

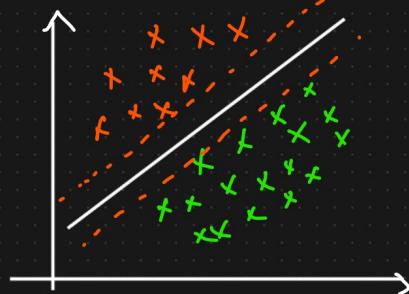
Agenda

- ① SUM Kernels → Practicals
- ② Bagging & Boosting Algorithms → Random Forest, Adaboost
- ③ Pick up any problem statement
- ④ ROC and AUC Curve } → Performance Metrics

Applied Data
II Scientist

Amazon, Microsoft

① SVC & SVR



Cost function

Minimize
(w, b)

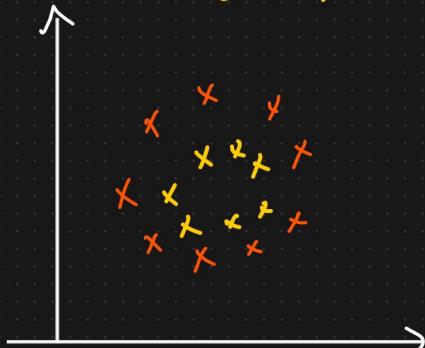
$$\frac{\|w\|}{2} + C \sum_{i=1}^n \xi_i$$

Constraint

$$y_i w^T x_i + b \geq 1$$

→ for all correct points.

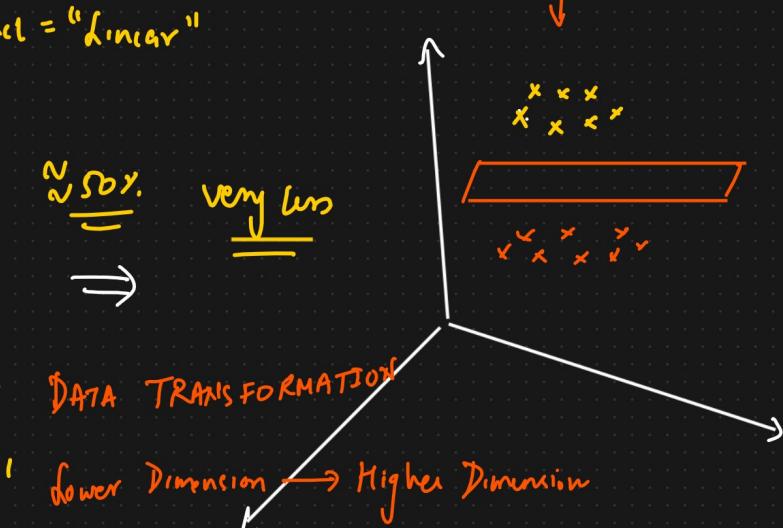
SVM → Kernel = "Linear"



\approx very less

DATA TRANSFORMATION

lower Dimension → Higher Dimension



PCA = High Dimension → Low Dimension

① Polynomial ✓

2D → 3D

x^2



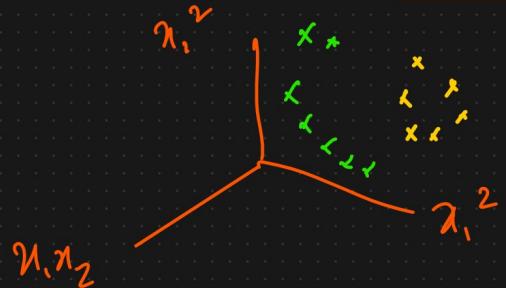
Applying one mathematical formula

$$d=2$$

Polynomial Kernel = $(\underline{\underline{x^T \cdot x}} + 1)^d$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \begin{bmatrix} x_1 & x_2 \end{bmatrix} = \begin{bmatrix} x_1^2 \\ x_1 x_2 \\ x_1 x_2 \\ x_2^2 \end{bmatrix}$$

$$x_1 \quad x_2 \quad \begin{bmatrix} x_1^2 & x_1 x_2 & x_2^2 \end{bmatrix} \Rightarrow \begin{bmatrix} \text{Linear} \\ \text{Hyperplane} \end{bmatrix}$$



Sklearn Kernel = "poly"

② RBF Kernel $\left\{ \text{Panasonic} \right\}$

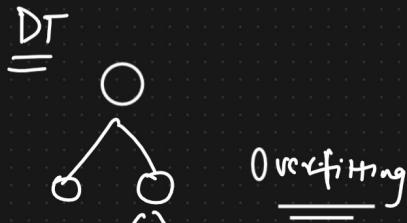
$$x_1 = \{1, 2, 3, 1\}$$

$$x_2 = \{5, 6, 7, 8\}$$

$y_{stack}(x_2)$

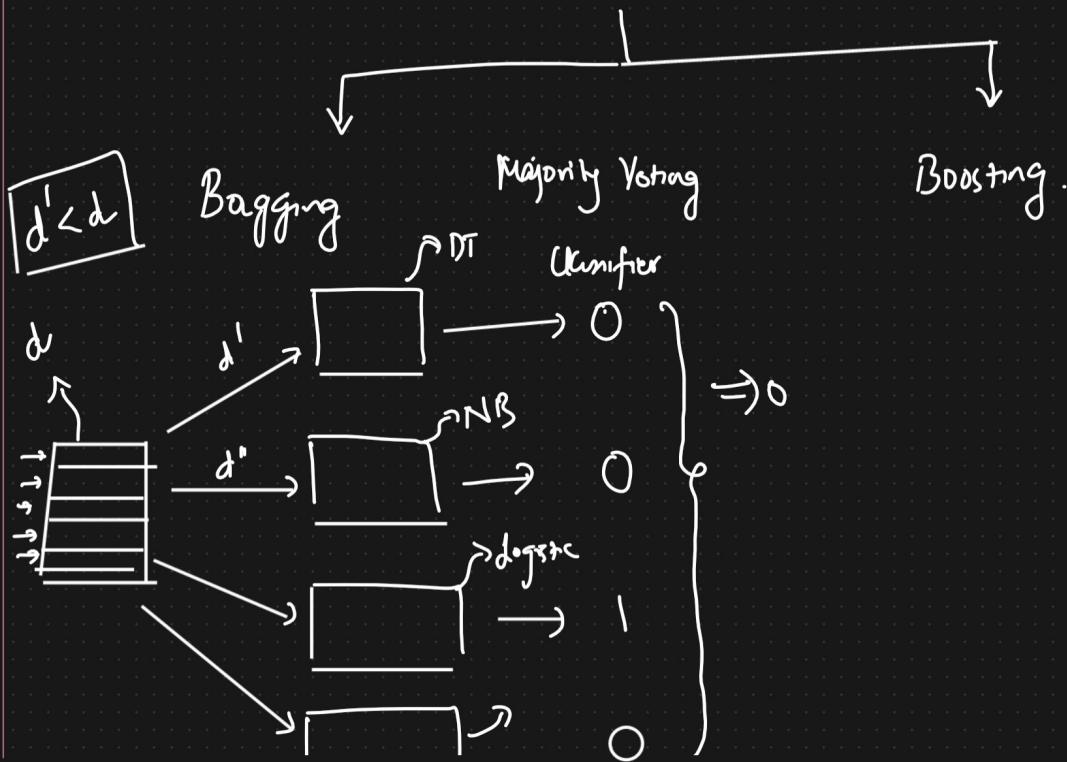
1
2
3
4
5
6
7
8

④ Decision Tree \Rightarrow Overfitting



$\Rightarrow \left\{ \begin{array}{l} \text{low Bias} \\ \text{High Variance} \end{array} \right\} \downarrow$
 low Variance $\left\{ \begin{array}{l} \text{Ensemble Techniques} \end{array} \right\}$

Ensemble Technique

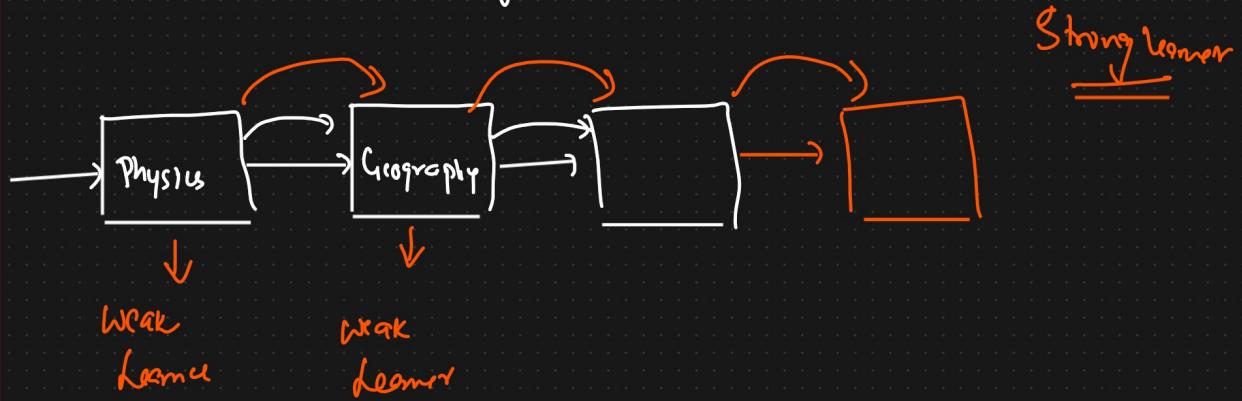


Avg(O/P)

Boosting → KBC

{ AdaBoost
Gradient
XGBoost }

→ Question → History Subject → Sequential



Strong Learner

Bagging

Boosting

① Random Forest Classification & Regression

① AdaBoost

② CatBoost

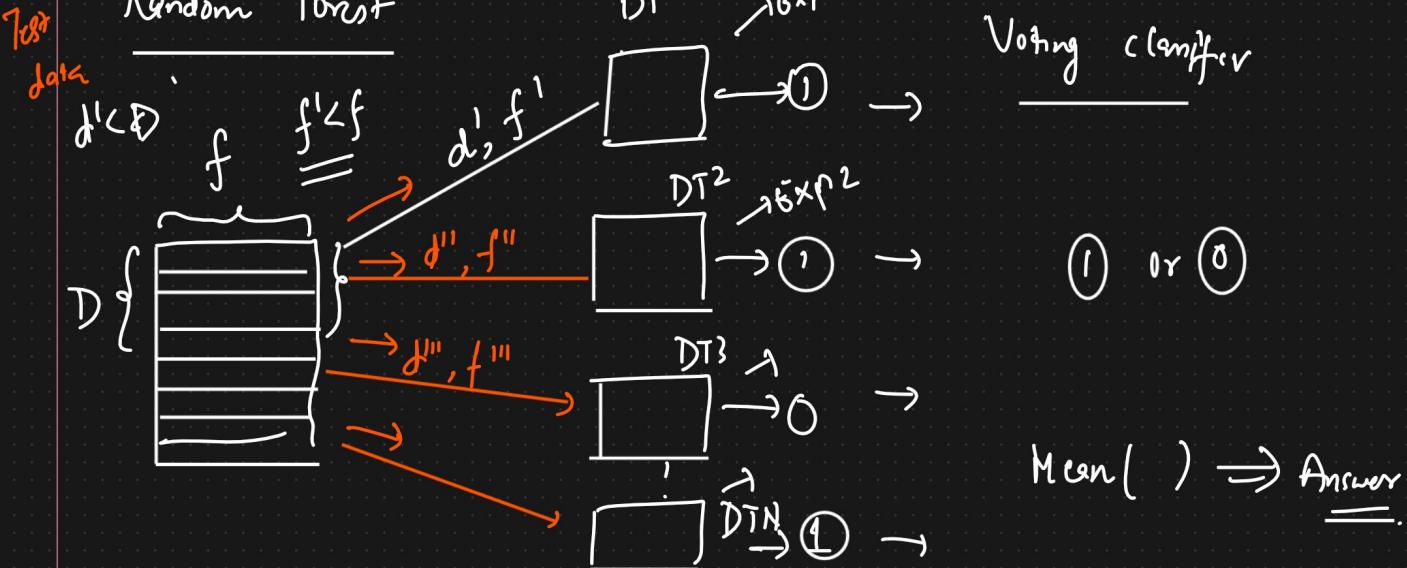
③ Gradient Boost

④ XgBoost

f^l = Subset of features

d^l = Subset of datapoints
Classification

Random Forest



{
 low Bias
 high Variance
 }

Performance Metrics

ROC and AUC Curve

[0, 0.2, 0.4, 0.6, 0.8, 1]

Healthcare Problem

Threshold = 0.5

y	\hat{y}_{pred}	$\hat{y}(0)$	$\hat{y}(0.2)$	$\hat{y}(0.5)$
1	0.8	1 ✓	1	1
0	0.96	1	1	1
1	0.4	1 ✓	1	0
1	0.3	1 ✓	1	0
0	0.2	1	0	0
1	0.7	1 ✓	1	1

$$TPR = \frac{2}{2+2} = 0.5$$

$$FPR = 0.5$$

$$TPR = \frac{4}{4+0} = 1$$

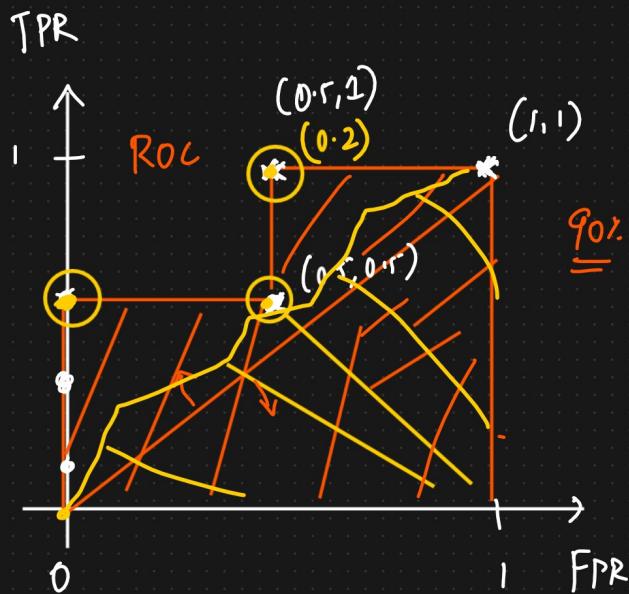
$$FPR = \frac{1}{(1+1)} = 0.5$$

Probability

$$TPR = \frac{TP}{TP+FN} = \frac{4}{4+0} = 1$$

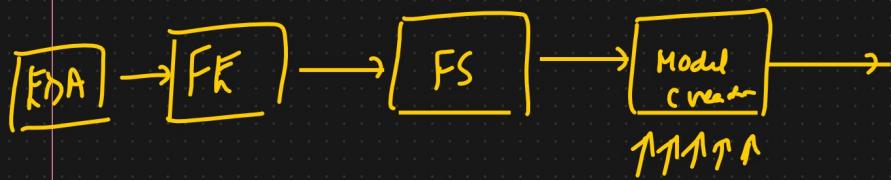
$$FPR = \frac{FP}{FP+TN} = \frac{2}{2+0} = 1$$

Domain Expertise \Rightarrow TP

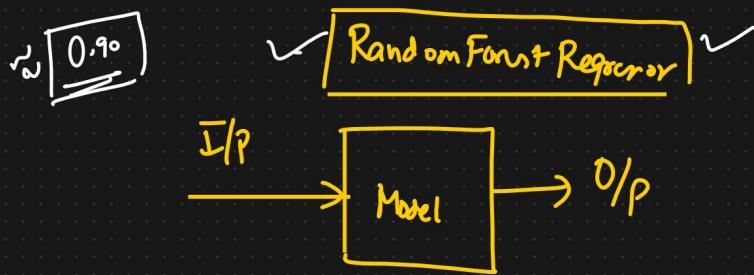


AUC

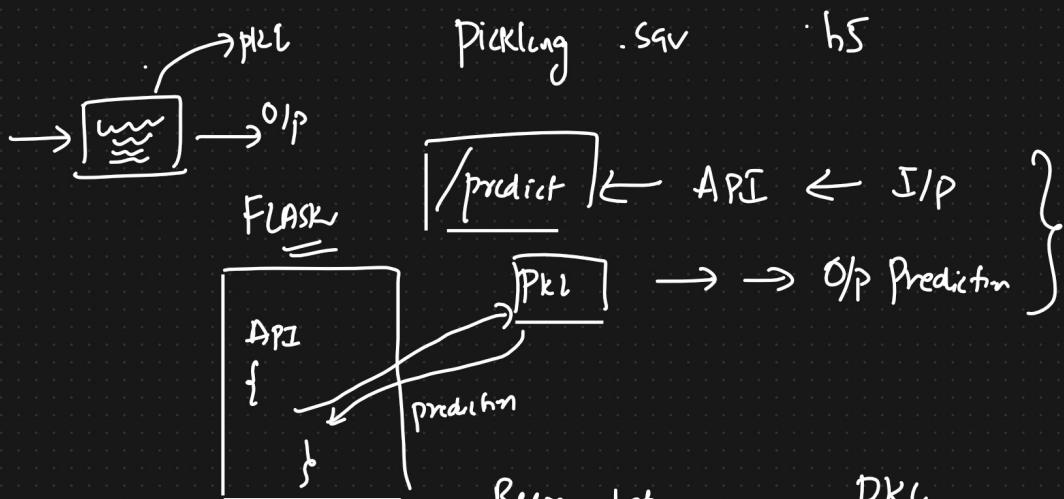
Threshold and Model performs well.



How should be
use this model



Serialized object format



{ → Model per
 → Postman }
 ==

Flask app

/predict

{
 load the model
 }