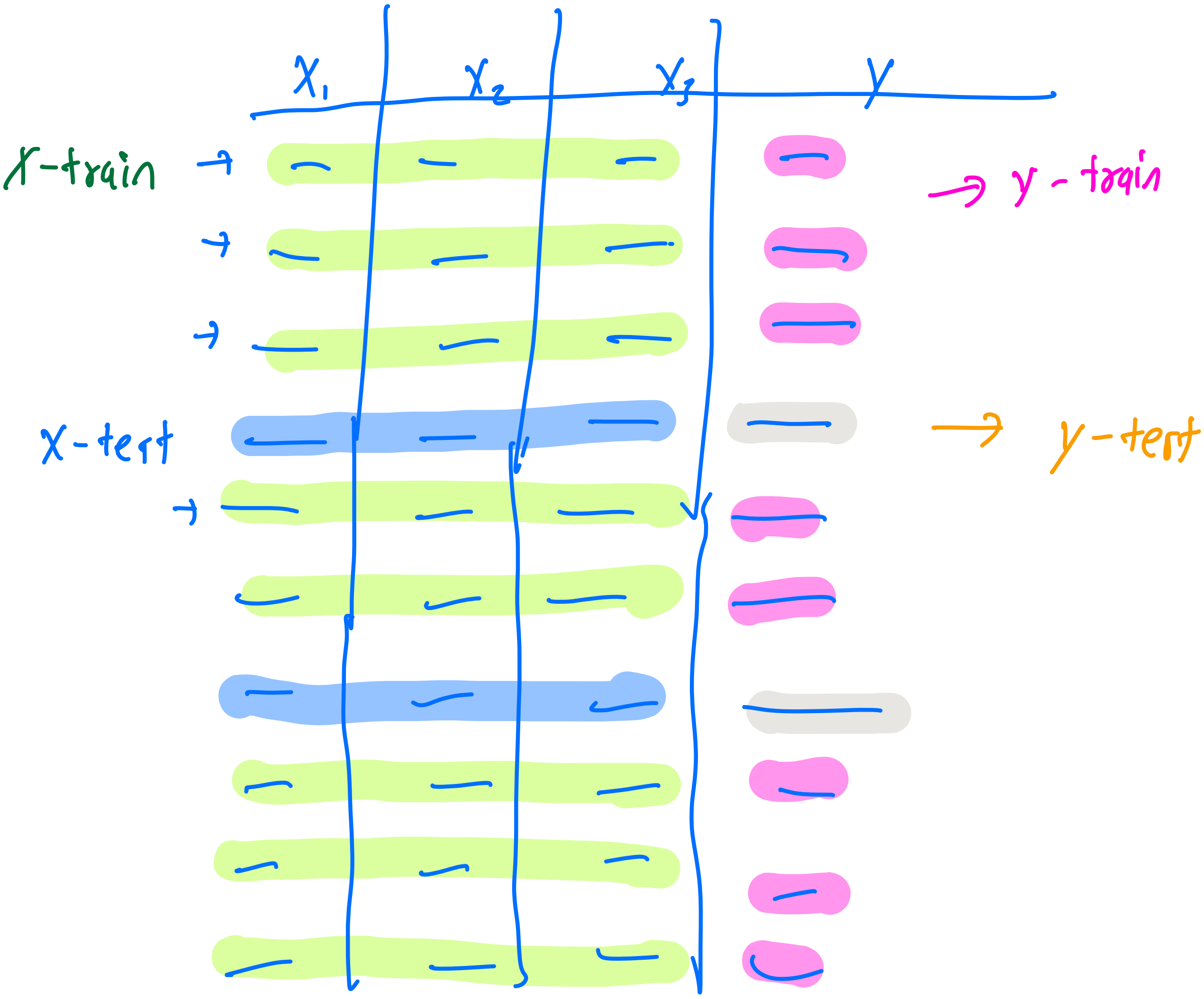
$\Rightarrow 10 * 7 = 70$

80% ↓ training

20% ↓ testing

$\Rightarrow$ func




X1
X2
X3
y
X-train
y-train
X-test
y-test

x_1	x_2	y
1	2	30
2	2	40
3	2	50
3	3	60
5	5	101

x-test

$\Rightarrow$ y-test

$$y = 10x_1 + 10x_2$$

$$10 * 5 + 10 * 5$$

$$\Rightarrow 100$$

CTRL + S

30 ×

Population

UP

120

100 cities

Rahul Gandhi

15

15

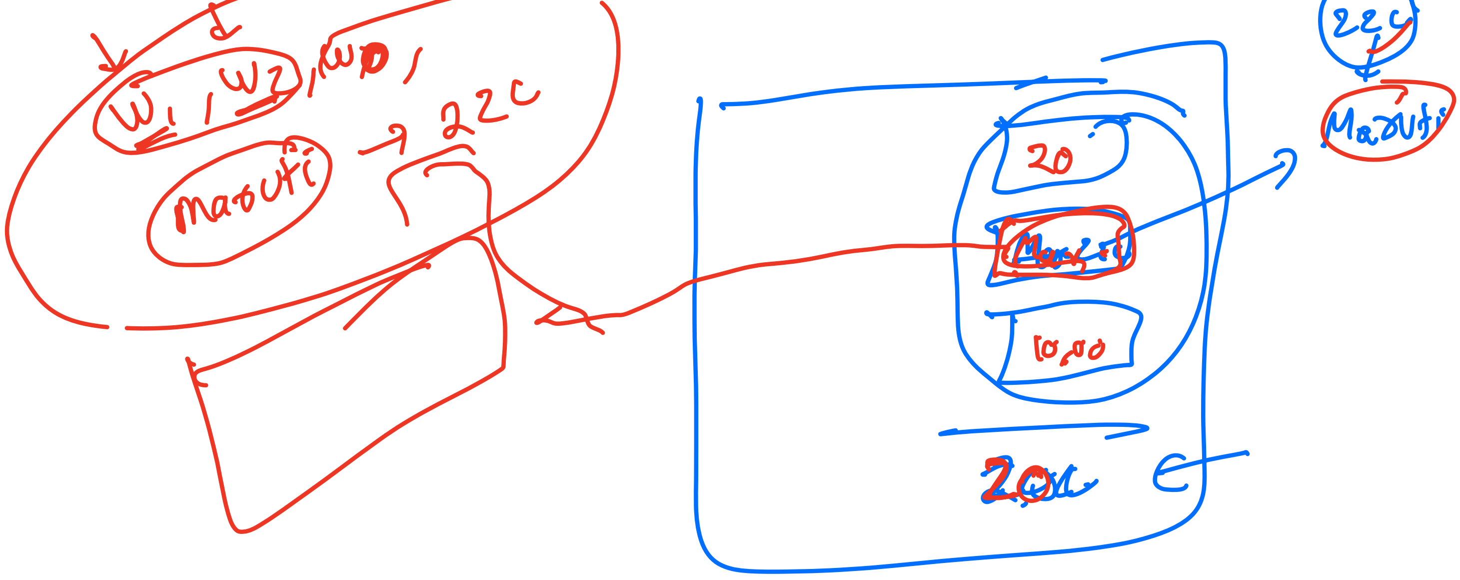
Bangalore

BJP

Rahul Gan

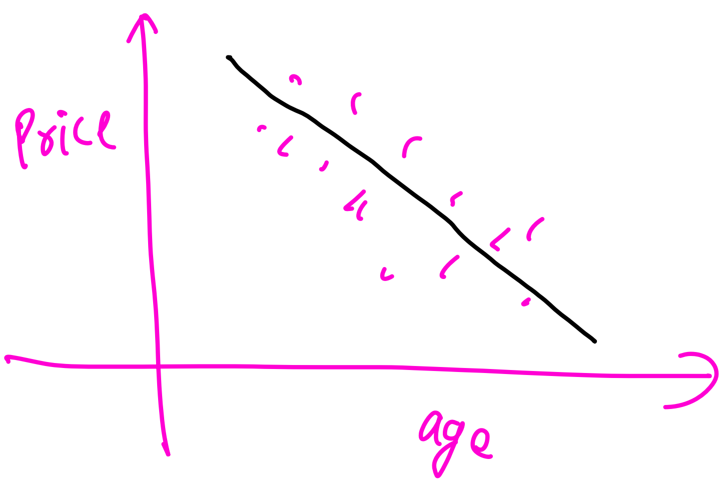
M $w_3x_3 + w_1x_1 + w_2x_2 + w_0 = 0$

1000



Google → Online learning

$$y \simeq \hat{y}$$



$$x^i \xrightarrow{f()} \hat{y}^i$$

$$\hat{y}_i = f(x^i)$$

$10x$

$x^2 + 3$

$\sin(x)$

$$\hat{y}^i = mx + c$$

$m \to w_1$, $c \to w_0$

Price = $w_1 * age + w_0$

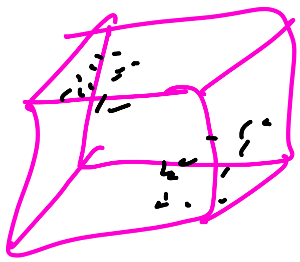
$$y^i = w_1 x + w_0$$

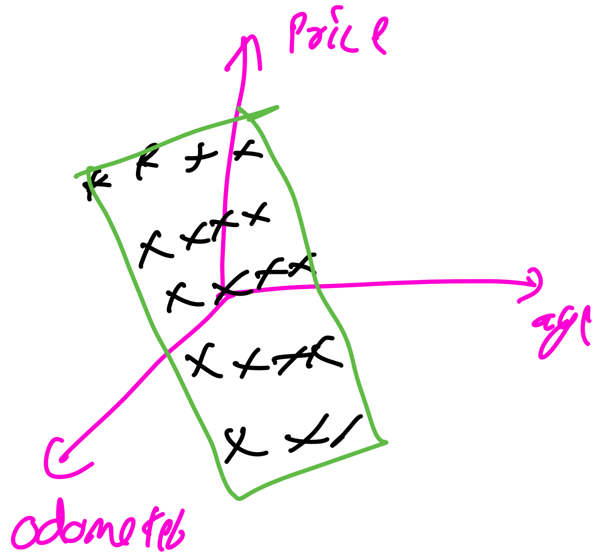
age

$\vec{w}, w_0$

age, odometer

$\hat{y}^i = w_1 x_1 + w_2 x_2 + w_0$





1 feature (Univariate LR) → straigth line

2 —— (Bivariate LR) → Plane 3D

Multivariate (d features) → d+1

$< w_0, w_1, w_2, ---, w_d >$

sklearn

X =

model.fit(x_train, y_train)

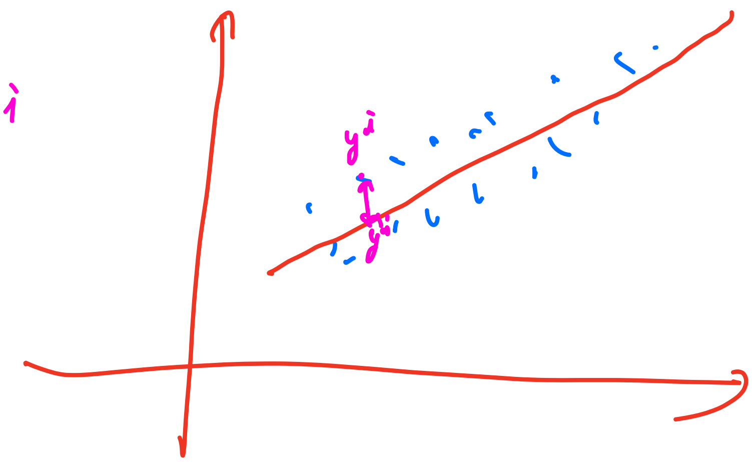
- → w_1 ? → model.coef_
- → w_0 ? → model.intercept_

Is my model Good?

$y^i - \hat{y}^i = error_i$

↓

minimize



$$\min \sum_{i=1}^{m} e_i \rightarrow \boxed{w_0, w_1}$$

$$\sum e^i = 0$$

Mod $\rightarrow$ $\frac{1}{m} \sum_{i=0}^{m} \left| \hat{y}^i - y^i \right| \rightarrow$ MAE

Square $\rightarrow$ $\frac{1}{m} \sum_{i=0}^{m} (\hat{y}^i - y^i)^2 \Rightarrow$ MSE

MSE $\rightarrow$ model1 $\Rightarrow$ 21 ✓

$\rightarrow$ model2 $\Rightarrow$ 31

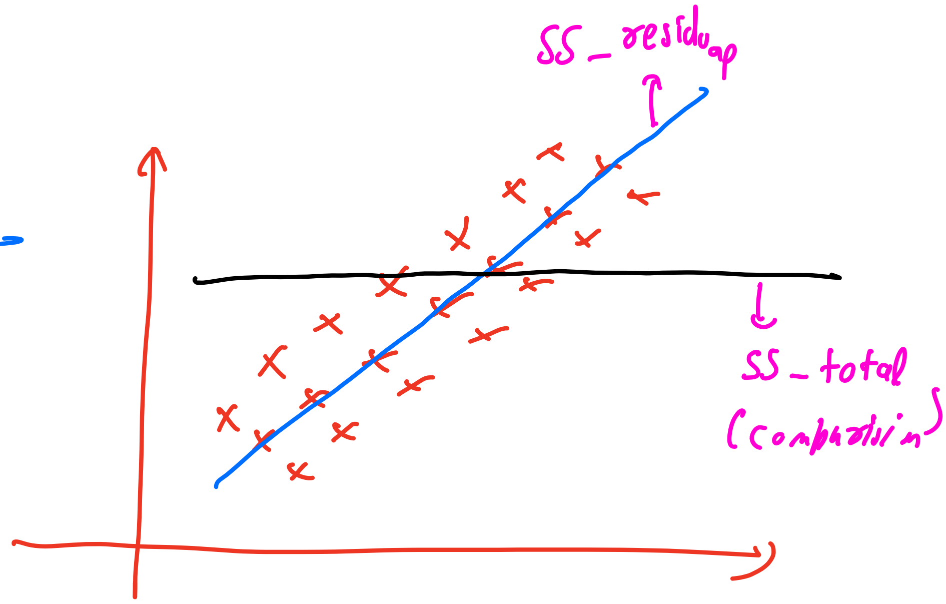
0 to ∞

$\boxed{\text{MSE} \rightarrow 100}$

$\boxed{R^2}$

R^2 score $\rightarrow$ coeff of determination

x_1	x_2	y
—	—	20
—	—	30
—	—	40
—	—	40



$$w_1 x_1 + w_2 x_2 + w_0 = y$$

$$\frac{20+30+40+40}{4} = \frac{130}{4} = 32 \cdot - -$$

$$\frac{SS_{residual}}{SS_{total}}$$

→ low → good

→ high → bad

20

1000

$$\boxed{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}}$$

$SS_{res} \Rightarrow 10,000$

$SS_{avg} \Rightarrow 100$

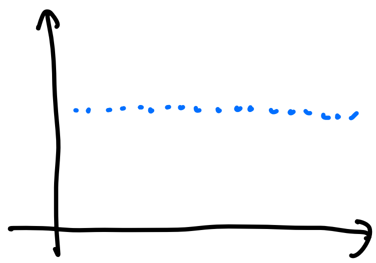
R^2 score $\rightarrow$ 1 $\Rightarrow$ Great model

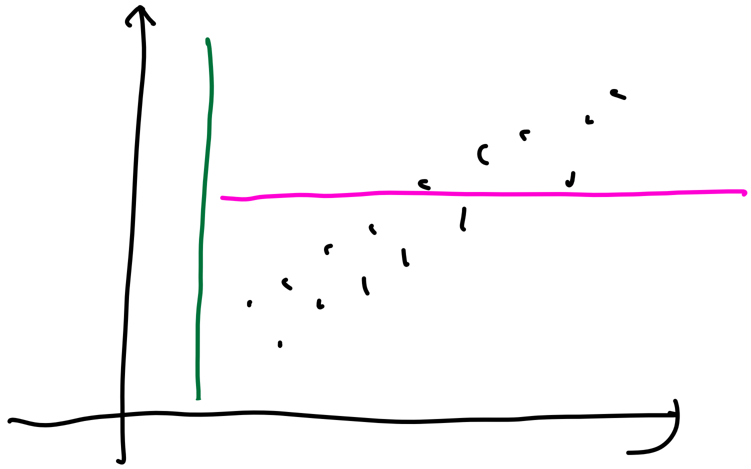
$\rightarrow$ 0 $\Rightarrow$ bad model

$\rightarrow$ $-\infty$ $\Rightarrow$ worst model

R2

R^2





$R^2 \rightarrow 1$

$\rightarrow 0.9$

$\rightarrow 0.8$

0.58

$$\Rightarrow \hat{y}_i = w_1 x_1 + w_2 x_2 + w_3 x_3 + \cdots w_d x_d + w_0$$

$\Downarrow$

Price $\Rightarrow$ $w_0 + w_1 x_1 + (-10{,}000)$ age $+ (10)$ odometer $+ w_0$

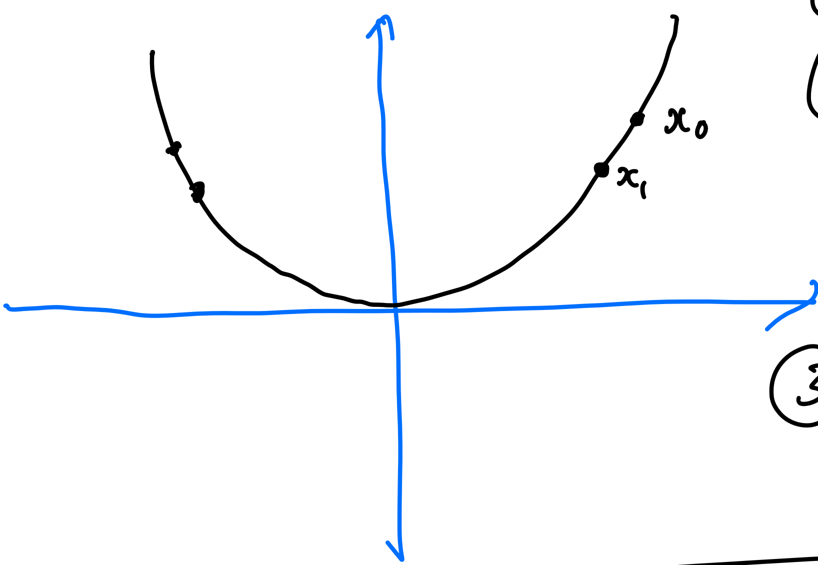
od $\rightarrow$ od $+1$ $\Rightarrow$ Price $\Rightarrow$ Pr $+ 10$

Age $\rightarrow$ Age $+1$ $\Rightarrow$ Price $\Rightarrow$ Pric $- 10{,}000$

Age $\rightarrow [1, 15]$

odo $\rightarrow [5000, 250000]$

$f(x) = x^2$



① Pick x_0 randomly

② $\left.\frac{\partial f}{\partial x_0}\right|_{x_0} \Rightarrow +ve$

③ $x_1 = x_0 + \eta\left(-\frac{\partial f}{\partial x}\right)_{x=x_0}$

Nov22 Tue

Aug22

$$\boxed{R^2 = 1 - \frac{SS_{res}}{SS_{tot}}}$$

$$SS_{res} = \Sigma\, (y - \hat{y})^2$$

$$\Downarrow$$

= np.sum (y - y_) ** 2

y_ = predict (self, X)

ss_tot = np.sum (y - y.mean()) ** 2

score = 1 - ss_res / ss_tot

$d \rightarrow$ columns

$d+1$

$$D \ni \{x^i, y^i\}_{i=1}^{m} \qquad x^i \in R^d,\ y^{(i)} \in R$$

$\hat{y}^{(i)}$ s.t.

$$\hat{y}^{(i)} = f(x^i) = w^T x^i + w_0$$

$\forall i \rightarrow 1$ to m

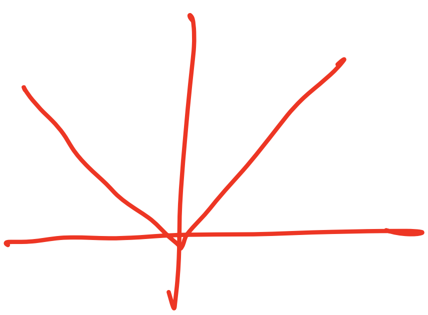
$$w^T = [w_1\ w_2 \ldots\ldots w_d]$$

$$\hat{y} \approx y$$

$\hat{y} \swarrow P$ $\qquad y \searrow R$

$$\boxed{\frac{1}{m} \sum_{i=1}^{m} (\hat{y}^i - y^i)^2} \rightarrow \text{Minimise}$$

$\rightarrow$ Error



$$|\hat{y}^i - y^i|$$

MSE ✓ MAE ×

$$\frac{d\,MAE}{d\,y_{pred}} = \begin{cases} +1 & y_{pred} > y_{true} \\ -1 & y_{pred} < y_{true} \end{cases}$$

$$L = \min_{w_0, w_1} \frac{1}{m} \sum_{i=1}^{m} \left[y^{(i)} - (w_0 + w_1 x^{(i)}) \right]^2$$

$$(2-3)^2$$
$$(3-2)^2$$

$$\frac{\partial L}{\partial w_0} = 0$$

$$\frac{\partial L}{\partial w_1} = 0$$

$$\vdots$$

$$\frac{\partial L}{\partial w_d} = 0$$

$$L(w_1, w_2 \ldots, w_0) = (y - (w_2 x_2 + w_1 x_1 + w_0))^2$$

$$\frac{\partial L}{\partial w_0} = -2(y - \hat{y})$$

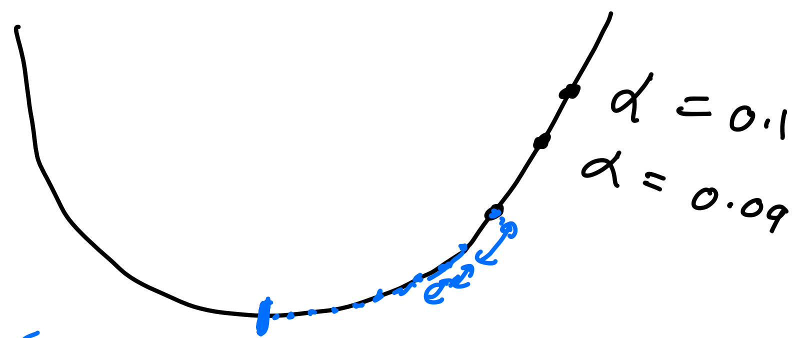
$$\frac{\partial L}{\partial w_1} = -2(y - \hat{y}) \cdot x_1$$

$$\frac{\partial L}{\partial w_2} = -2(y - \hat{y}) \cdot x_2$$

$$\boxed{\begin{aligned} \frac{\partial L}{\partial w_0} &= \frac{1}{m} \sum_{i=1}^{m} -2(y - \hat{y}) \\ \frac{\partial L}{\partial w_d} &= \frac{1}{m} \sum_{i=1}^{m} -2(y - \hat{y}) \cdot x_d \end{aligned}}$$

$$w_0 = w_0 - \alpha \frac{\partial L}{\partial w_0}$$

$$w_d = w_d - \alpha \frac{\partial L}{\partial w_d}$$



$$\alpha = \frac{0.1}{1} = 0.1$$

$$\alpha = \frac{0.1}{2} = 0.05$$

$$\alpha = \frac{0.1}{3} = 0.033$$

$$\alpha = \frac{0.1}{4} = 0.025$$

$$\alpha = \frac{\alpha}{n}$$

$$\frac{\partial L}{\partial w_0} = -2(y - \hat{y})$$

$$\frac{\partial L}{\partial w_1} = -2(y - \hat{y}) \cdot x_1$$

$$\frac{\partial L}{\partial w_2} = -2(y - \hat{y}) \cdot x_2$$

```
y_pred = self.predict (self.X)

dw = -(2* (self.X.T).dot (self.y-
                y_pred)) / self.m

db = -2 * np.sum (self.y - y_pred)/self.m
```

self.w = self.w - self.learning_rate * dW

self.b = self.b - ______ * db

np.dot(~~w~~x, ~~x~~) ⇒ $w^T x$

X, w   x ^ ~~x~~w.

b -> bias
↓
w0

$\boxed{x^T\text{.dot}(w)}$

[- - -] [- - -]

[- - -] [- - -]

```
def fit( x, y):
    m , d = X.shape
    w = np.zeros(self.d)
    b = 0
    error_list = [ ]
    for i in range(iteration):
        update_weights( )
        y_pred = x.dot(w) + b

error = np.square(np.substract(y, y_pred)).mean()
    error_list.append(error)
```

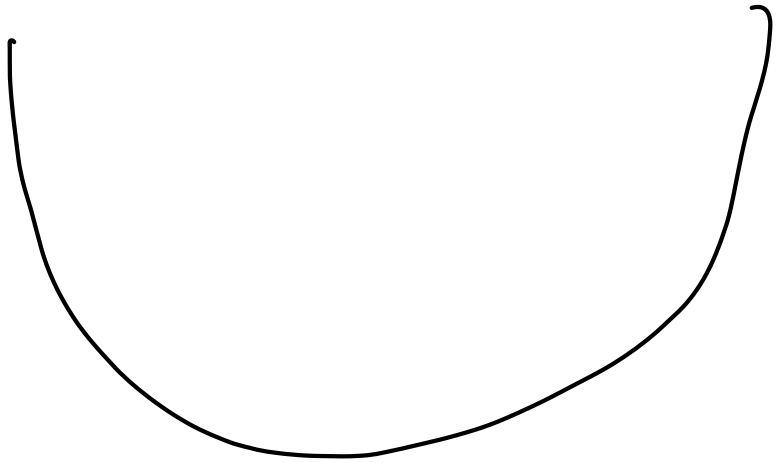
np.random(___ , random_state = 13)

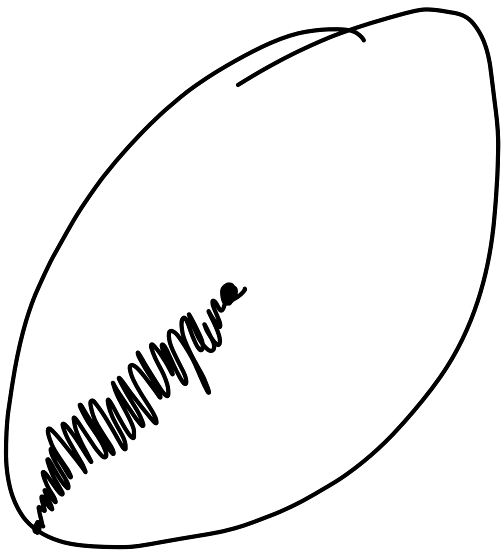
1 to 10

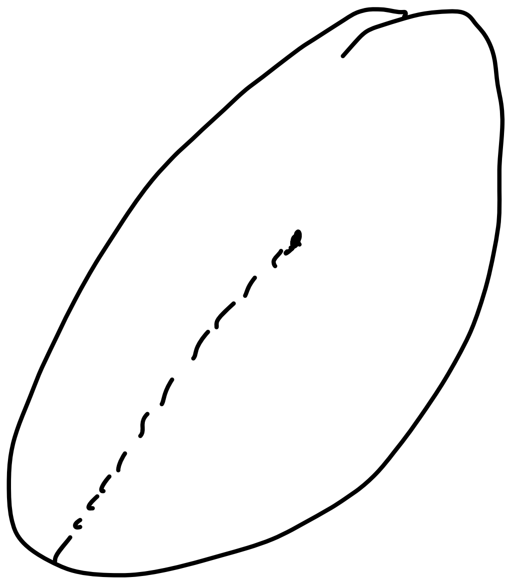
6

np.random(___ , random_state = 13)

6







Adjusted R^2

N feature $\Rightarrow R^2$

+

1 feature

$$w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + w_{n+1} x_{n+1}$$

Adjusted R^2

$$= 1 - \left[\frac{(1-R^2)(n-1)}{(n-d-1)} \right]$$