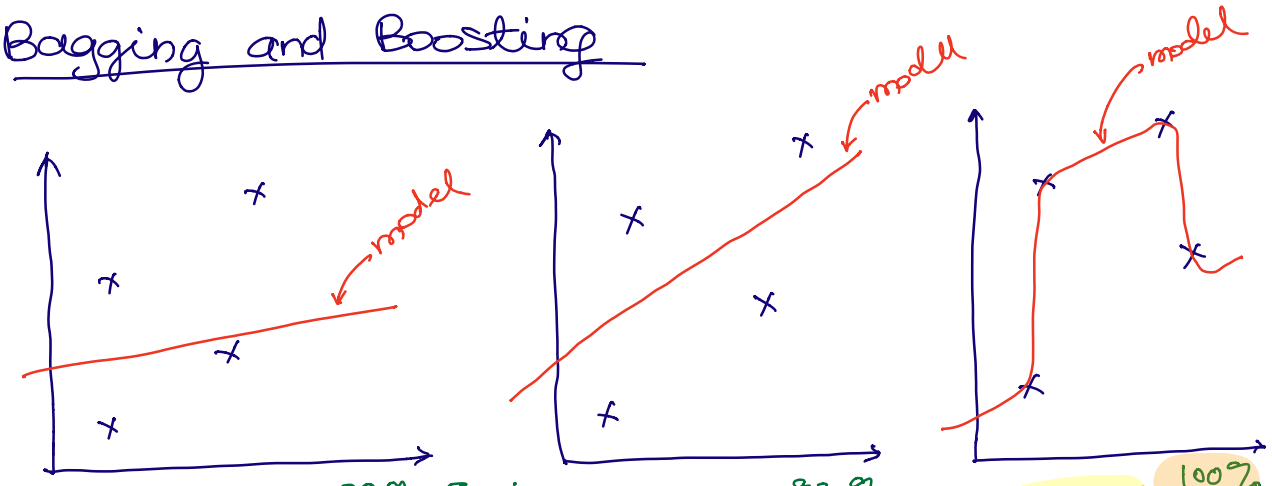
Bagging and Boosting



Model 1 — 29% - Train, 30% - Test

Underperforming

→ High Bias

→ Underfitted model

Model 2 — 82%, 90%

generalized

Good score

Model 3 — 100%, 98%

Well performing

But not generalized

→ High variance

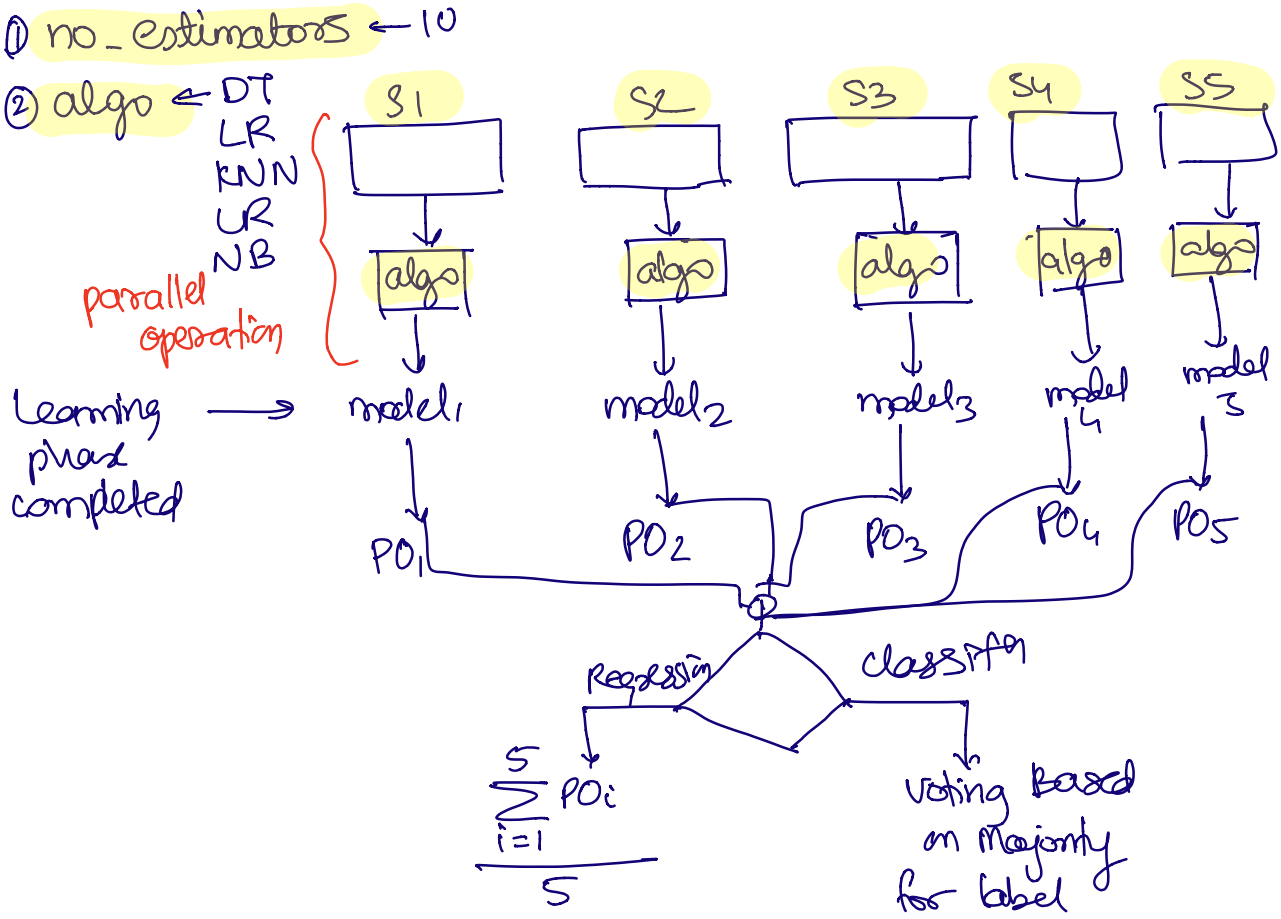
→ Overfitted model

$$Bias \propto \frac{1}{Variance}$$ ... observed mathematics

→ To solve underfitting issue, you can use BOOSTING.

→ Bagging to remove overfitting

Bagging → Bootstrap Aggregation



$$\frac{\sum_{i=1}^{5} PO_i}{5}$$

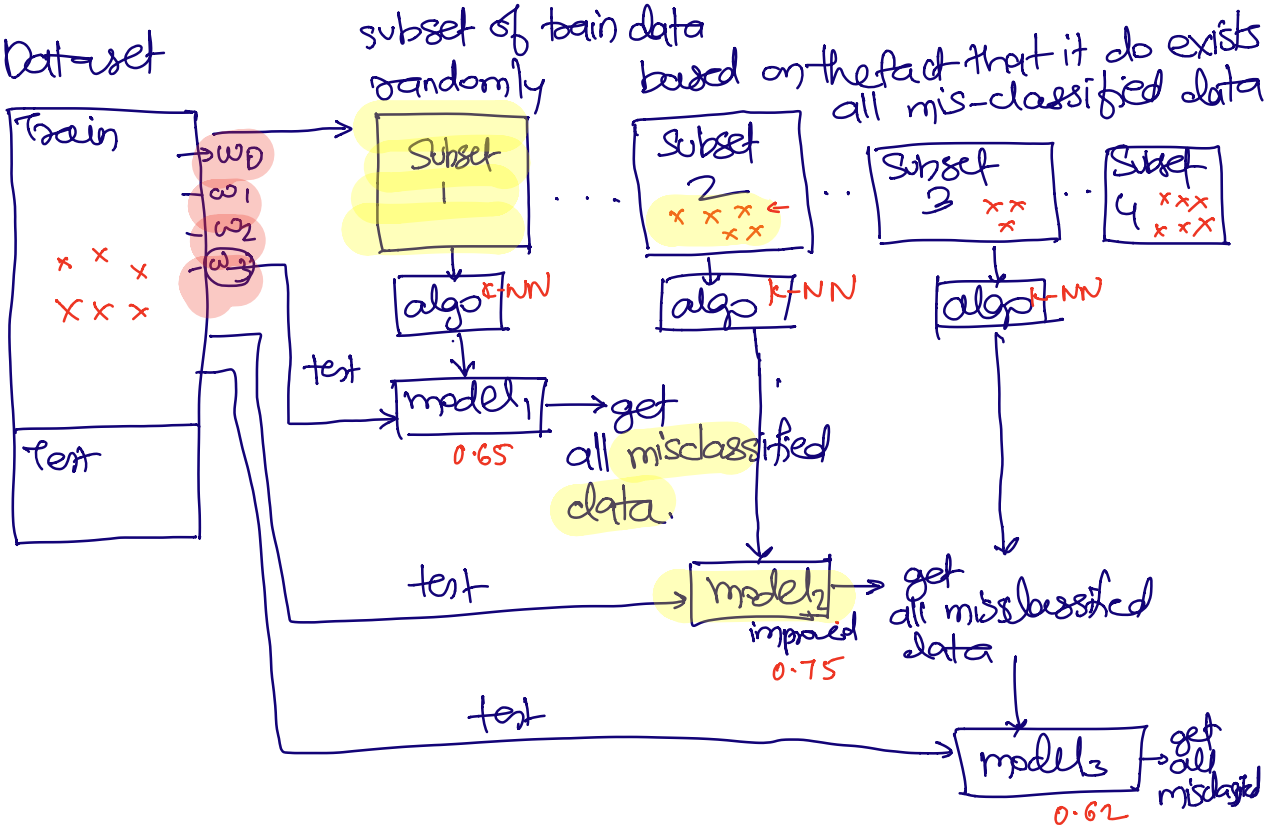
Boosting: (ideally used in Neural networks)

→ helps you to improve the performance of the model in terms of score.

→ Ideally, boosting is used only for classification use-cases, however we do have an option to use it in Regression

n_estimators
4

algo
K-NN

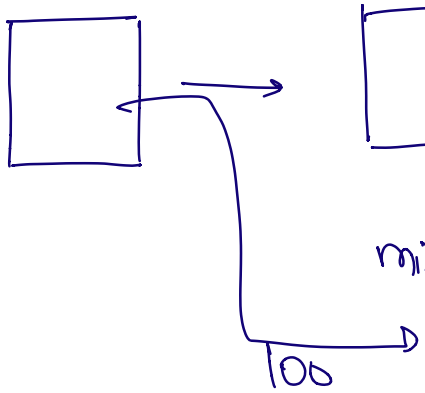


n_estimators : 5

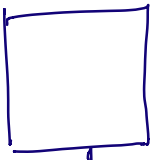
model2 is selected for prediction

How much data does each subset takes?

n=100

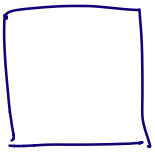


iteration1 (all 100)



misclassified

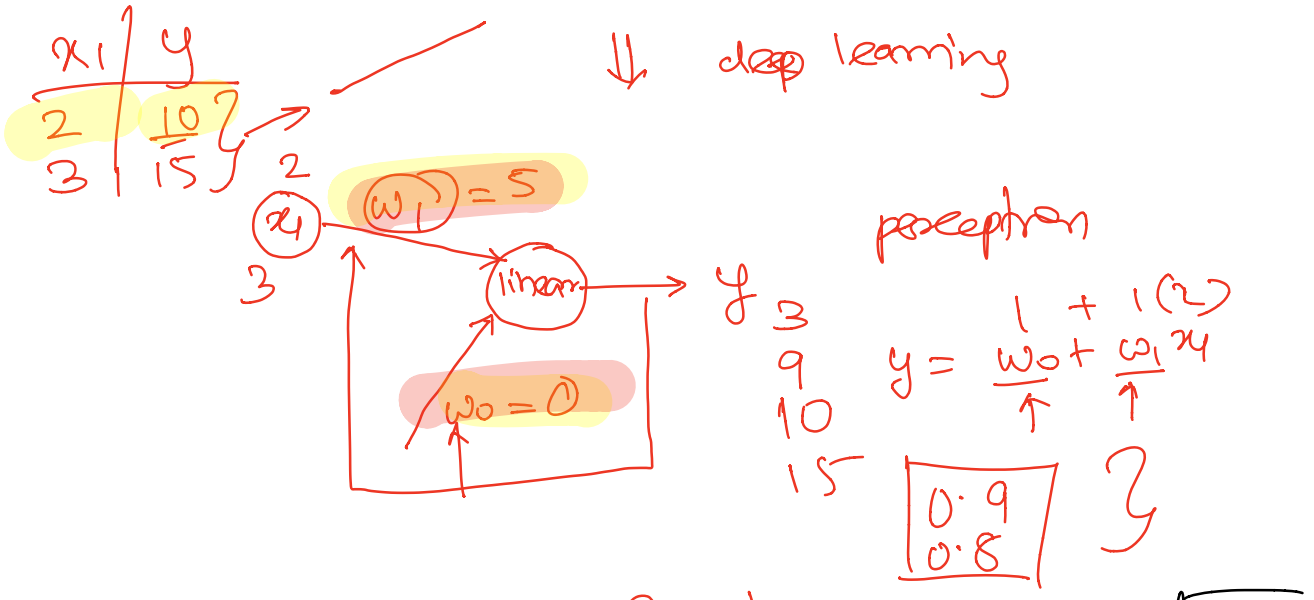
iteration2 (<100) ensuring all misclassified data is present

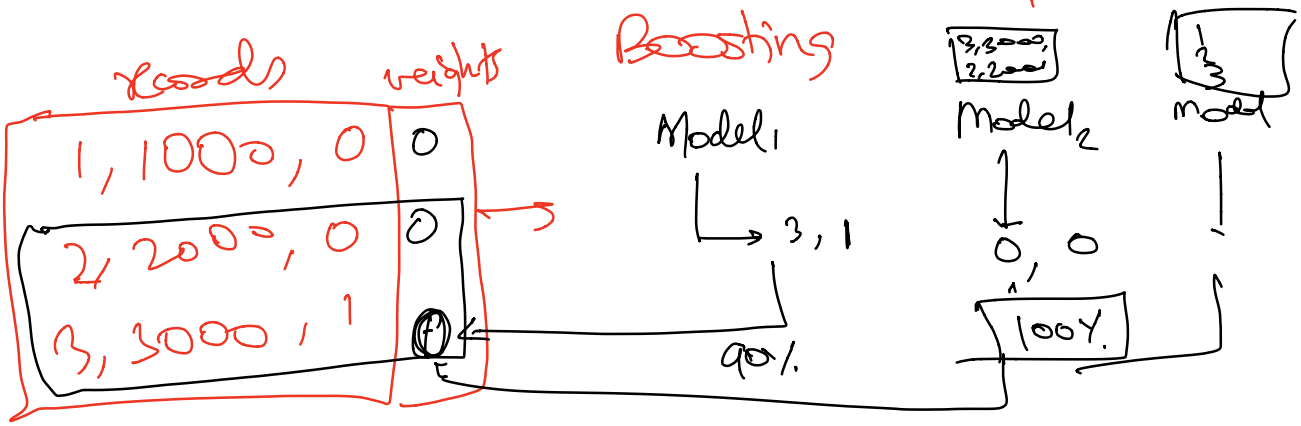


Based on weights of each record

$y = b_0 + b_1 x_1$ ⟶ Simple linear regression

1 feature
1 label



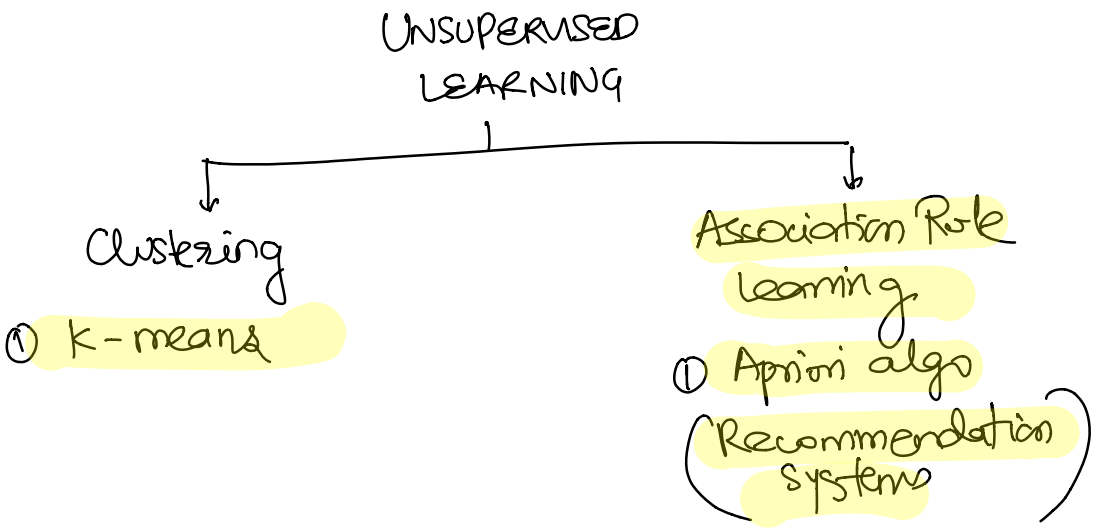


n_iteration = 10

Unsupervised Learning

features but no label

finding some hidden patterns or associations.

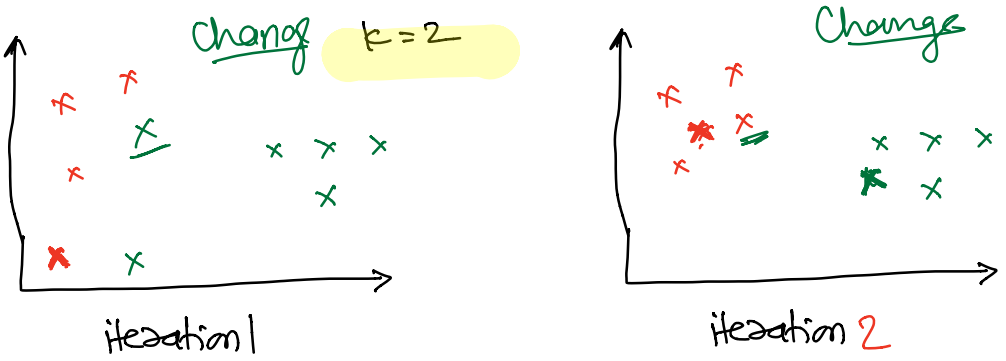


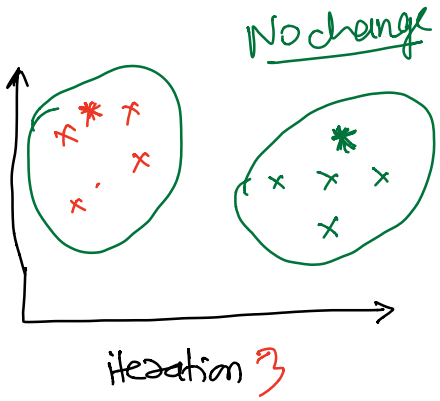
K-means → identifying and discovering k-clusters from the data

Two techniques to figure out the ideal value of k.

① Use EDA and checkout for clusters (pairplot)

② Use elbow technique (K-means)





STOP the ITERATION

gender	R&D spend	Profit
m	10000	100000
m	50000	100000
f	30000	200000

m	f	R&D spend
1	0	

Profit → model

gender → model

R&D spend →

```
if (gender == 'm')
    feature = m(1, 0, R&D)
else
    feature = m(0, 1, R&D)
```