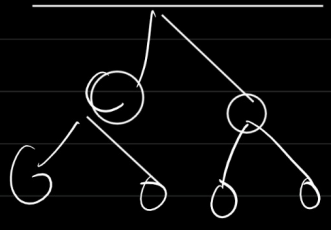
Till now :-

Decision Tree



→ Gini, Entropy.
→ Information gain

Agenda

- → Pickle
- → Prepruning & Post pruning.
- → Practical implementation DT Classifier.
- → DT Regressor
- → Pract - DT - Regressor.

* SVM
* Naive Bayes
* Ensemble

// * Decision tree post pruning and pre-pruning

DT → greedy algorithm

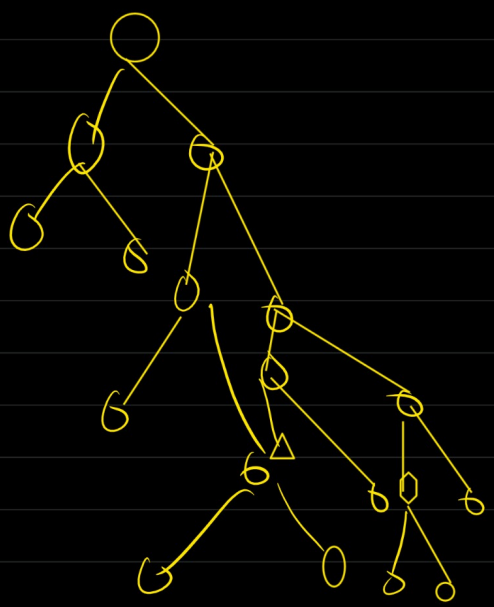
⇓

fully grown

⇓

Overfitting (memorization)

↓ train acc ↑
test acc ↓

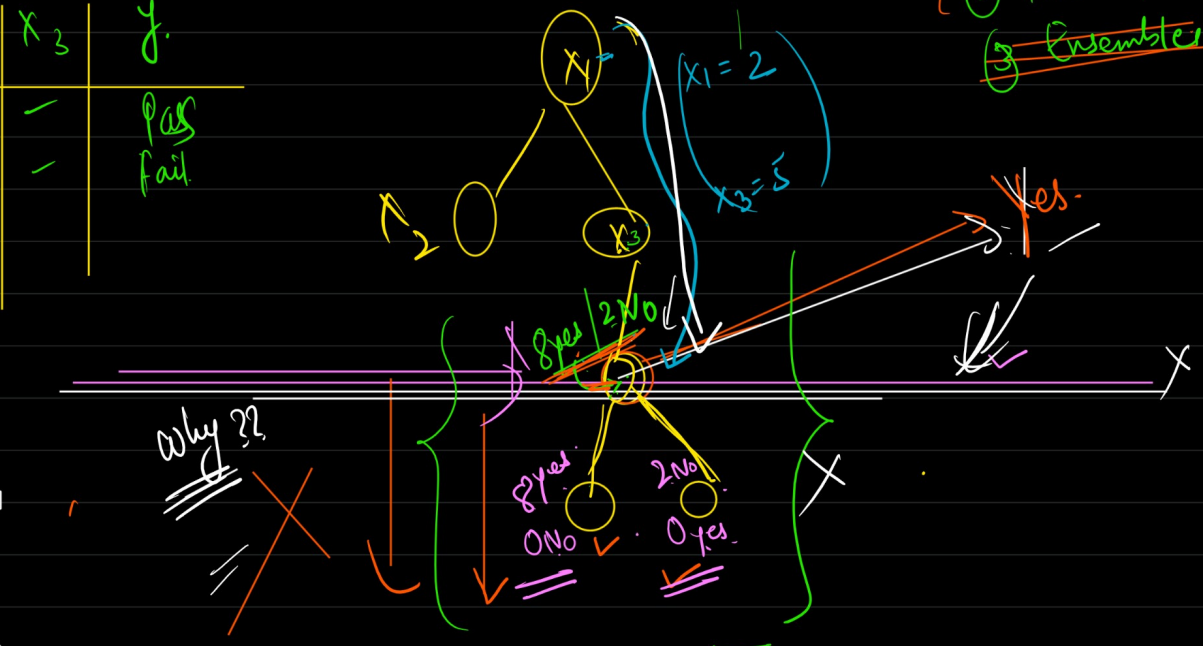


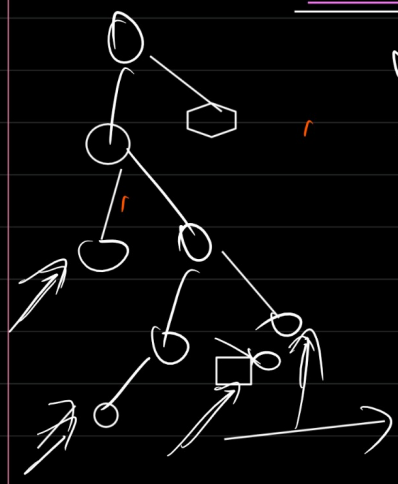
* Generalised model → low bias, low variance

train acc ↑ test acc ↑.

① Post Pruning
② Pre Pruning
③ Ensembles

X_1	X_2	X_3	y.
–	–	–	Pass
–	–	–	fail



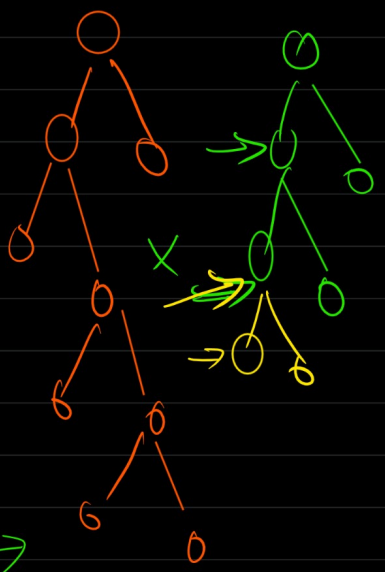


→ introduce Some bias

Parameter to create DT

→ Criteria — Gini / Entropy.

→ max_depth → maximum depth of tree.



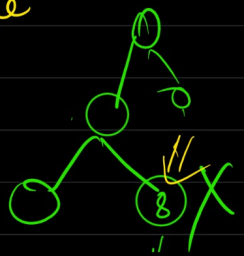
→ min_sample_leaf

⇓

The minimum no of Samples needed to be considered as a leaf node default is 1.

Visualisation

Hyperparameter tuning → using Cross validation



→ split tree

→ min_samples_split

→ The minimum no of sample a node most contain to consider split

* max feature

① Post pruning (cutting of DT after it is created)

↓

cutting

→ Construct the entire dt until leaf node

→ Prune the DT

→ Works well with small dataset.

*② Pre-pruning

→ Limit the growth of decision tree.

→ Hyperparameter tuning to select best parameter

* Decision Tree Regressor

DT classifier

- Entropy / Gini
- IG

Regression

$y \rightarrow$ continuous

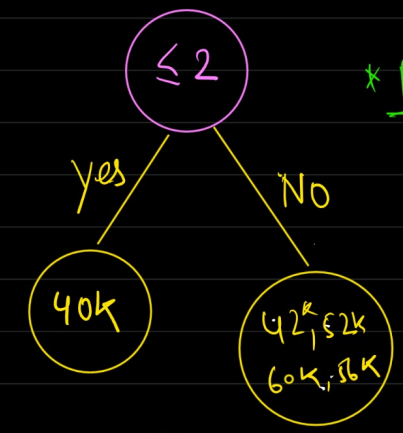
Variance

Variance reduction

Experience	Career Gap	Salary
2	Yes	40k
2.5	Yes	42k
3	No	52k
4	No	60k
4.5	Yes	56k

$\bar{y} = 50k$

[40k, 42k, 52k, 60k, 56k]



* Final aim
  - To reduce the variance
  - We want feature / value which give highest feature reduction

$$\text{variance} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 \quad \text{(mean squared error)}$$

$$\text{Var}_{(\text{root})} = \frac{1}{5}\sum_{i=1}^{n}\left((40-50)^2 + (42-50)^2 + (60-50)^2 + (56-50)^2\right)$$

$$= \frac{1}{5}(100 + 64 + 4 + 100 + 36)$$

$$= 60.8$$

$$\text{Var}(2.5) = \frac{1}{1}\left((40-50)^2\right)$$

$$= 100$$

$$\text{Var R.C} = \frac{1}{4}\left(8^2 + 2^2 + 10^2 + 6^2\right)$$

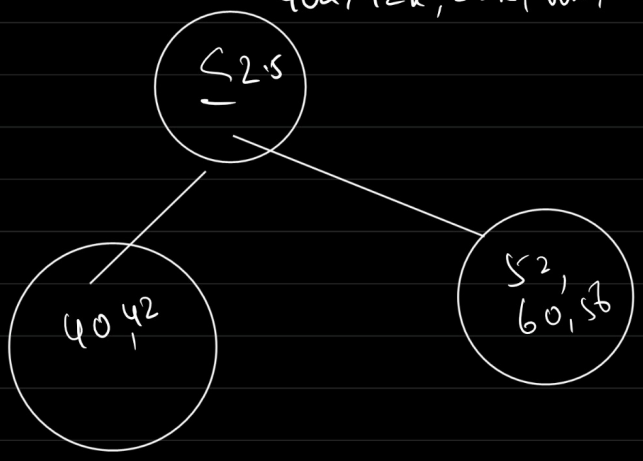
$$= 51$$

$$\text{Variance reduction} = \text{Var P} - \text{V child combi}$$

$$= 60.0 - \left(\frac{1}{5} \times 100 + \frac{4}{5} \times 51\right)$$

$$= 0$$

40k, 42k, 52k, 60k, 56k



Var P_{root} = 60.8

Var (left child) = $\frac{1}{2}(100 + 64)$

= 82

Var (right child) = $\frac{1}{3}(4 + 36 + 100)$

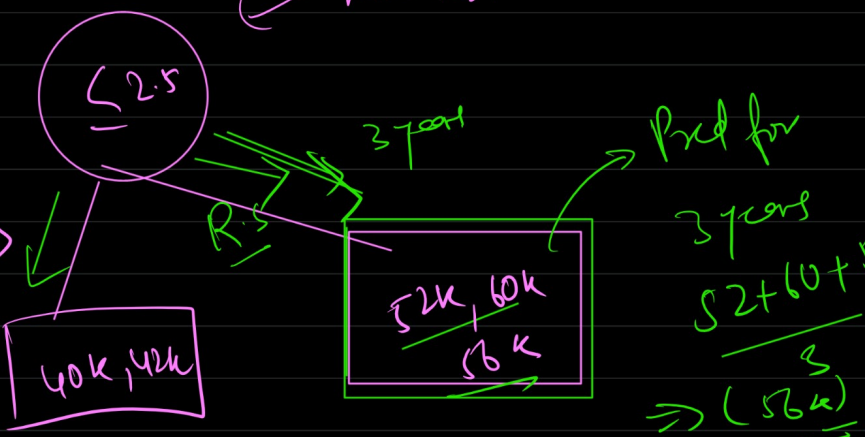
= 46.66

Variance reduct = $60.8 - \left(\frac{2}{5} \times 82 + \frac{3}{5} \times 46.66\right)$

= 0.004

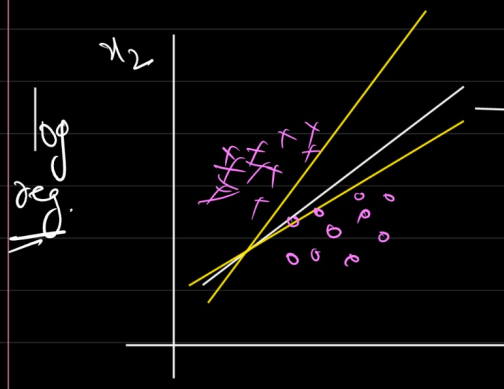
V.R for 2.5 < Variance reduct for 2.5

← new test data ⇒ 3 years



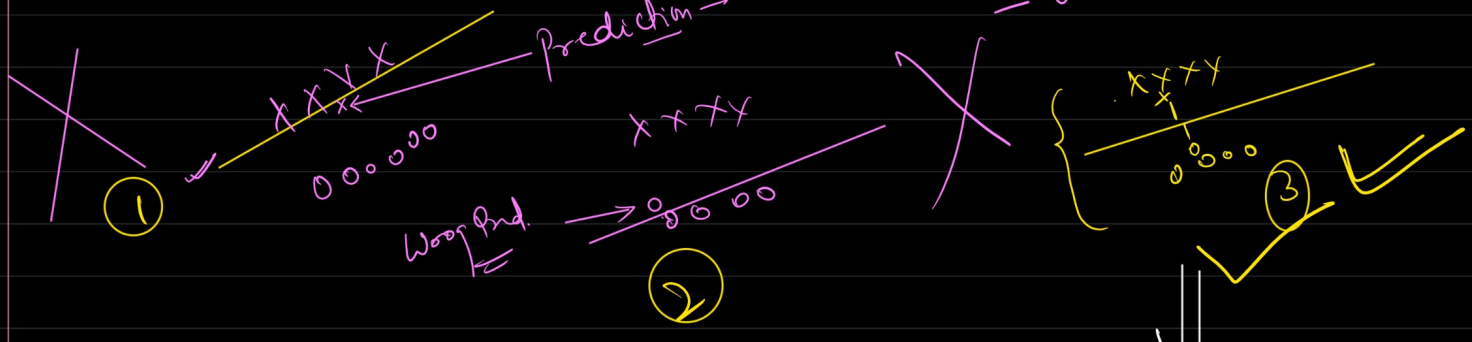
SVM (Support Vector Machines)

- Support Vector classifier → Classification (SVC)
- " " Regressor → Regressor (SVR)



→ Logestic Regression finds linear decision boundary.

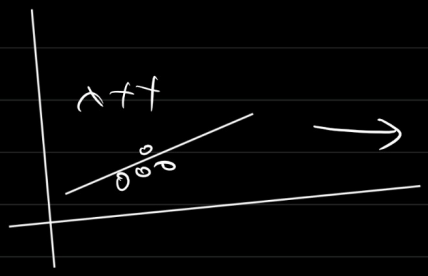
→ Any line can be decision boundary if it segregates both the class.



SVM

* Logistic regression model doesn't care about margin / space across two class.

* if the data point changes slightly, log reg will give error.

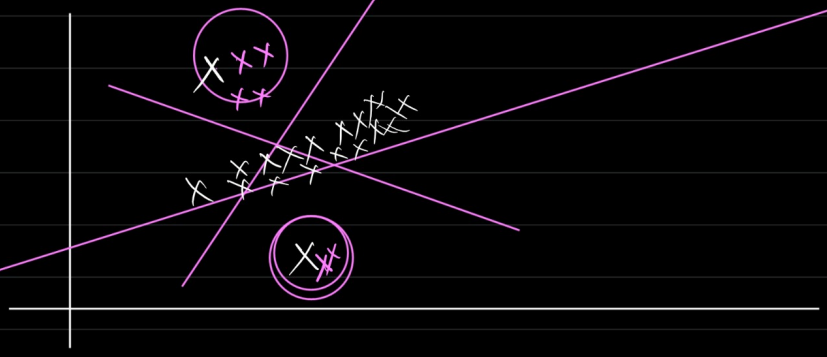


→ Prone to error.

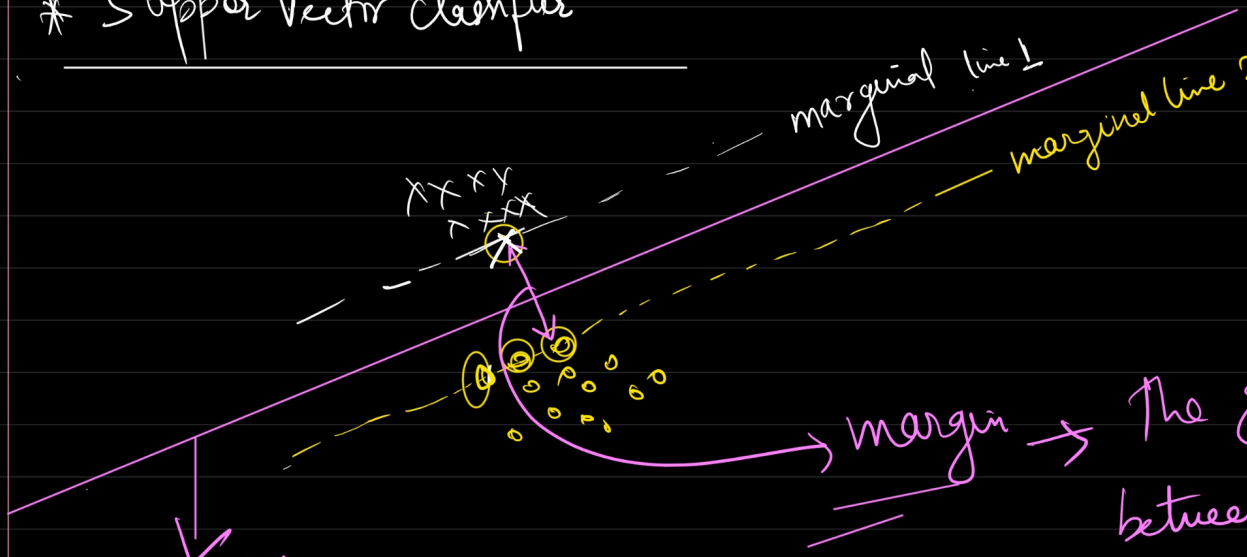
wrt Regression

↓

SVM Regression



* Suppor vectr classifier



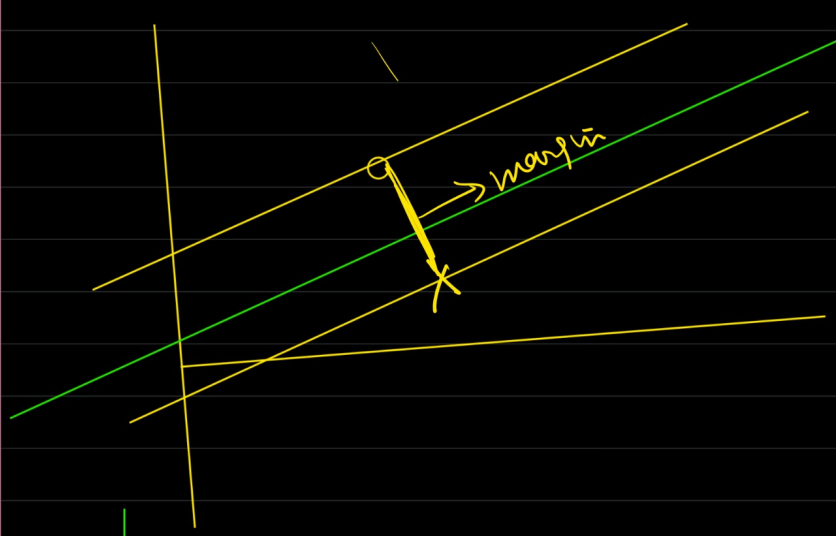
margin → The gap between nearest datapoint of both class

↓

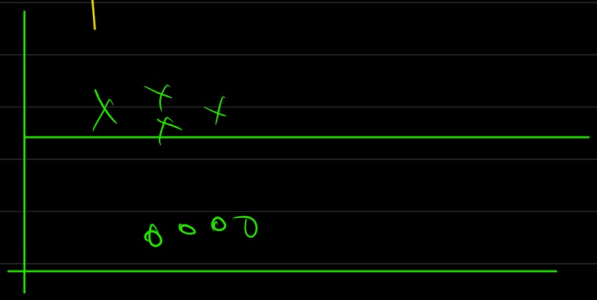
Support Vectors

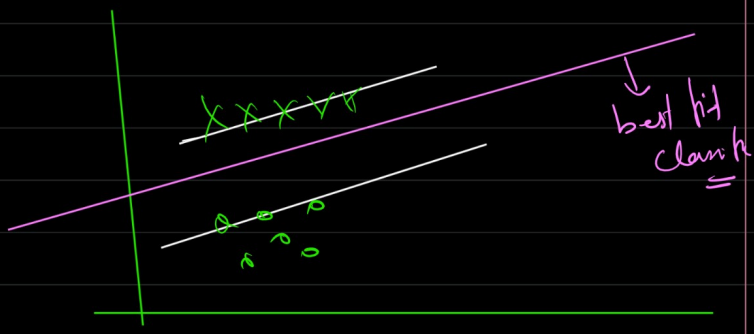
best fit line
decision boundary

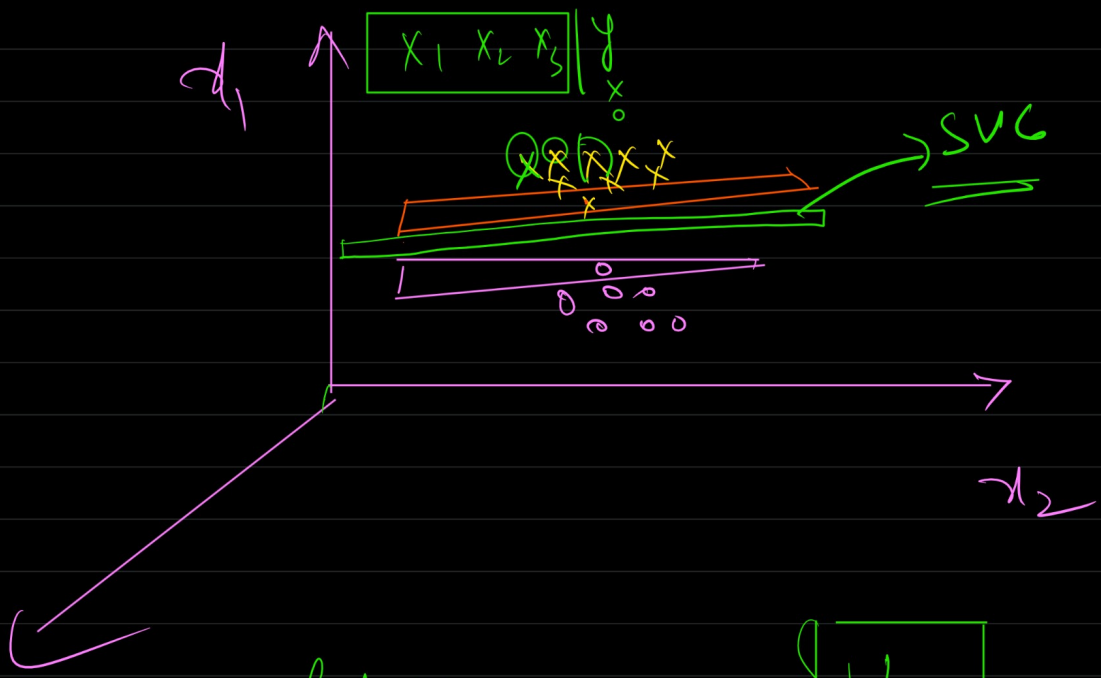
To make marginal line for both the class you need atleast two support vectors.



SVC

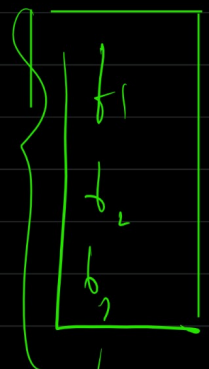






feature selection

↳ Gini / Entropy




XG Boost
Adaboost