Ensemble Learning ← ML

↑

Collection of many thing

Ensemble Learning → group of models
group of algos
Combining many models

↓

individual model

↓

wisdom

↓

final

→ **Wisdom of crowd**

= Audience Poll

KBC ⇒ Life line

Ques. → Life line

↑ {whole Population of hall}

↑

– majority

A	$\boxed{B}$	C	D
↑	↑	↑	↑
20	50	10	30

KBC → Candite
individual → 20

↑ Correct ans

99%

wisdom of crowd

. Poll ⇒ 20 50 {80 100}

20	50	80	100
↑	↑	↑	↑
5	30	40	25

100 ∴ 65

Democarcy =
Wisdom of the crowd
majority vote

flipcart ⇒ review of the Product

– (Nike ⇒ 5K → {2 renew})

– Puma ⇒ 5K ⇒ 510 renew

↓

4.5

{decision making will easy} ← {wisdom of crowd}

Core idea ⇒

cgpa	iq	Placement

Classification

cgpa	iq	LPA

Regression

Difference ⇒

M_1 . M_2 . M_3 . M_i . . . . M_n

Same model → Homogenous — LR, LR, LR, LR, LR

diff grp → Hetrogenous — LR ←→ SVR ←→ Rd reg ←→ DT ←→ KNR.

generalise

Manish → KBC ⇒ Hotseat ⇒ Struck

field → Audience Poll

less Robust

mixed

Anysonr ↻ Avg

generalise

Datascience → (Data Scientist) ← generalise

CA

SE

Politician

CEO

variety ⇒ 3rs

2nd type

Robust

Generalise

homo/hetro → M_1 - - - M_2 - - - M_3 - - M_4 - - - - M_n

0 1 0 0 0

Classification ⇒ Majority Count ⇒ 0 ← More count

Placement

final answer

Reg ⇒

LPA

3.4 5.1 6.3 8.5 . . . . 10.6

Reg ⇒ Avg ⇒ mean

Type of ensemble technique

Ensemble
- Voting
  - stacking
- Bagging → Boosting
  - {Random forest}
- Boosting
  - Ada Boosting
  - Gradient
  - XGBoost
  - Cat Boosting
- Stacking

① Voting ⇒ heterogenous (Diff technique) model
- classification → Majority voting
- Regression → avgerage

Data
- SVM (w_1) (w_0) → 1
- LOR (w_2) (w_0) → 0
- DTC (w_3) (w_0) → 1

→ 1

Base model

⇒ Mode

1 → 2
0 → 1

1 > 0

hard classifier

→ {Soft classifier}

$y = m_1x_1 + m_2x_2 + c$

weight

$= m = \frac{y}{x}$

$m = \frac{y}{x}$

$y\uparrow$ $x\uparrow$

5 5

1 unit

1

Simple

Good Bad tea

Tea
- Water — $\frac{1}{2}$ w_0 ($\frac{1}{2}$)
- Milk (cola) — $\frac{1}{2}$ w_0 (1)
- tea leaf — $\frac{1}{2}$ w_0 (300g) 100g
- Sugar — $\frac{1}{2}$ w_0
- Ginger — $\frac{1}{2}$ w_0 (10g)
- elchi — $\frac{1}{2}$ w_0 (2g)

Stacking :-

Data

M1 SVM — 1

M2 LoR — 0

M3 DT — 1

(Prediction)

Meta data of the model

(Data) ← M4 (KNN)

SVM o/p	LoR o/p	DT R	target
1	0	1	
0	0	1	
⋮			

Stacking
↓
Stack of the model

Meta → Pred.

{Voting (stacking | Bagging | Boosting)}

① training

② Predicting

iq	cgpa	placement
130	8.5	1
125	8.0	0
135	9.0	1
110	8.1	0
160	9.7	1

M_1 SVM	M_2 LoR	M_3 DT
0	1	1
0	0	1
1	1	0
0	1	0
1	1	0

↓

M4 ⇒ KNN

metamodel

③ Bagging ⇒ Bootstrap + Aggregation

↓

Random forest: Robust, generalize

BAGGing

Data

B B B

Sampling

M1 M2 M3

Homogenous

⇒ SVM SVM SVM

Random forest ← DT DT DT

RF

{Bootstrap

↓

Diff way of a sampling

① Row Sampling

② Column

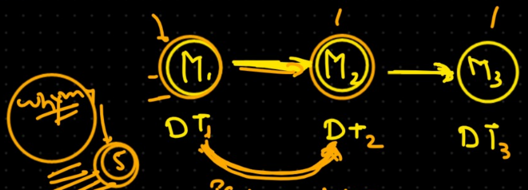
③ Combined

Sampling → ① with Replacement

{majority voting / average value}

② sampling without Replacement

4 **Boosting** (homo) => Sequential method



Boosting
↳ model

Voting ⇒ hetro
↓
Stacking ⇒ hetro

homo

Bagging ⇒ Bootstrap + Agg.
(sampling) (MC) (Avg)

(Parallel)

Voting classifier

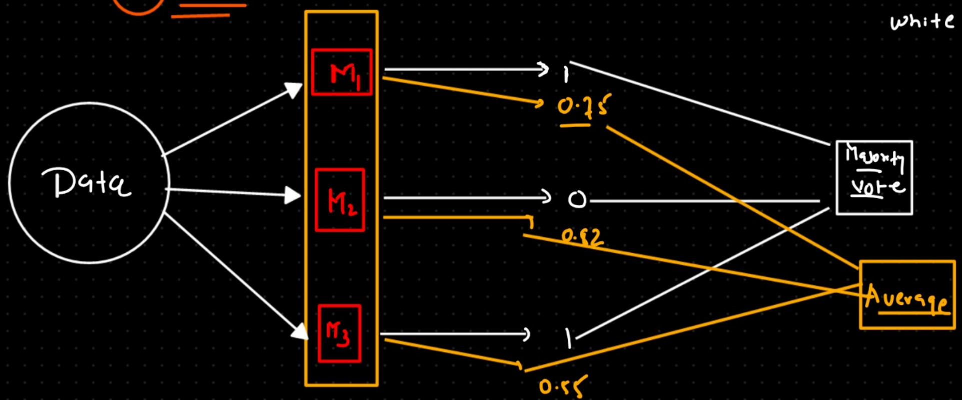
① Core Idea

② Votting classifier

③ Votting Regressor

Voting ① train ② test

Yellow ⇒ Reg.
white ⇒ class.



Model Number

M1 ... Mn ⇒ odd number

{Don't take even no.}

Hard voting (Soft voting)
↓ ↓
Mode ⇒ highest freq (Majority Counting)   Probability

$\rightarrow$ M1 — M2 — M3

0.7/0.3 $\rightarrow$ 1 | 0.7/0.3 $\rightarrow$ 1 | 0.7/0.3 $\rightarrow$ 1

- Count
- Prob — soft voting

- 1) training
- 2) testing

- $Prob_Y = \frac{2}{3} \Rightarrow 1 - \frac{1}{3} \Rightarrow \frac{3-2}{3} = \frac{1}{3}$

2 Yes
1 NO

$Prob_N = \frac{1}{3}$

1/0 $\rightarrow$ acc/error

error = 1 - acc

70%

error = 30%

70%

30%

Prob

per/comb

M1 $\rightarrow$ 0.7 $\rightarrow$ M2 $\rightarrow$ 0.7 $\rightarrow$ M3 $\rightarrow$ 0.7 $\Rightarrow$ yes $0.7 \times 0.7 \times 0.7 = 0.343$

0.3 $\Rightarrow$ $0.7 \times 0.7 \times 0.3 = 0.147$

M2 $\rightarrow$ 0.3 $\rightarrow$ M3 $\rightarrow$ 0.7 $\Rightarrow$ $0.7 \times 0.3 \times 0.7 = 0.147$

0.3 $\Rightarrow$ $0.7 \times 0.3 \times 0.3 = 0.063$

M1 $\rightarrow$ 0.3 $\rightarrow$ M2 $\rightarrow$ 0.7 $\rightarrow$ M3 $\rightarrow$ 0.7 $\Rightarrow$ $0.3 \times 0.7 \times 0.7 = 0.343$

0.3 $\Rightarrow$ $0.3 \times 0.7 \times 0.3 = 0.063$

0.3 $\rightarrow$ M3 $\rightarrow$ 0.7 $\Rightarrow$ $0.3 \times 0.3 \times 0.7 = 0.063$

0.3 $\Rightarrow$ $0.3 \times 0.3 \times 0.3 = 0.027$

0.027

1

M1 $\rightarrow$ 80%
M2 $\rightarrow$ 55
M3 $\rightarrow$ 77% } $\Rightarrow$ voting

classification — logreg

M1: 0 $\rightarrow$ 0.6, 1 $\rightarrow$ 0.4 $\rightarrow$ $0 = 0.6 \times 0.8$

M2: 0 $\rightarrow$ 0.8, 1 $\rightarrow$ 0.2 $\rightarrow$ 0.4×0.2

hard $\rightarrow$ mode

Prob = -

M1: 0.6 $\rightarrow$ 0, 0.4 $\rightarrow$ 1

M2: 0.8 $\rightarrow$ 0, 0.2 $\rightarrow$ 1

M3: 0.4 $\rightarrow$ 0, 0.6 $\rightarrow$ 1

M1 $\rightarrow$ 0
M2 $\rightarrow$ 0 ? 0 hard

M3: 0 0.4, 1 0.6

M1 = 0
M2 = 0 $\rightarrow$ 0
M3 = 1

Soft

Prob

0.6, 0.8, 0.4 — 1.8 $\Rightarrow$ 0.4, 0.2, 0.6 — 1.2

$0.6 \times 0.8 \times 0.4$

0.192

$0.4 \times 0.2 \times 0.6$

0.048

0.192 + 0.048 = 0.240

0.24

```
for estimator in estimators:
    x=cross_val_score(estimator[1],X,y,cv=10,scoring='accuracy')
    print(estimator[0],np.round(np.mean(x),2))
```

```
estimators = [('LOG',clf1),('DT',clf2),('KNN',clf3)]
```

LOG DT KNN

100

10 → 100/10   10   mean

(1) acc (2) acc (3) acc -- (10) acc

1. cross_val_score (log, x, y, cv=10, accuracy)
2. cross_val_score (DT, x, y, cv=10, accuracy)
3. cross_val_score (KNN, x, y, cv=10, accuracy) ⇒ cv=10 ((acc, acc, acc - - acc))

cv 10 ⇒ 10 10 mean (acc, acc, acc - - acc)

mean

Question Benefit ⇒

1. Bias variance
2. Robust output
3. improvement in the performance

When to use ⇒

1. Competition (ensemble technique)
2. good acc
3. org. / company ⇒ Ensemble

right

huge → ANN

Powerful

ANN ↑ ↑

less/intermediate →

ensemble → best

Bias

(error at the time of training)

(at the time of pattern reading of the data)

Variance

(at the time of the pred)

(test data how much varying from my predicted data)

Bias ↓ variance ↓

Underfitting — Bias ↑ variance ↓ — Linear

Overfitting — Bias ↓ Varianc ↑ — Decision tree

⇒ market Algo

Generalised Model

ensemble Learning ⇒

- → Bias, variance will be near to each other
- → much diff in bias and var
- → bias ↓ var ↓
- → bias & var almost equal to each other