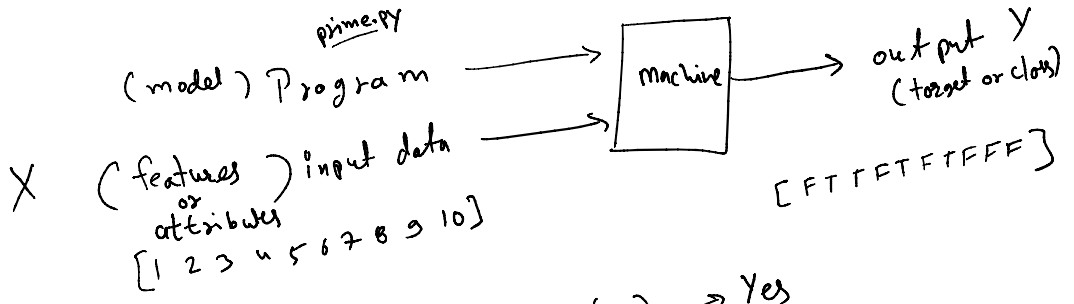
MachineLearning
Thursday, June 25, 2020 12:54 PM

Traditional x Programming



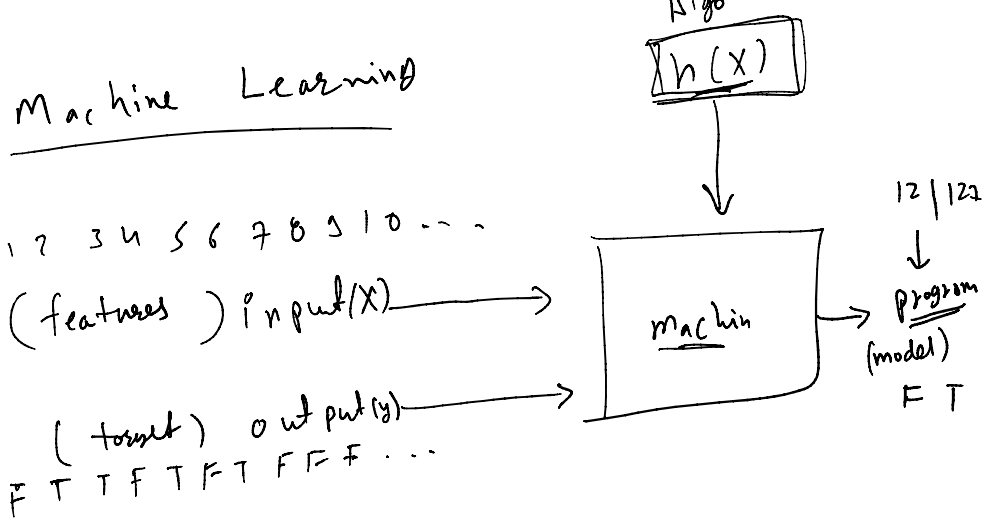
$y = h(X)$

hypothisis (X) → Yes / No

$y \in$ yes / No

$h(n)$ → yes if n%p != 0 where $p \in [2, n-1]$ / No

Machine Learning

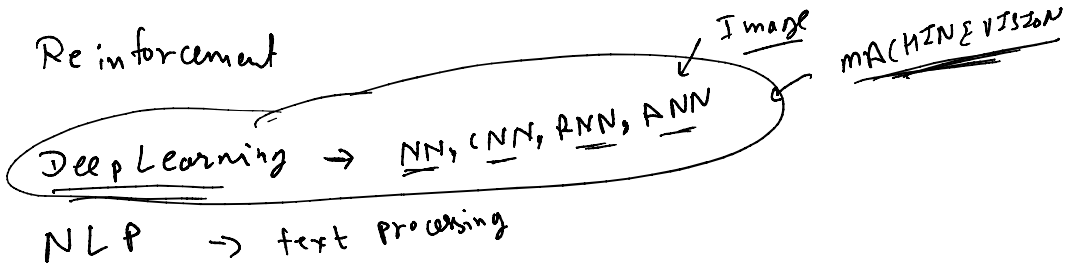


→ Supervised ML (Labeled Data)

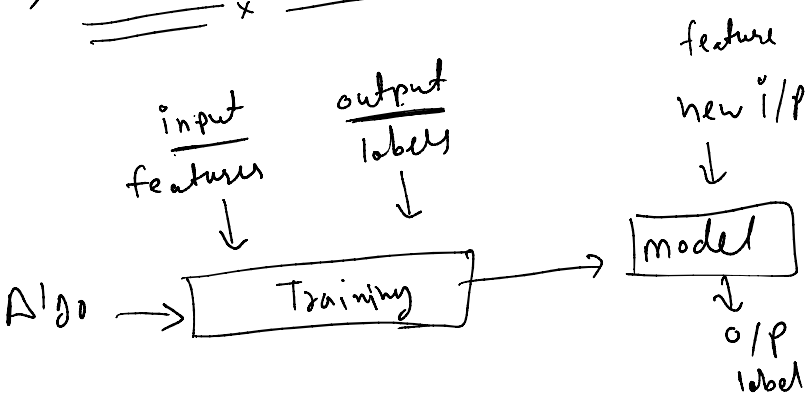
→ Unsupervised ML (Unlabeled Data)

Semi-Supervised (labeled + Unlabeled)

Reinforcement



→ Supervised machine Learning



eg: Regression → Continous target

Classification → Categorical target

Regression →

... of a dependent variable on the

Regression →

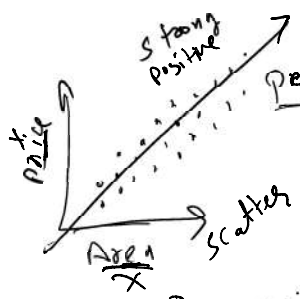
Predicting value of a dependent variable on the basis of independent variables.

$$\underset{\downarrow}{y} = h(x) = f(\underset{\downarrow}{x})$$

dependent variable — independent variable

→ study hours — independent variable

→ percentage result — dependent variable

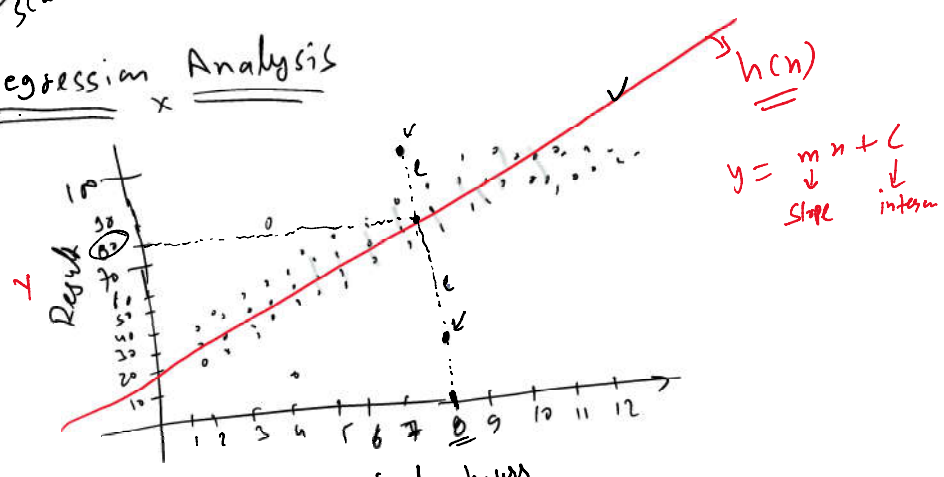


Percentage result = some factor × study hour + Bias

$\beta_1 \times X + \beta_0$

Result ← h (study hours)

→ Regression Analysis



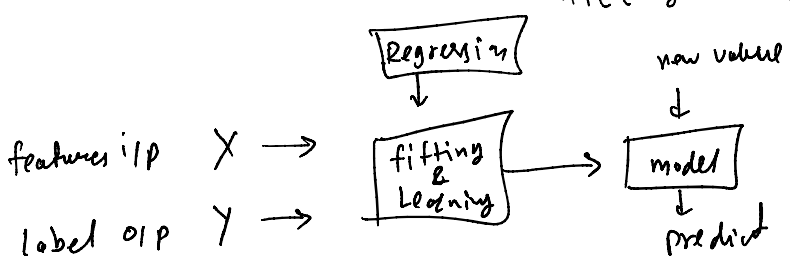
$y = h(x)$

$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 X$

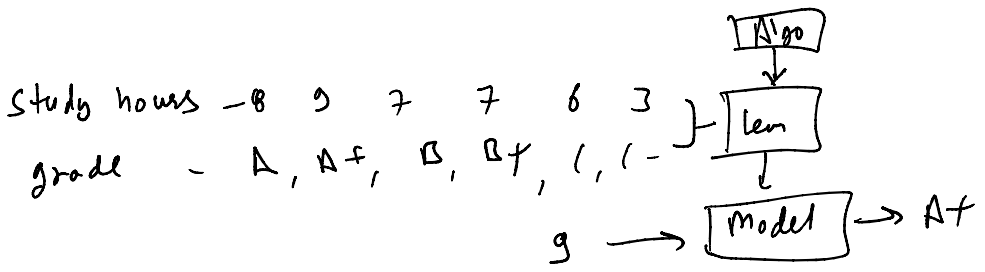
$\hat{\beta}_0$ — intercept, $\hat{\beta}_1$ — coeff

parameters β_0, β_1

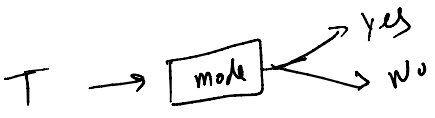
the process of finding the value of the parameter is known as fitting or Learning.

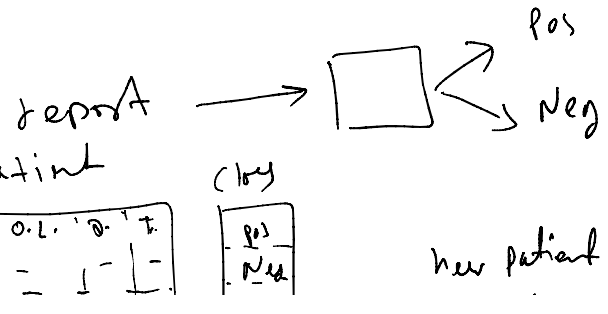


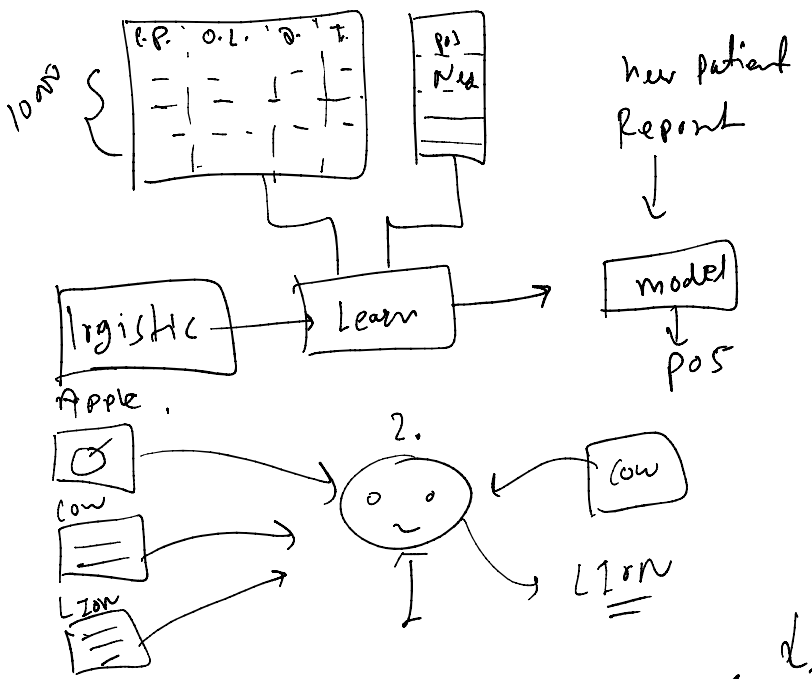
Supervise Model → Regression / classification



Fraud detection







Navie Bays

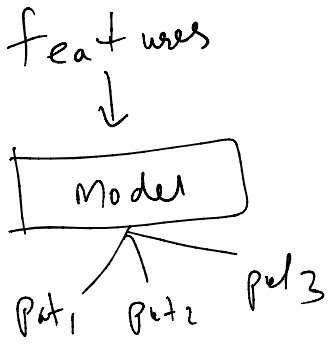
$$P(A|B) = \frac{P(A)\,P(A \cap B)}{P(B}$$

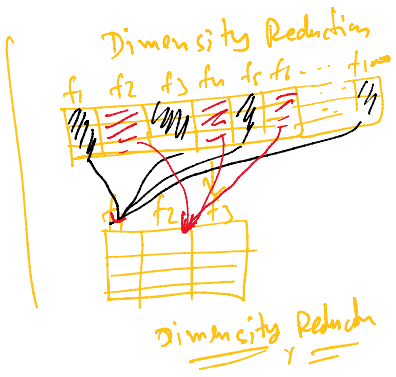
$h(X)$

$\Rightarrow$ supervised Learning

- Regression – Linear Regression, Polynomial Regression
- classification – Logistic Regression, Navie Bays, Decesion Trees, Random forest, K-NN, SVM

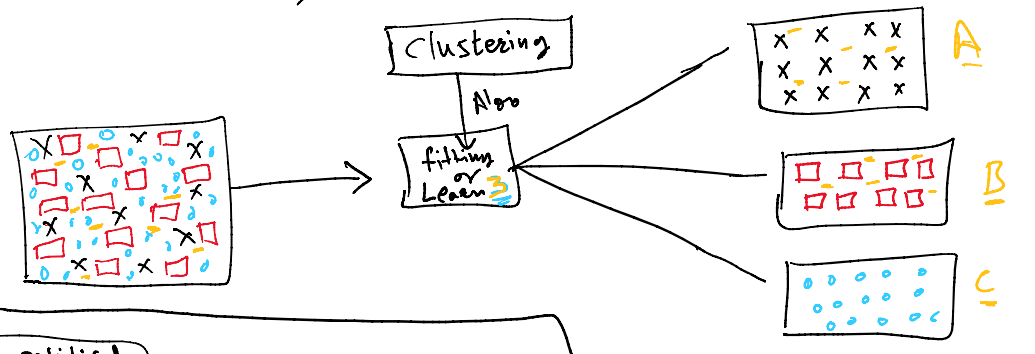
$\Rightarrow$ Un-supervise Machine Learning $\rightarrow$



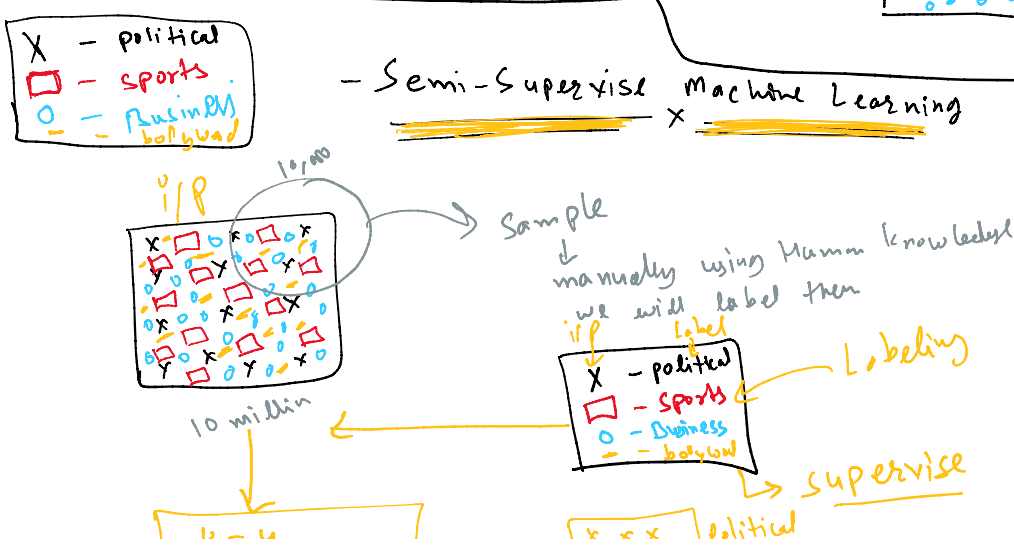


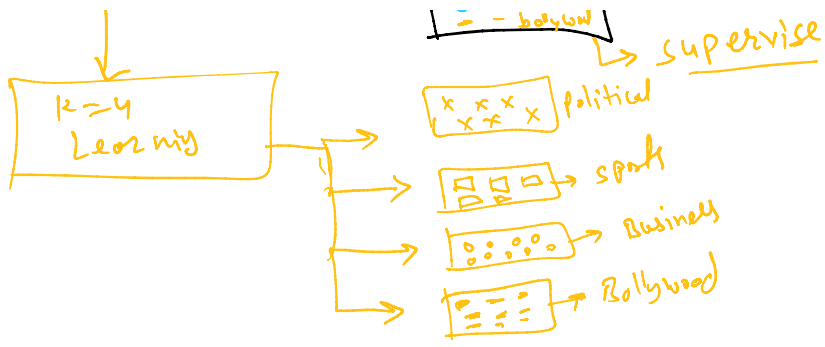
Unsupervise

- Clustering – K-means
- Dimensity Reduction – PCA, LDA

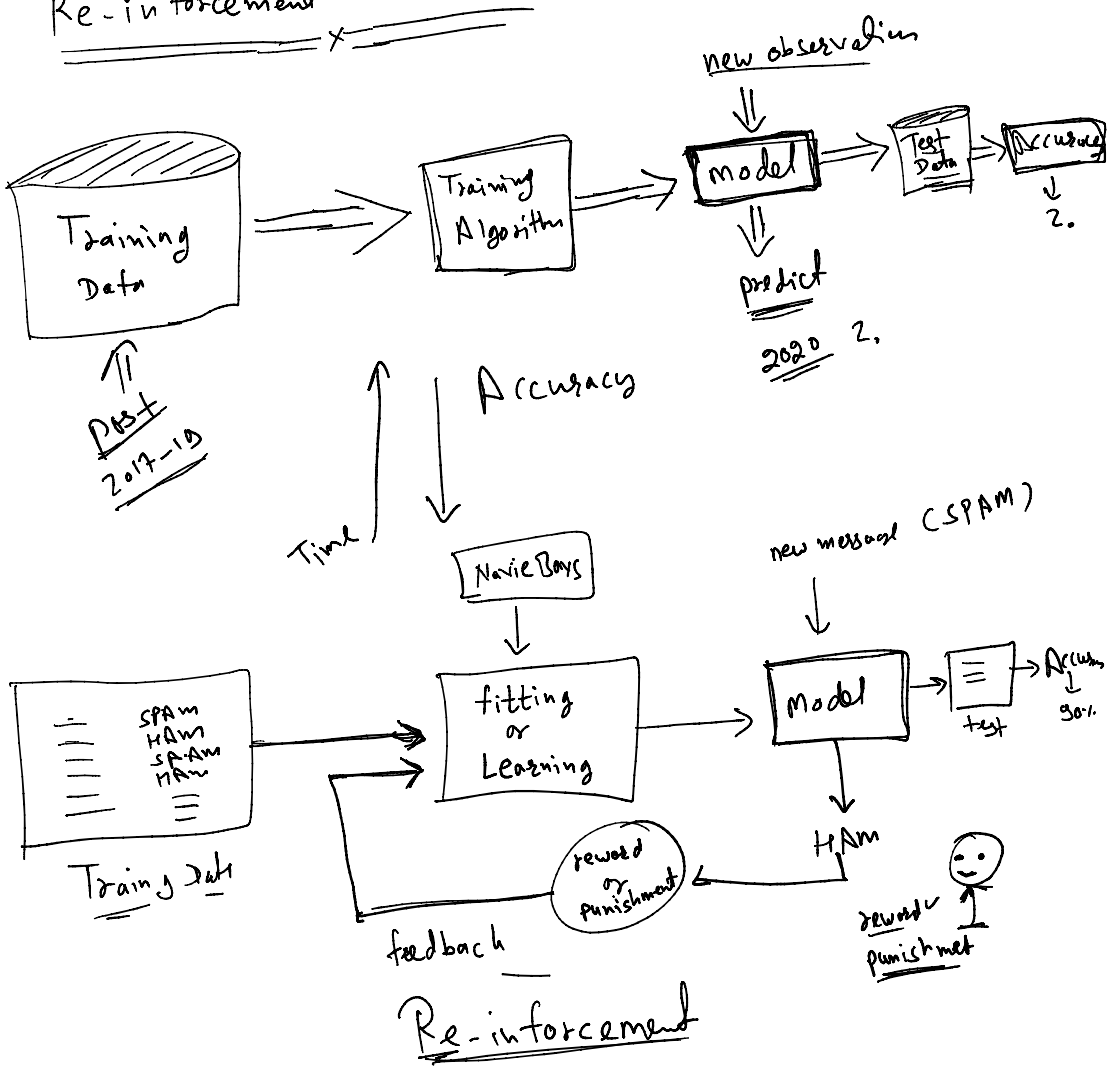


- Semi-Supervise Machine Learning





→ Re-inforcement Learning

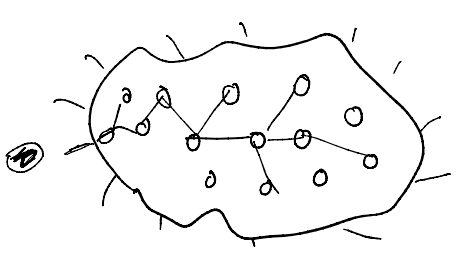


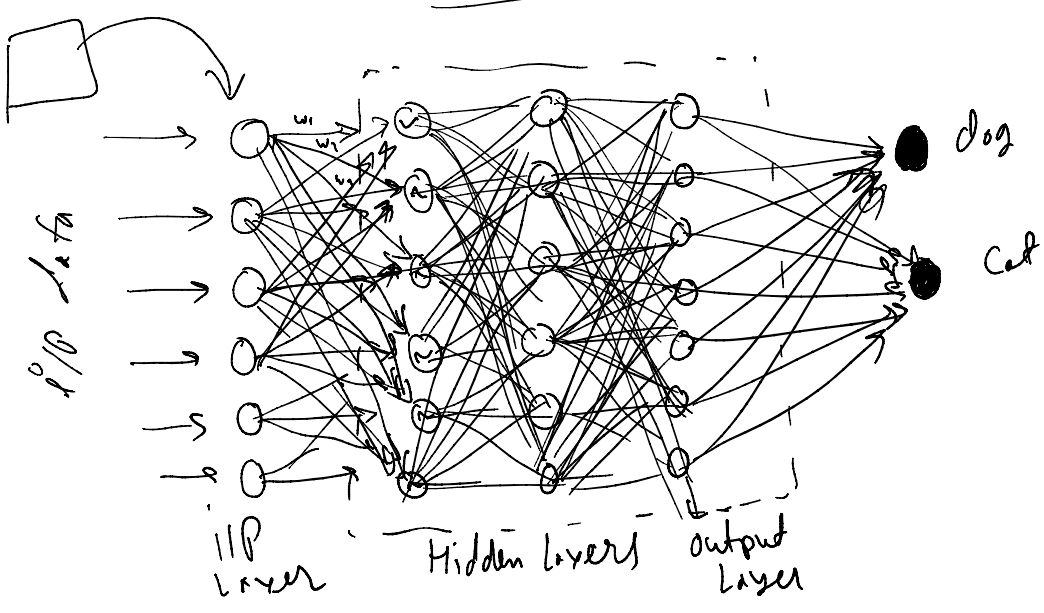
→ Deep Learning

Structured Data

Unstructured Audio, Video, Images

Action millions neurons





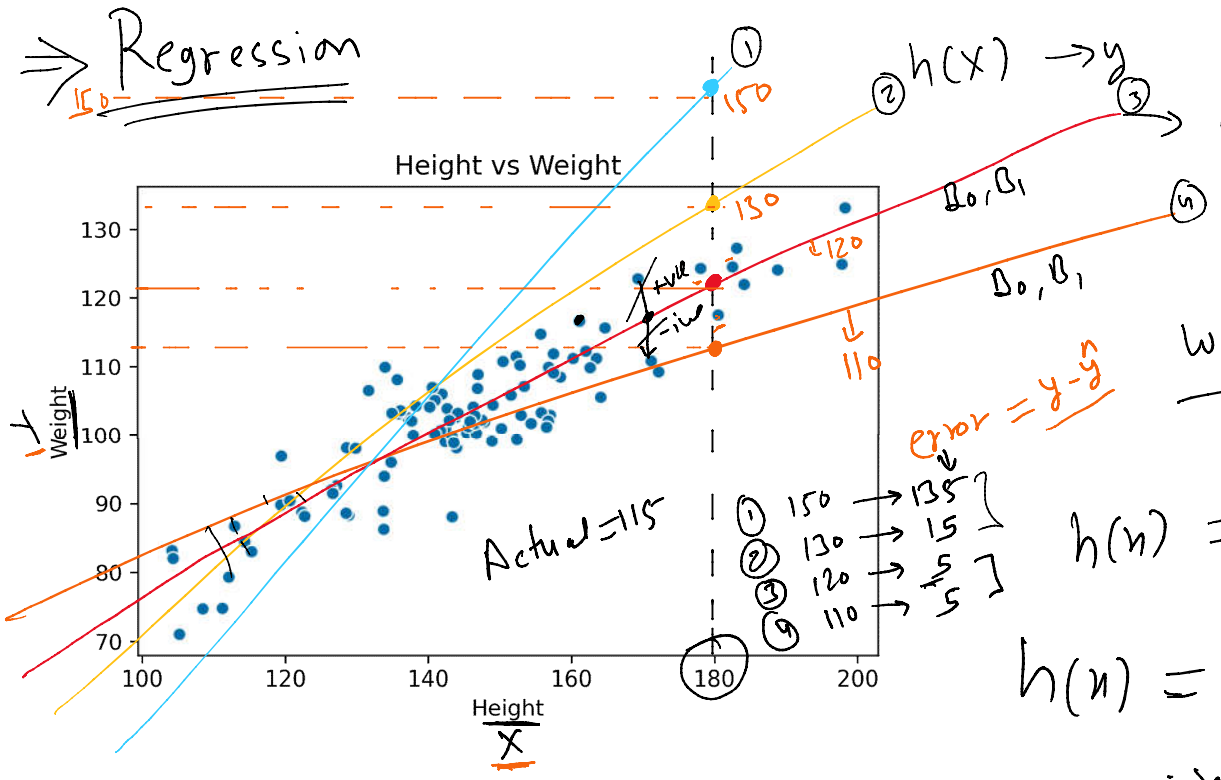
NLP → Natural Language Processing

① Language Understanding

① Language Understanding

② matrix & vectors

⇒ Regression



① ② ③ ④ fitting lines

best fit

$h(x) \rightarrow y$

B_0, B_1

B_0, B_1

Why 3 ?

error $= y - \hat{y}$

① 150 → 135
② 130 → 15
③ 120 → 5
④ 110 → 5

$h(x) = mx + c$

$h(x) = B_0 + B_1 X$

intercept, coeff

→ Cost function / Error function / Loss function

$C(f) = 2.$

① $C(f) = 100$
② $C(f) = 80$
③ $C(f) = 2$
④ $C(f) = 60$

Mean Absolute Error

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N}$$

y = Actual weight

$\hat{y}$ = predicted weight

Mean Squared Error →

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

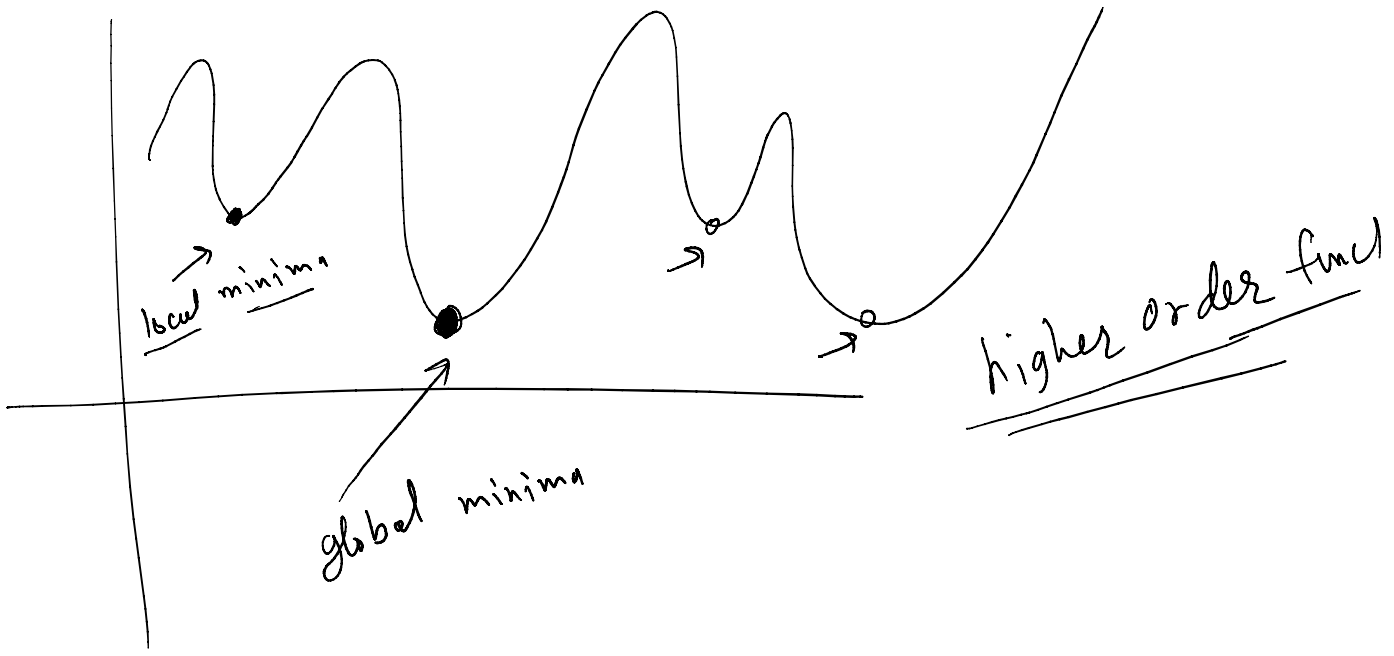
L2 - NORM LOSS

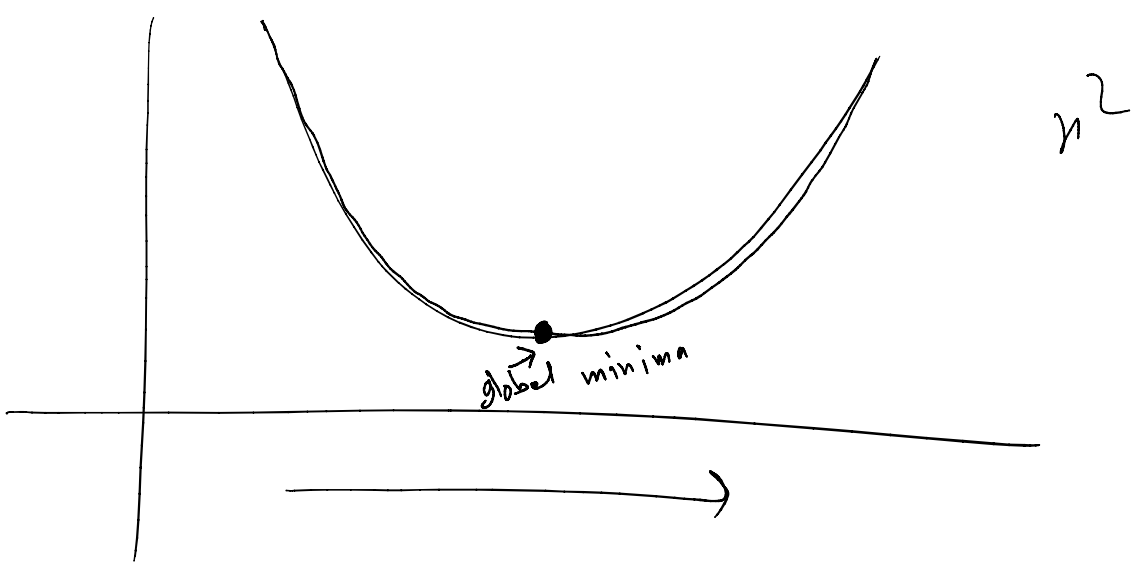
$$L(f) = \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \quad , \quad \hat{y} = mx + c$$

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

min value of this function

Calculus $\frac{\partial f(x}{\partial x} = 0$ → min value





$$C(x) = \frac{\sum_{i=1}^{N} \left(y_i - \hat{y}_i\right)^2}{N}$$

x → height
y → weight

$$h(x) = \beta_0 + \beta_1 x$$

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_i$$

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

↓ minimize this cost function

finding Best value for parameters B_0 and B_1 is known as Learning or fitting.

$$\frac{\partial Cf(n)}{\partial B_0} = 0 \quad \text{---- (i) } B_0 = ?.$$

$$\frac{\partial Cf(n)}{\partial B_1} = 0 \quad \text{---- (ii) } B_1 = ?.$$

$$B_1 = \frac{\sum_{i=1}^{N} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{N} (x_i - \bar{x})^2} \quad \text{--- (1)}$$

remember this

$y = mx + c$

$m = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)$

$$B_0 = \bar{y} - B_1 \bar{x} \quad \text{--- (2)}$$

$x, y \rightarrow \hat{B}_0, \hat{B}_1$

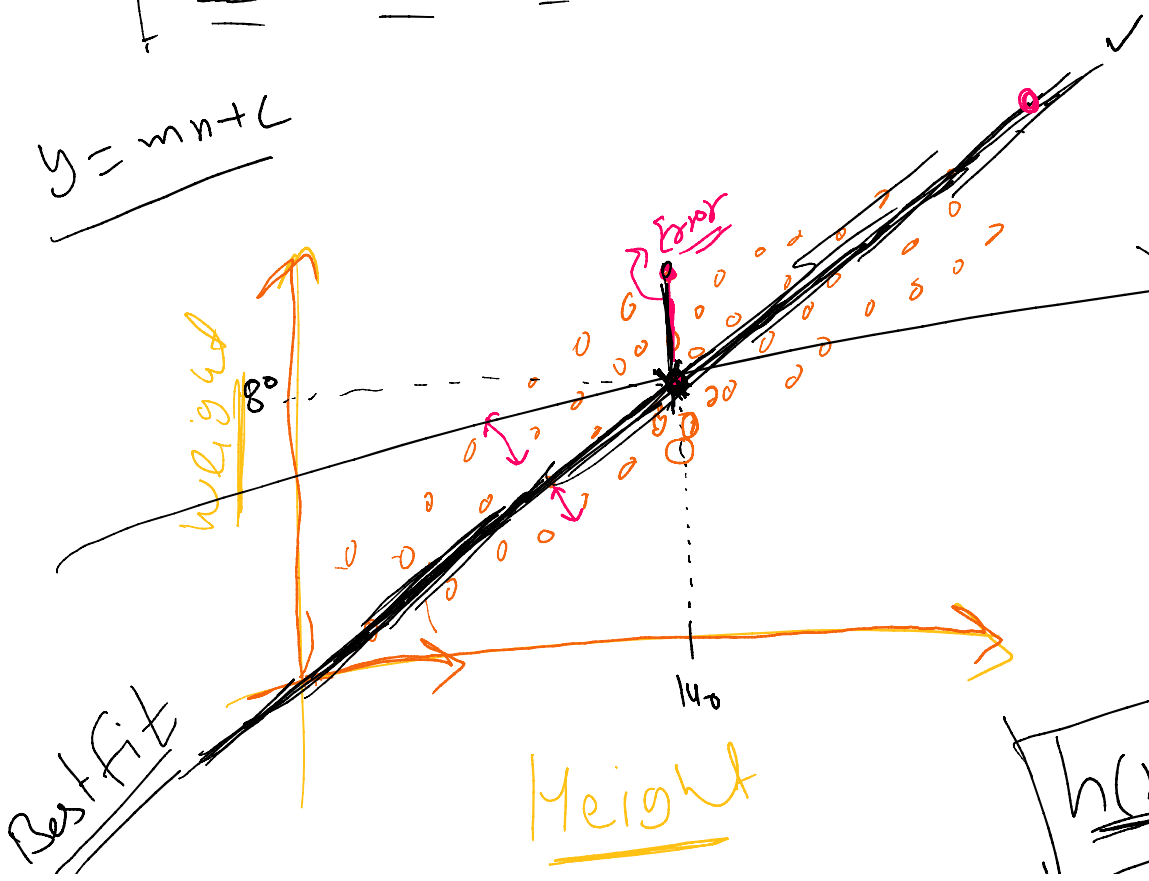
Training → Parameters

ML

$\hat{B}_0, \hat{B}_1$

$$h(x) = \hat{B}_0 + \hat{B}_1 X$$

$y = mx + c$



Continous data

$h(x) \rightarrow$

weight

$$h(x) = \underset{c}{\hat{B}_0} + \underset{m}{\hat{B}_1} X$$

Best

Weight

$$\text{weight} = \hat{B}_0 + \hat{B}_1 * \text{height}$$

$$\hat{y} = \hat{B}_0 + \hat{B}_1 X$$

prediction

2. 2.

hypothesis

i/p
X
80 kg
70 kg

o/p
y
5.10
5.5

actual

model

5.5

prediction

5.10

$\hat{y}$

Cost function

$$\text{Error} = y_{actual} - \hat{y}_{pred}$$

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

$error_1 = 5.10 - 5.5 \Rightarrow 0.5$

$error_2 = 5.5 - 5.10 \Rightarrow -0.5$

total Error = 0

$\Rightarrow$ L2 - NormLOSS Error

→ Mean Squared Error

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

→ we want to minimize

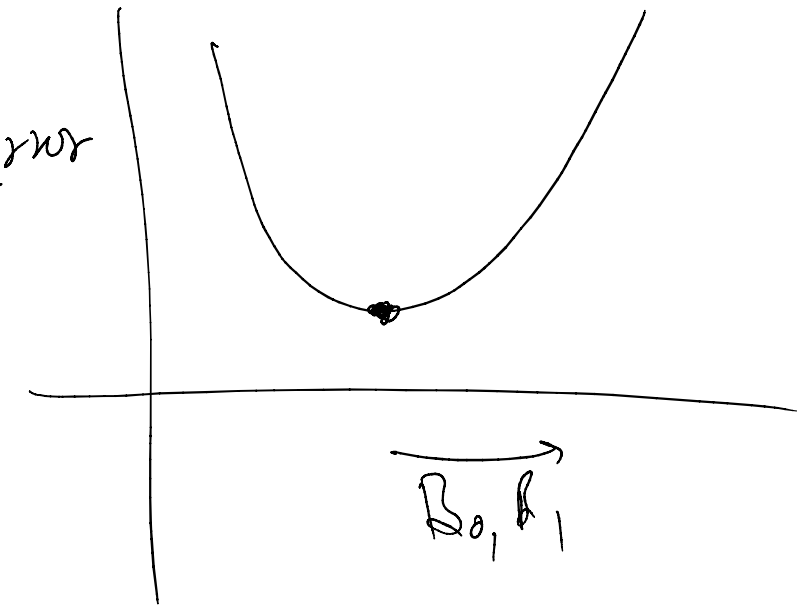
$$\hat{y}_i = \hat{B}_0 + \hat{B}_1 * X_i$$

Cost function → $\sum_{i=1}^{N} \frac{\left(y_i - (\hat{B}_0 + \hat{B}_1 x_i)\right)^2}{N}$

→ minimize cost function for best-fit

$\frac{\partial}{\partial \hat{B}_0} (f(n) = 0$

$\frac{\partial}{\partial \hat{B}_1} (f(n) = 0$



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

→ (i) Coefficient

$$\hat{B}_0 = \bar{y} - \hat{B}_1 \bar{x}$$

→ (ii) → Intercept

(f) x →

(t) y →

fitting → $\hat{B}_0$ $\hat{B}_1$ model: $\hat{y} = \hat{B}_0 + \hat{B}_1 x$

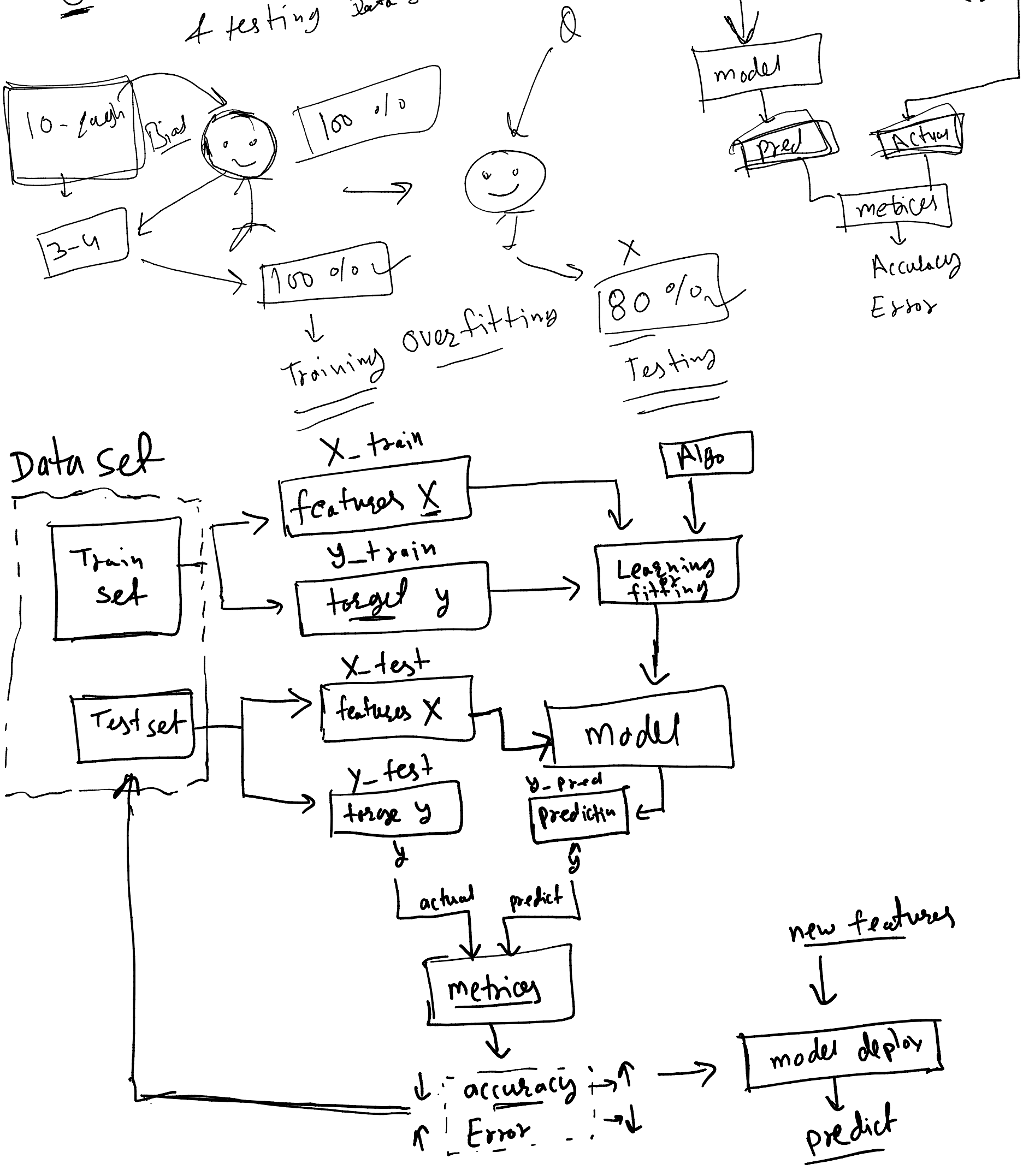
x → model → $\hat{y}$

→ How to proceed for ML project

① Load your Data & EDA

② feature selection

③ Split your Data into training & testing Data set



learn your dataset

① Load, clean, Explore nd select features

② Split data into training and testing data s

③ select best algorithm for your data, m

④ Train your model / fitting (find used D

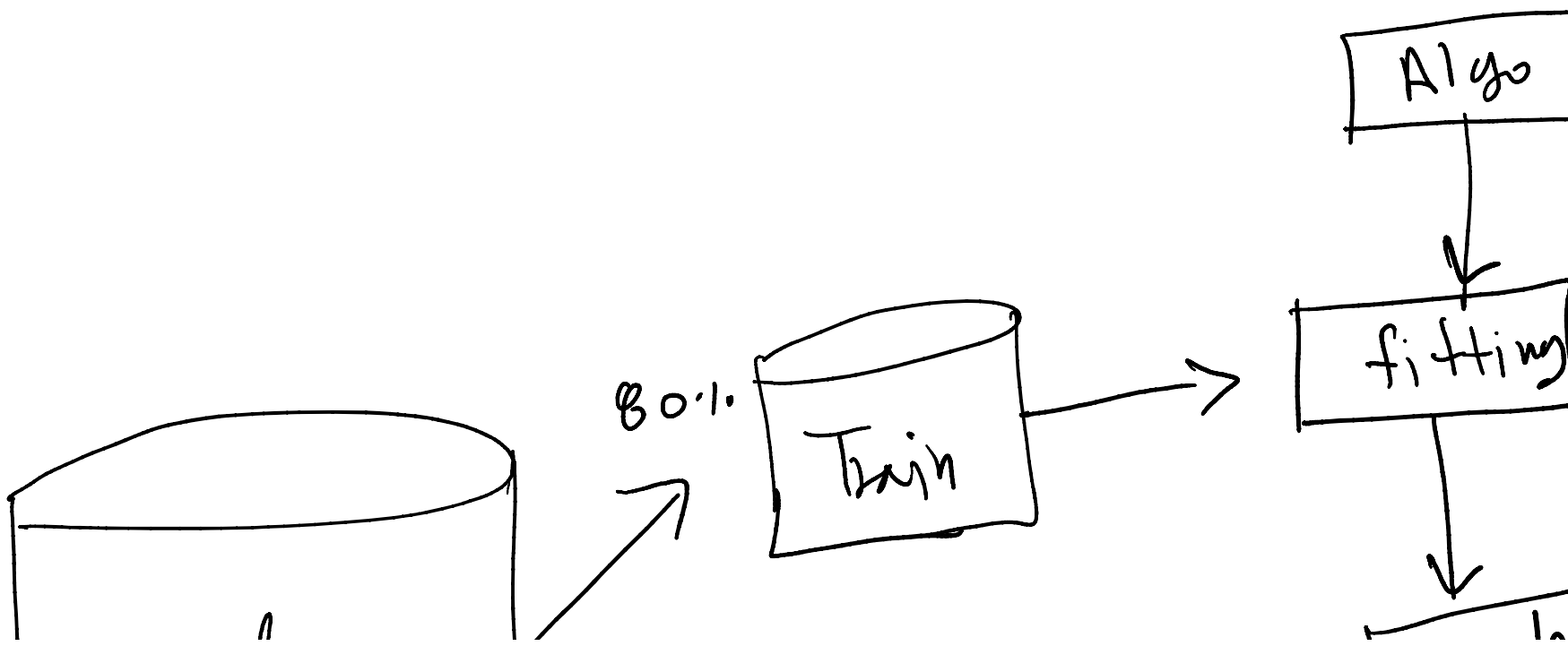
⑤ Evaluation of model

Accuracy & Error

⑥ model deployment / product creation

⑦ Optimization

⑧ Monitioring / maintince

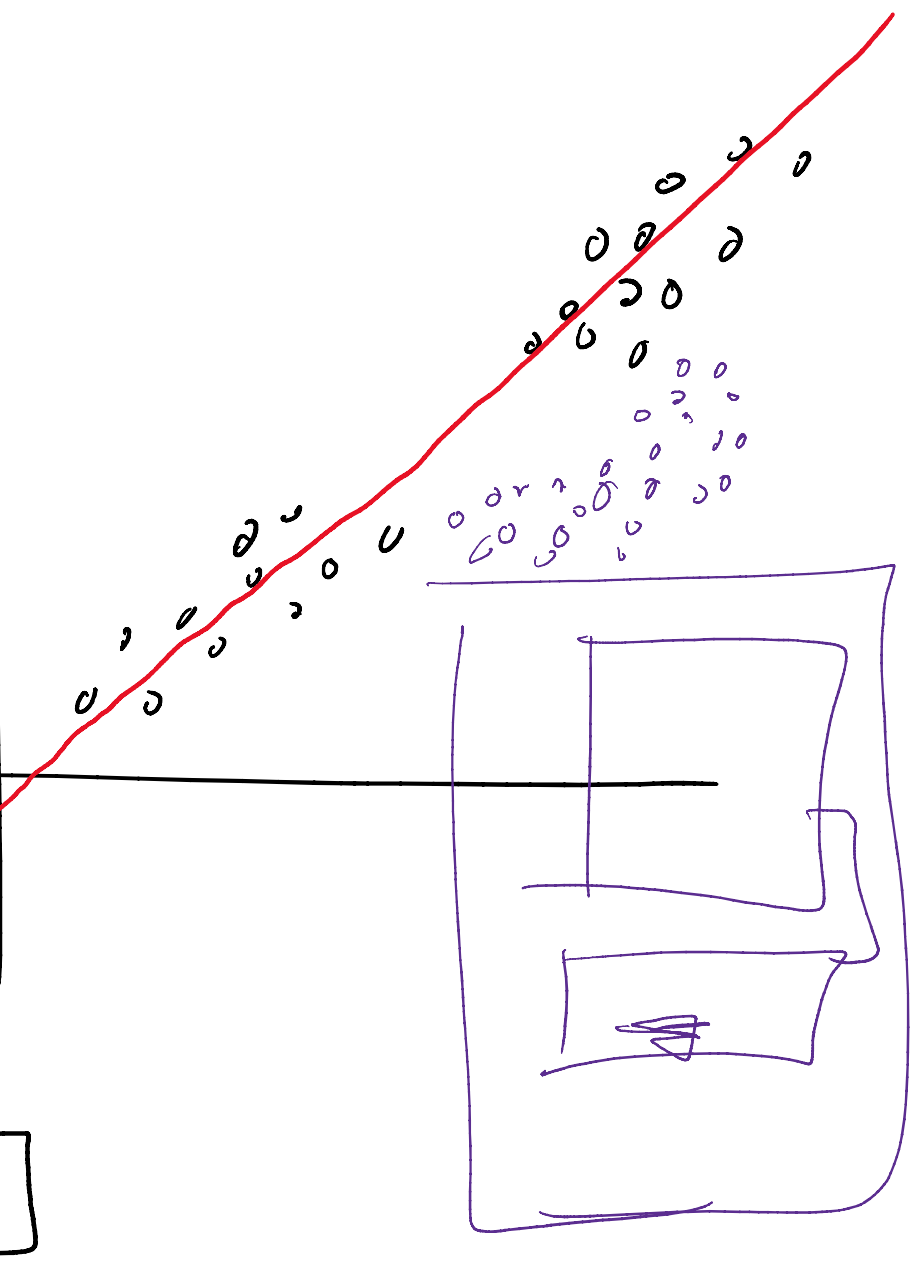


from your dataset

ets (for Evaluation purpose)

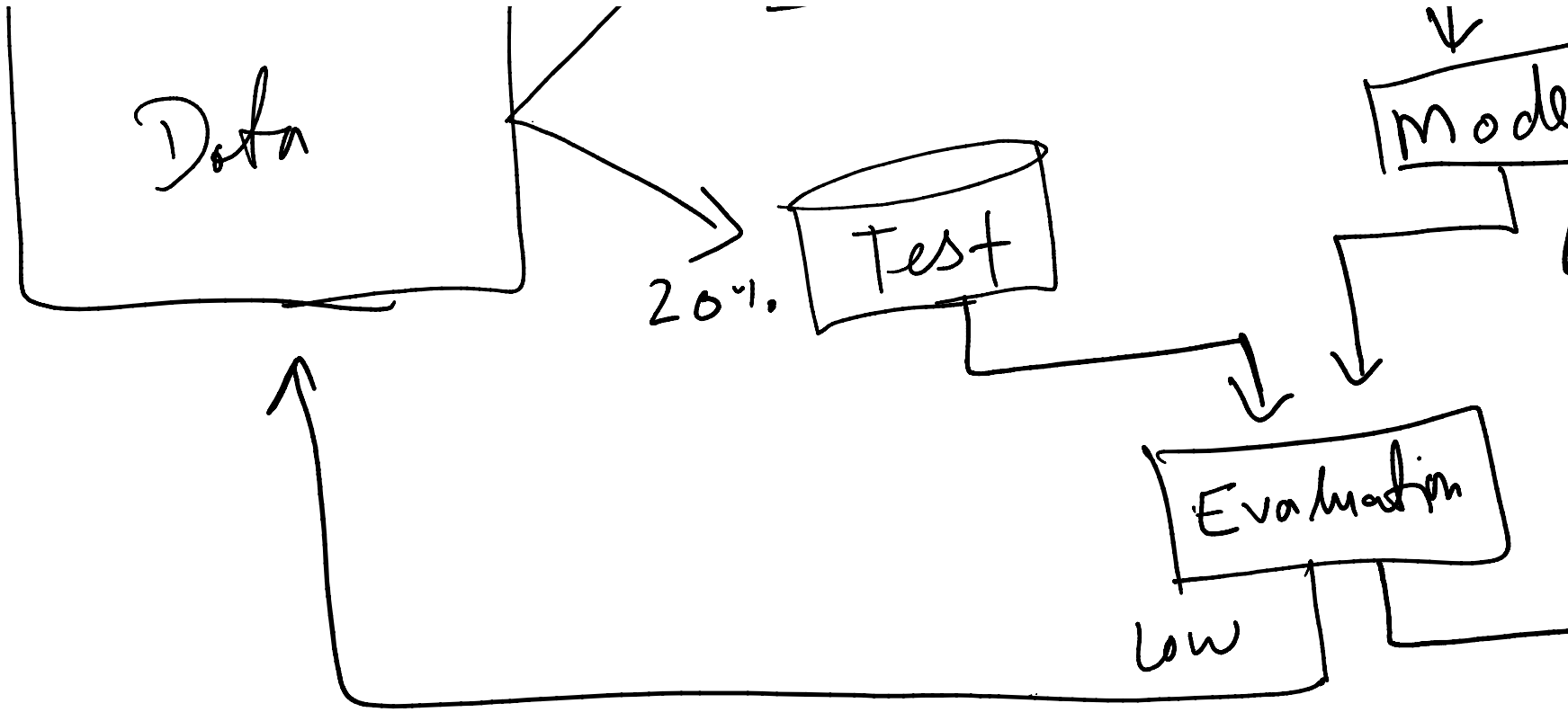
odel selection

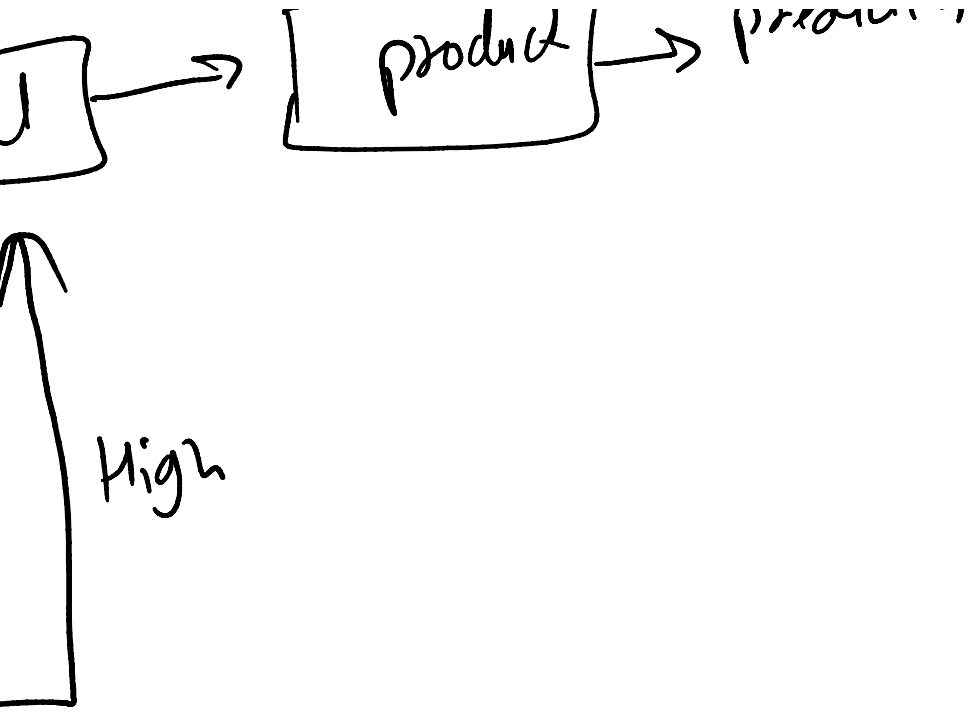
value of parameters
in hypothesis using Training
ata)



new Data

product → prediction


Data
20%
Test
Mode
Evaluation
Low


product
High