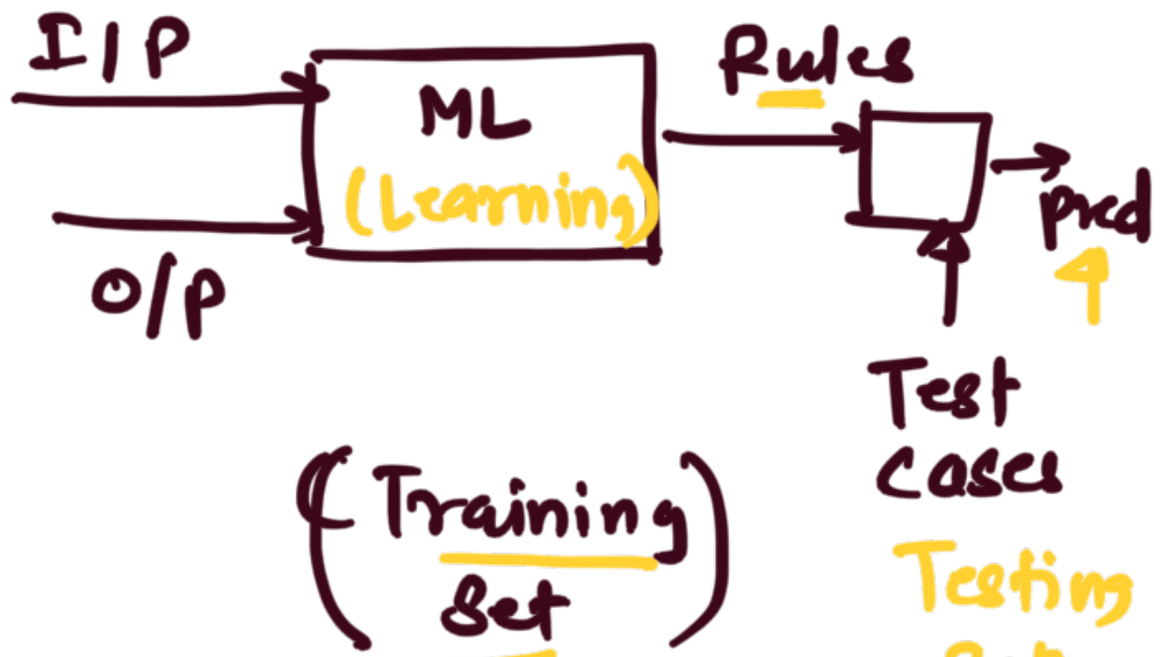
Classification



Classification → Training Set

Type of Records → I/P

Rules

Parametics

Different Algorithms

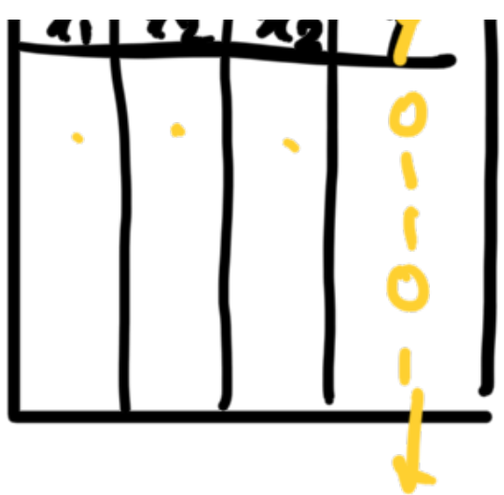
- Non-parametric
- Categorical
- Numerical

Set above are classification (Rules)

- Class A =
- Class B =
- Class C =

Classification → collection of record, (Training set)

attributes



attributes

↓

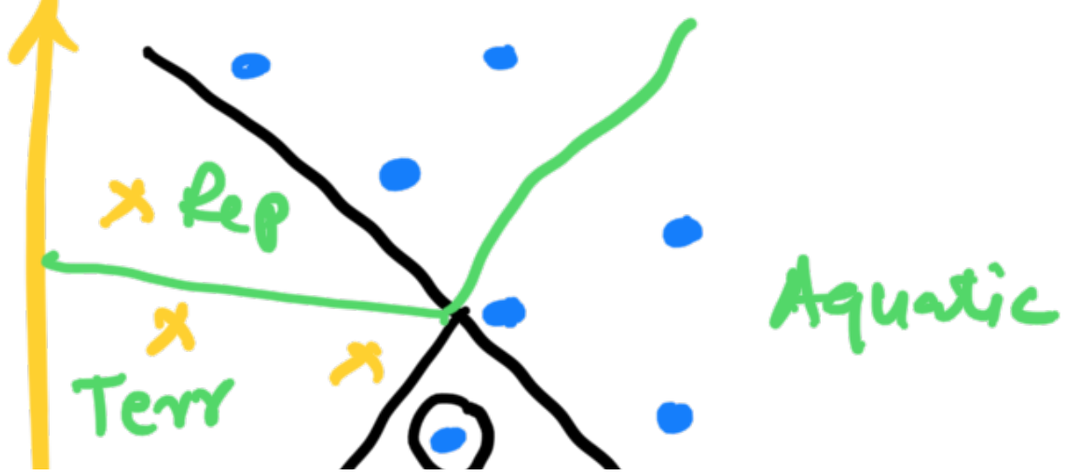
class

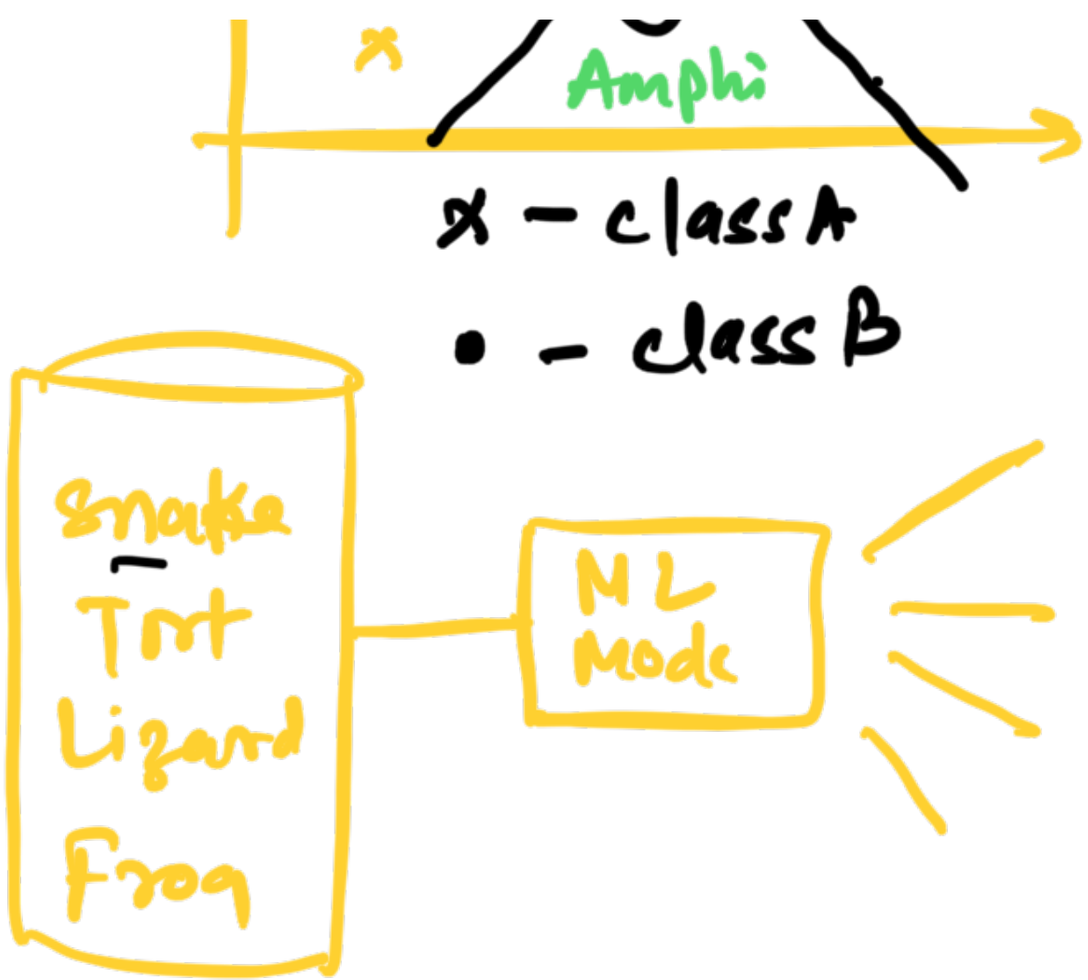
(Dependent Variable)

class

$y < \begin{matrix} 0 \\ 1 \end{matrix}$ → Binary classifier

$y = \begin{matrix} \text{Reptiles} \\ \text{Amph} \\ \text{Aquatic} \\ \text{Terrest} \end{matrix}$ → Multi classifier (>2 class)





ML paradigm (Training set)

1. Supervised Learning
   → I/P & O/P — categorical
      - Classification
      - Regression
2. Unsupervised Learning — continuous
   → I/P → Identify patterns

→ Clustering → grouping

→ Association → occurance

3. Reinforcement Learning

→ Learning Control

Task	Measures
1. Classification	Error
2. Regression	Error (MSE, R^2), Regularization
3. Clustering	scatter, purity

4. Association ⟶ support / confidence

5. Reinforcement Learning ⟶ cost/reward

→ Best Model ? → Model selection

Imp
- data
- how much data
- Accurate
- Training set

I/P + o/p [Classification]

Attributes

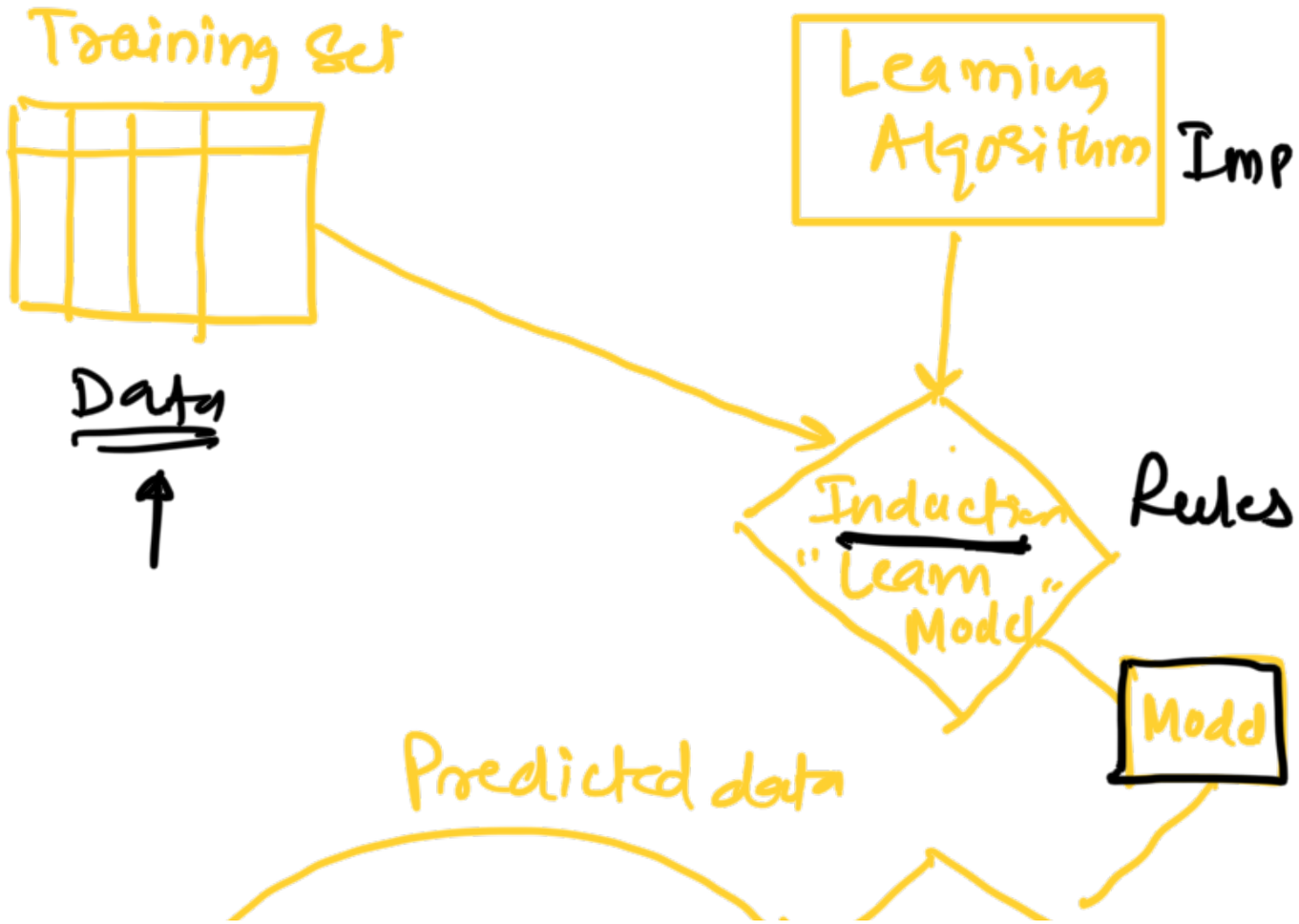
$\{x_1, x_2, x_3, \cdots x_n\}$

Class label

$\{y\} \leftarrow \begin{matrix} 0 \\ 1 \end{matrix}$

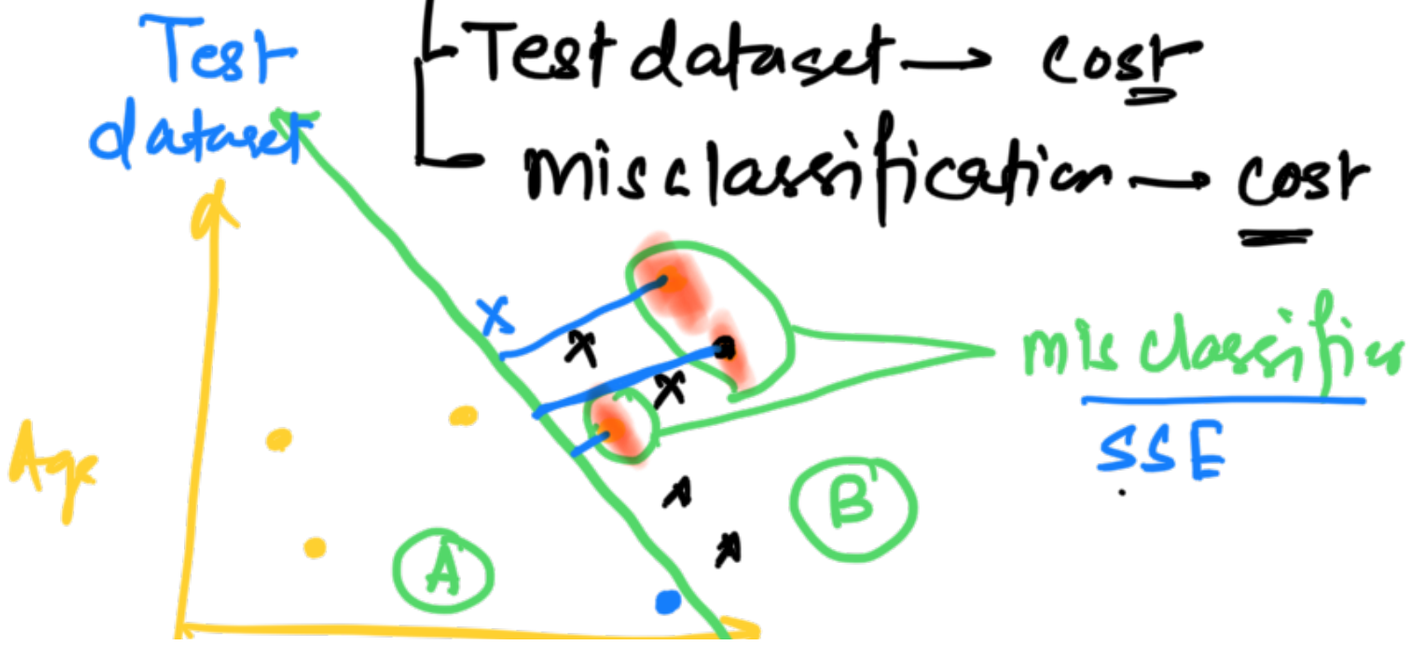
→ Discr, Text, Image, Sound, Video

General Approach for Classification Model



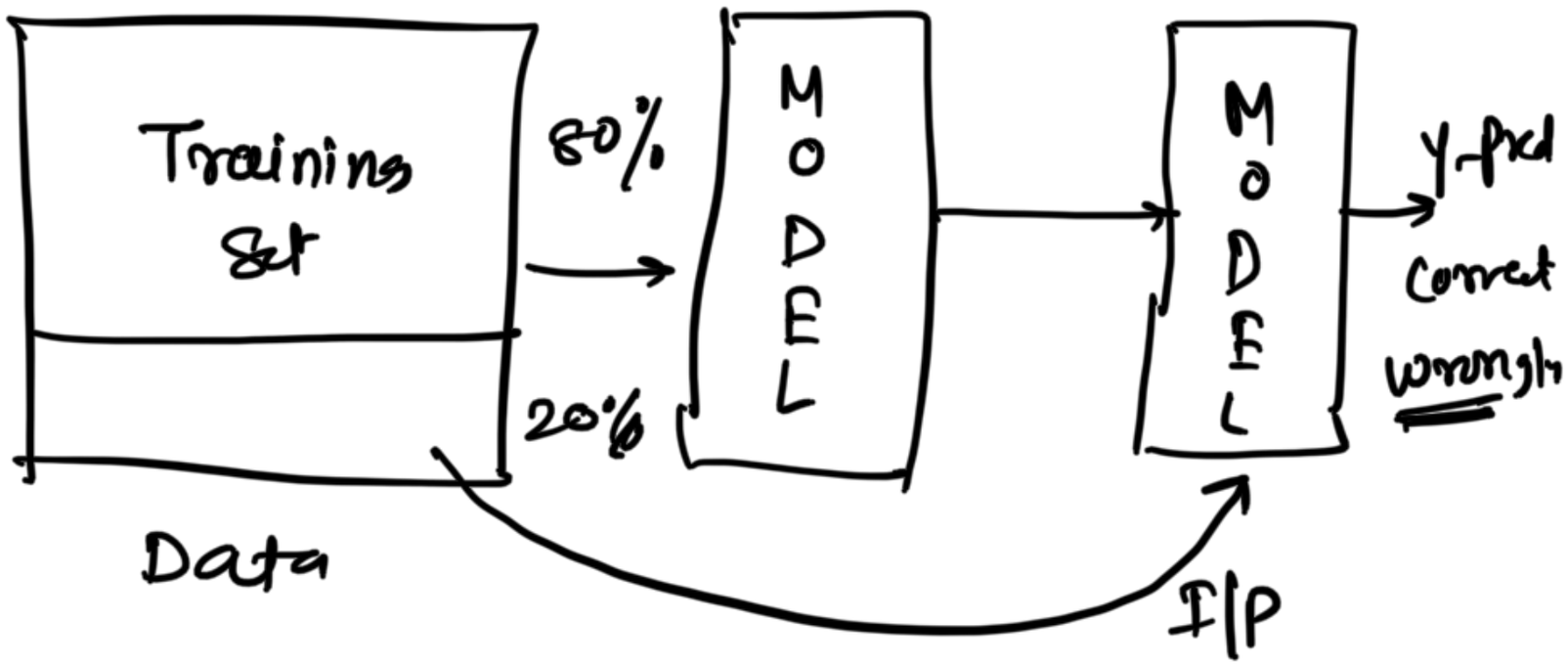
Evaluate classifer: outcomes

- Accuracy
- Confusion matrix
- Cost-sensitive data analysis
  - Test dataset → cost
  - misclassification → cost

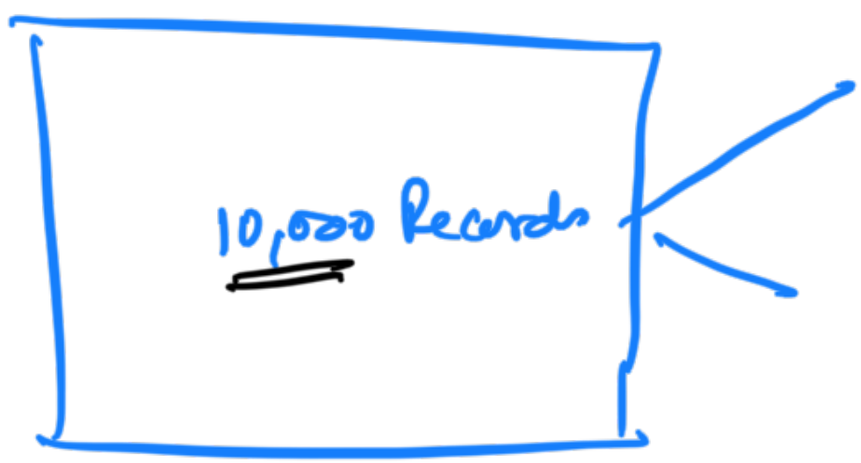


Salary

Decision boundary



Person < Happy → 98%
Sad → 2%



95000 → Training

1,2
500 → Testing
Sad

99.9%

Accuracy

→ Confusion Matrix

Unbalanced dataset

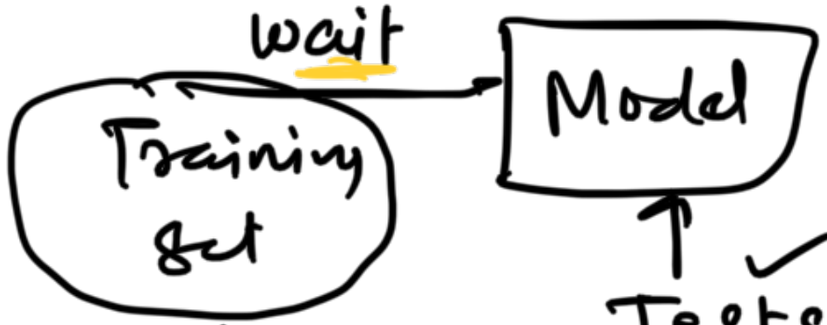
Type I Type II

Errors

Balance Dataset →
- 5000 → Happy
- 5000 → Sad

Learners → Lazy Learners
→ Eager Learners

Lazy Learners → eg: KNN
Case base learning



DT, RF, ANN, Naive Bayes

pd.read_csv(data)

train_test_split (80-20)

Training set → Model ← y-pred

Case → Lazy learner

Age	Salary	Computer Y/N
		→ 80%

Test

20	80,000	? ✓

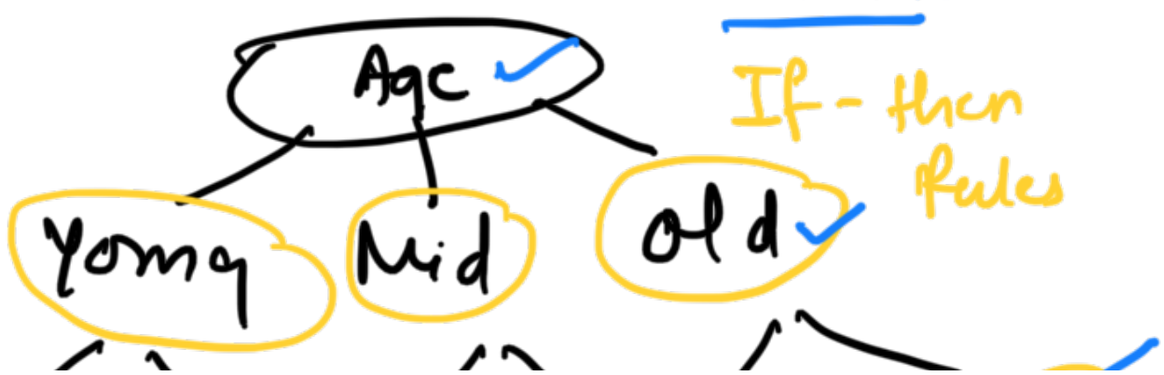
User

wait $d(x_1, y_1)^2$ → formula

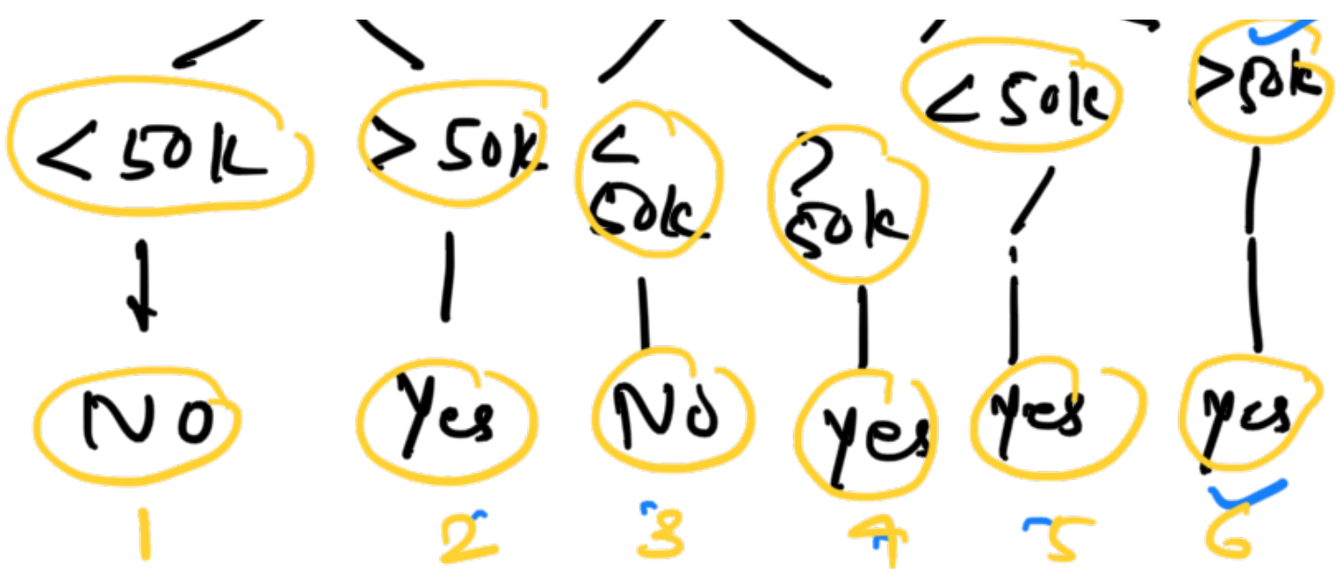
Eager learner

Case →

Decision Tree



If - then Rules



Age = 52, Sal 3Lac → yes

---

classifier → kNN

$$d(x,y) = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$$

Test case:-

John → 37, → 50k → 2 ∋ ?

yes/No

---

1. $\sqrt{(35-37)^2+(35-50)^2+(3-2)^2}$

$= \sqrt{2^2+15^2+1^2} = \sqrt{230}$

$= 15.16$ — d_1

2. $\sqrt{(22-37)^2+(50-50)^2+(2-2)^2}$

$= \sqrt{225} = 15$ — d_2

3. $\sqrt{(65-37)^2+(200-50)^2+(1-2)^2}$

$= 152.65$ — d_3

4. $\sqrt{(59-37)^2+(170-50)^2+(1-2)^2}$

$= 122.20$ — d_4

5. $\sqrt{(25-37)^2+(80-50)^2+(4-2)^2}$

= (15.74) — ds

1) 15.16 ⟶ class → No
2) (15) ⟶ class → yes
5) 15.74 ⟶ class → ↑ yes

K=1 John, 37, 50K, 2, yes

K=2 yes/No → yes

K=3 ⟶ Yes

K=4 No → yes

K=5

---

y
0/1
class

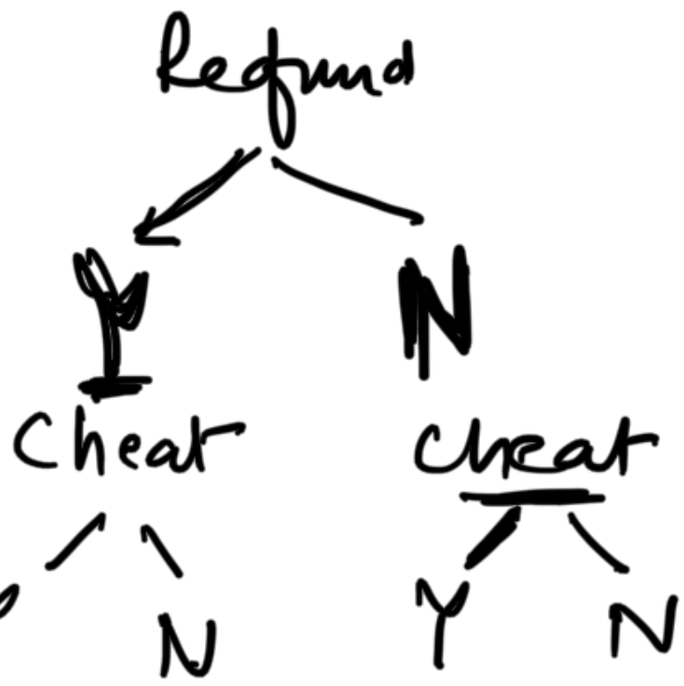
Refund (value in Attribute)

Yes No

Y N

Yes

No



0 3 3 4 ⇒ 10

Class \ Refund	Y	N
Y	0	3
N	3	4

= 7

	Y	N.
	0/3	3/3
	3/7	4/7

Prob

Probability Algorithm

$$d_1(v_1, v_2) = \leq \begin{vmatrix} n_1 & n_2 \\ n_3 & n_4 \end{vmatrix}$$

KNN ⇒ min. distance parameter

↳ class label for the testcase