Machine_learning
Wednesday, June 24, 2020 9:12 AM

Data Science

Raw data
↓ Transform
Structured valid Data
↓
Analysis & Visualization [finding pattern]
↓ ↓
→ new → Machine Learning → Algo
↓
prediction

Bike → Error / Accuracy — high → low — 30 → 90

Exp. → fosh ← Data

Traditional Programming

Prime — code → m/c → outcome
1,2,3,5,9,100,200 data →
1 2 3 5 9 100 200
F T T T F F F

→ Machine Learning

Algo ↓

F T T F T F T F F F
(target) out come (output) → m/c Learn / fitting → program or model → prediction
1 2 3 4 5 6 7 8 9 10
(features) data (input) →

13 65 121 14
new features ↓ program or model
prediction: T F F F

→ Types of Machine Learning

→ Supervised → we have i/p & o/p data, and Algo find pattern to generate o/p from i/p

→ Unsupervised

→ semi-supervised

→ Deep Learning
NN

→ re-inforcement

→ Supervised ML →

Regression — continous prediction
— categorical prediction

Regression — continous prediction

Classification — categorical prediction

features → 1 2 3 4 5 6 9 (input data)

Label F T T F T F F (outcome)

$\boxed{\text{Algo}}$

i/p → $\boxed{\text{model}}$ → o/p

Houses data

features

#rooms	Area	Crime	tax
3	200	0.3	0.5
10	3000	0.1	0.6
–	–	–	–
–	–	–	–
–	–	–	–

X_1 X_2 X_3 X_4

continous

target outcome

price
10
100
–
–
–

Y

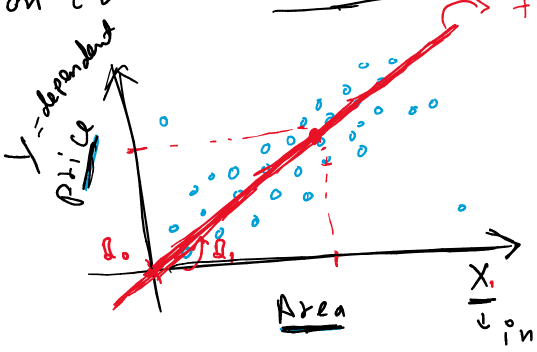
continous

$price = \beta_0 + \beta_1 \#Rooms + \beta_2 Crime + \beta_3 tax$

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$

Regression → Predicting value of a dependent variable on the Basic indendent

→ Corr coef



fitting line

$y = mx + c$

error = $y_{actual} - y_{pred}$

X → features

Y → target/outcome

Regression

$y = f(x)$

estimate, some change in X

Algo

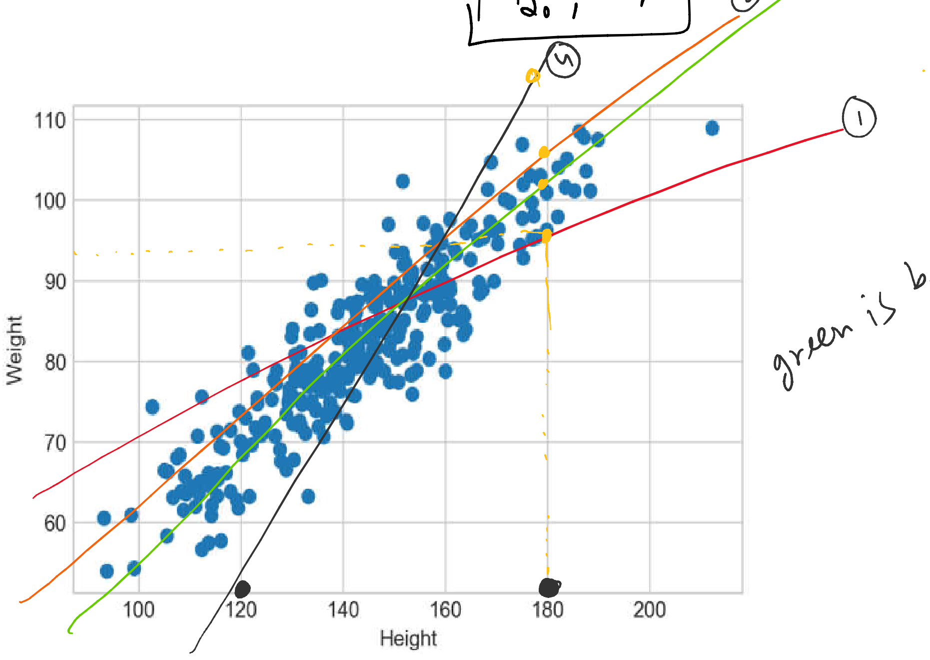
$\boxed{y = \beta_0 + \beta_1 X}$

target, Intercept, coeff, feature

here learning or fitting is all about finding the values of parameters based on given i/p & labels

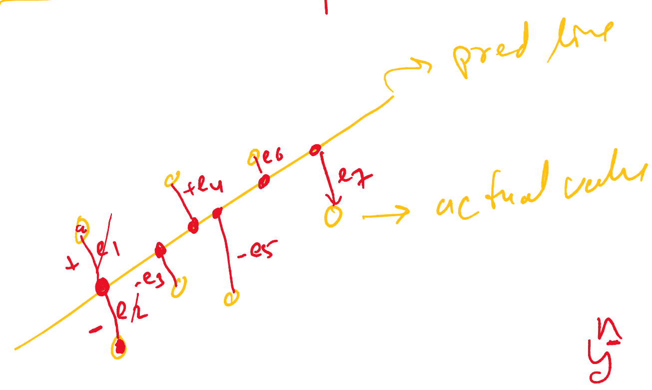
Parameters → β_0, β_1 ← [X ✓, Y ✓]

$\boxed{\hat{\beta}_0, \hat{\beta}_1}$ → Regression



green is best fit ?

$$\text{Actual Error} = \left| y_{actual} - Y_{pred} \right|$$



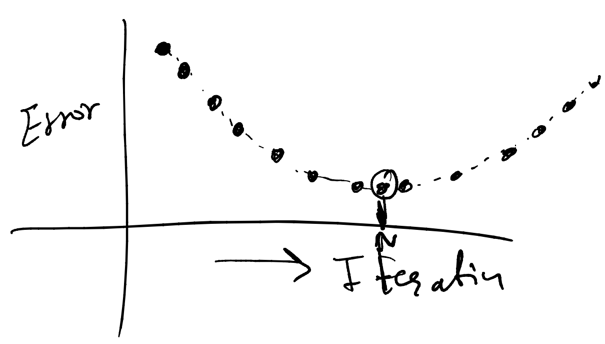
$\hat{y}$ → prediction

Cost function →

$$\text{absolute total Error} = \sum_{i=1}^{N} \left| y_i - \hat{y}_i \right|$$

$$\boxed{\text{mean squared Error} = \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i \right)^2}{N}}$$

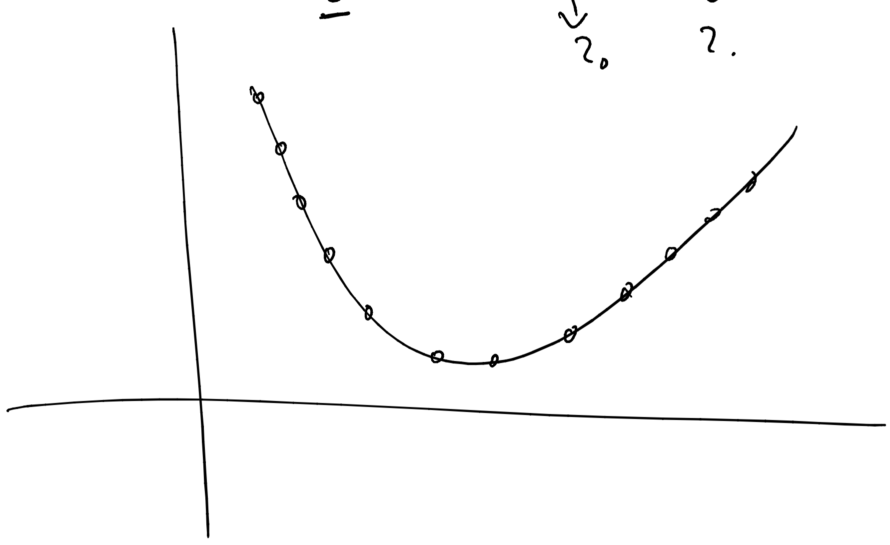
→ to find best parameters we try to reduce cost function



$X < \begin{matrix} \beta_0 \\ \beta_1 \end{matrix}$

$y - \hat{y}$

$$\hat{y} = \underset{?}{\beta_0} + \underset{?}{\beta_1} X \qquad (1)$$



$$f(x) = ax^2 + bx + c$$

→ Calculus

$$f(x) = 2x^2 + 3x + 10$$

$$\boxed{f'(x) = 4x + 3}$$

$$4x + 3 = 0$$

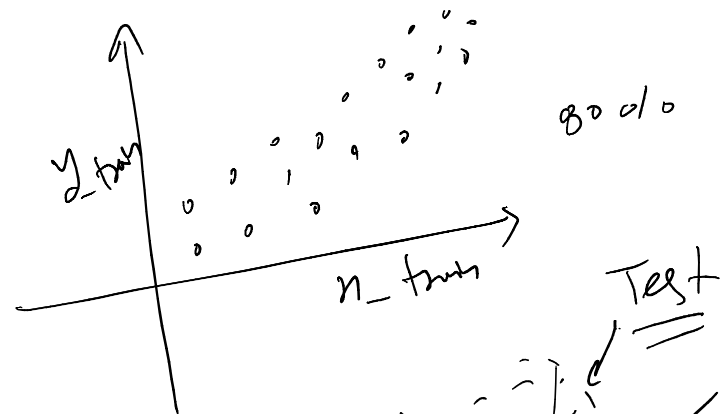
→ $x = -3$

→ $x = -3/4 = -0.75$

Cost

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

(L2 NORM LOSS)

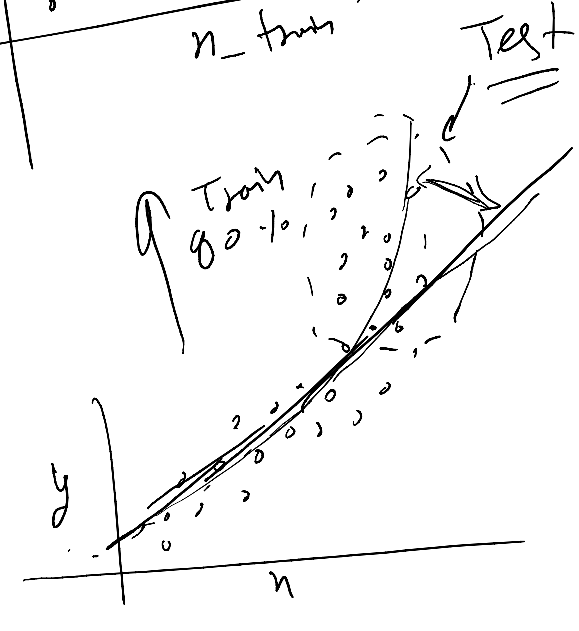


$ax^2 + bx + c$ — 2o

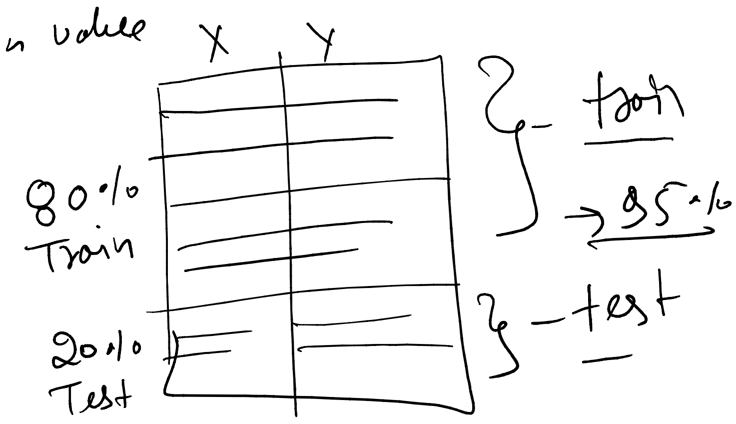
f(x) minimize

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X \ldots (i)$$

$$f(x) = \frac{\sum_{i=1}^{N}\left(y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)\right)^2}{N}$$



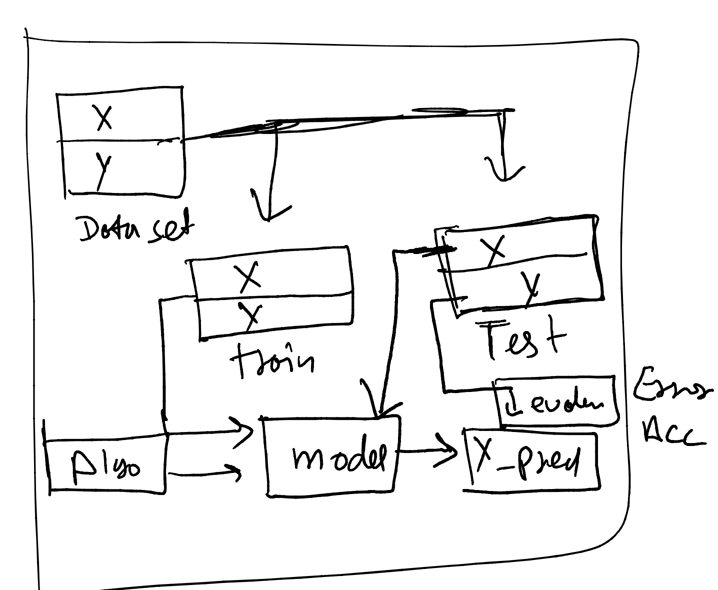
just differentiate wrt $\hat{\beta}_0$ and $\hat{\beta}$ and min value of equation



Learning

$$\boxed{\hat{\beta}_1 = \frac{\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N}(x_i - \bar{x})^2}} \quad (i)$$

$$\boxed{\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}} \longrightarrow (ii)$$



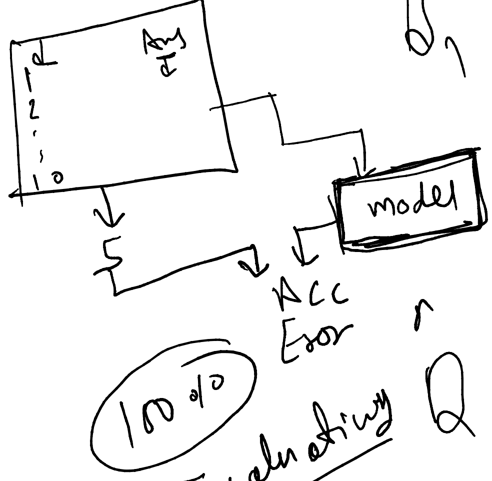
y = weight, x = height } → β_0, β_1 → ?

model $\boxed{\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X}$ new value

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 \ldots$$

$$\hat{\beta}_0 = \bar{y} - \left(\hat{\beta}_1 \bar{X}_1 + \hat{\beta}_2 \bar{x}_2 + \hat{\beta}_3 \bar{X}_3 \ldots \bar{\beta}_n \bar{X}_n\right)$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{N} (x_1 - \bar{x}_1)(y_i - \bar{y})}{\sum_{i=1}^{N} (x_1 - \bar{x}_1)^2}$$

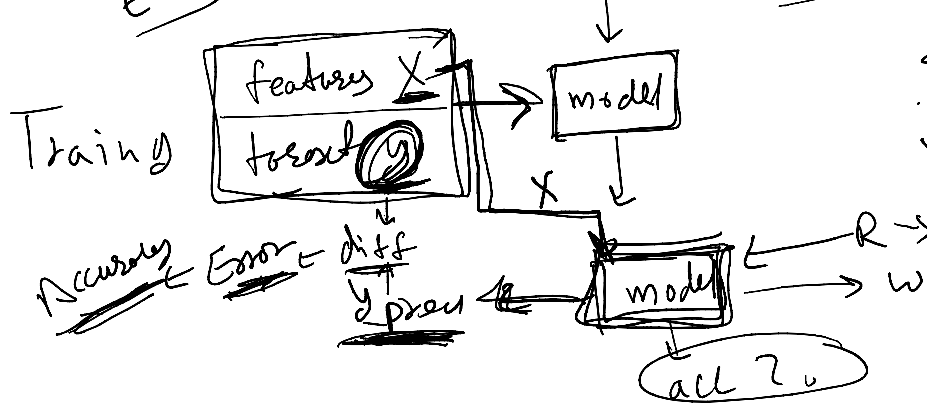


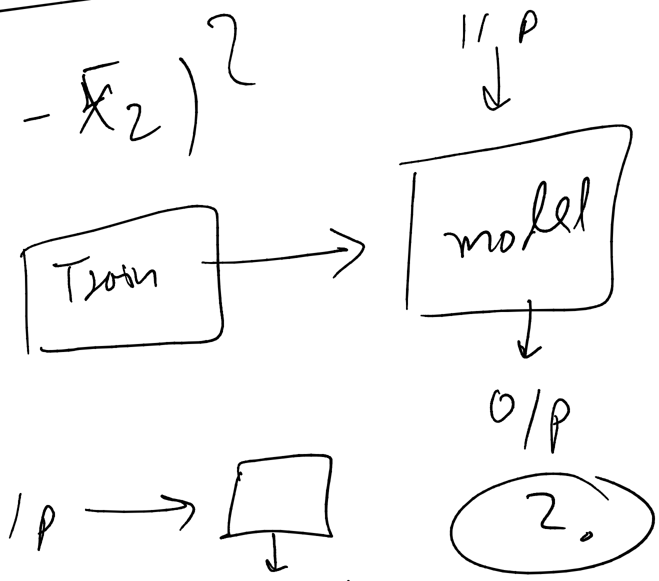
2. Biasing

Evaluating

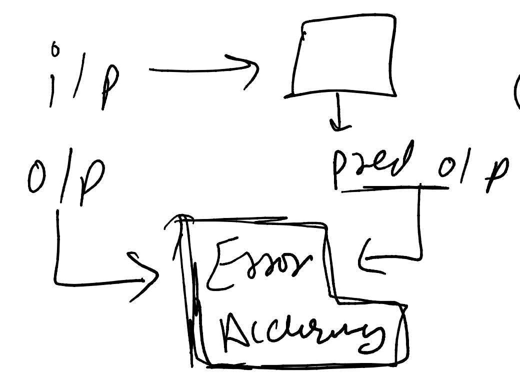
$$\hat{\beta}_2 = \frac{\sum_{i=1}^{N} (X_2 - \bar{X}_2)(y_i - \bar{y})}{\sum_{i=1}^{N} (X_2 - \bar{X}_2)^2}$$

Traing





→ multiple Linear Regression



→ ML work-flow

1. Load your data
2. EDA to check - which Algo to use, and feature selection
3. Split your data into traing and testing data set
4. model selection
5. model training (finding parameters)
6. model Evaluation (Accuracy, Error)
7. model Deploy
8. model Assement