#100DAYSOFML

100 Days Of Machine Learning

MACHINE LEARNING ALGORITHMS

Sameer Kulkarni
sameerk126

Contents

1. Linear Regression
2. Ridge and Lasso Regression
3. Logistic Regression
4. Decision Trees
5. Support Vector Machines
6. Naïve Bayes
7. Random Forest Algorithm
8. Boosting Algorithm
9. Gradient Boost
10. XG Boost
11. K Nearest Neighbour
12. Principle Component Analysis (PCA)
13. K - Means Clustering
14. DBSCAN
15. Hierarchial Clustering

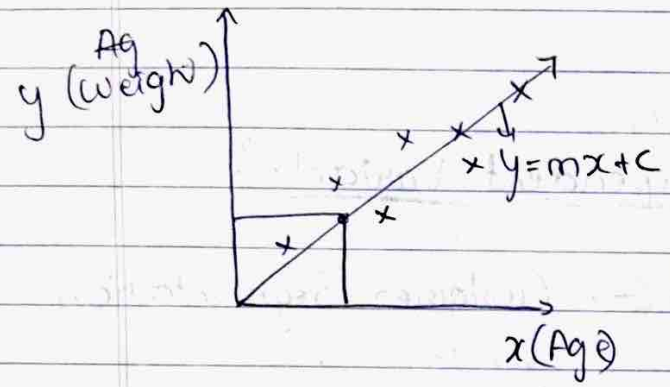
#100DAYSOFML

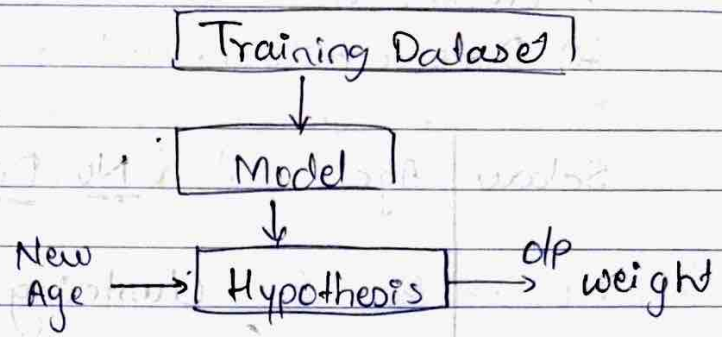
Machine Learning Algorithms

1.Linear Regression

Sameer Kulkarni
sameerk126

① Linear Regression





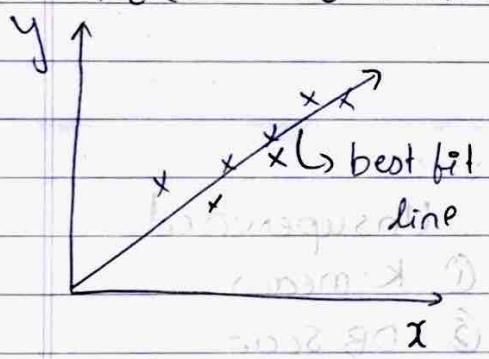
y is a linear function of x

Equation of straight line → $y = mx + c$

$y = B_0 + B_1 x^{(i)}$

$h_\theta(x^{(i)}) = \theta_0 + \theta_1 x^{(i)}$

$h_\theta(x) = \theta_0 + \theta_1 x$



θ_0 - intercept when $x=0$ $y=?$

θ_1 - Slope - with unit movement in x axis moment in Y axis

x_i - Data point

Aim → Dist b/w predicted line & Datapoint must be least

Hypothesis $h_\theta(x) = \theta_0 + \theta_1 x$

Cost function - $\sum (h_\theta(x) - y)^2$

$h_\theta(x)$ - predicted value

$$\boxed{J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2}$$

↳ Square error function

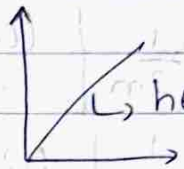
$\frac{1}{m}$ - gives the avg value

$\frac{1}{2}$ - Derivation purpose

What we need to solve

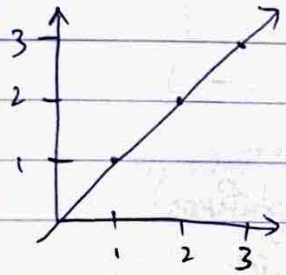
Minimize $\rightarrow \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \longrightarrow J(\theta_0, \theta_1)$

(*) $h_\theta(x) = \theta_0 + \theta_1 x$ if $\theta_0 = 0$



$\hookrightarrow h_\theta(x) = \theta_1 x$

$h_\theta(x) = \theta_1 x$ $\theta_1 \rightarrow$ slope



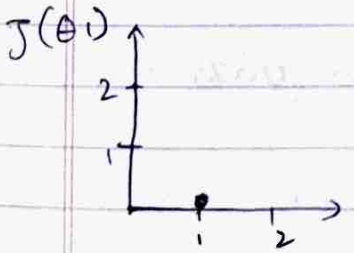
(1,1)
(2,2)
(3,3)

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{3}\left(h_\theta(x^i) - y^i\right)^2$$

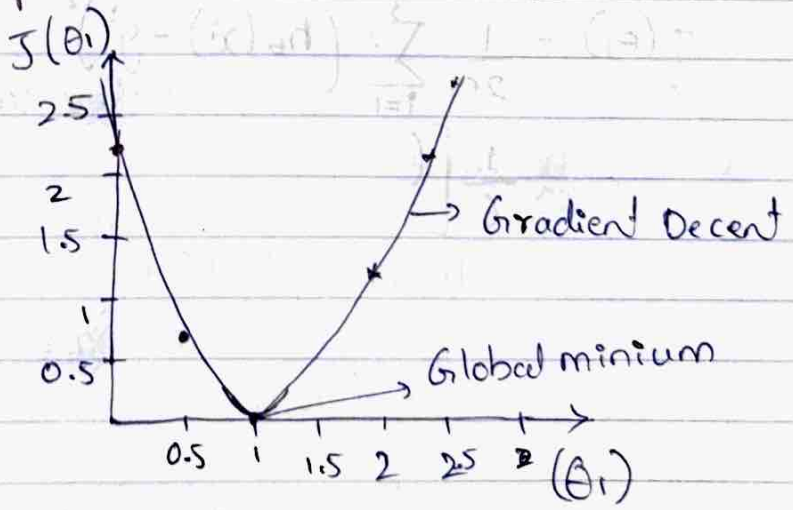
$$J(\theta_1) = \frac{1}{2m}\left[(1-1)^2 + (2-2)^2 + (3-3)^2\right] = 0$$

When $\theta_1 = 1$ $J(\theta_1) = 0$

Cost function graph



$\theta_1 = 0.5$ $J(\theta) = 0.58$



ii) $\theta_1 = 0.5$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} \left(h_\theta(x^i) - y^i \right)^2$$

$$= \frac{1}{2m} \left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2 \right]$$

$$= \frac{1}{2\times 3} \left[0.25 + 1 + 2.25 \right] = 0.58$$

$\theta_1 = 0$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} \left(h_\theta(x^i) - y^i \right)^2$$

$$= \frac{1}{2m} \left[(0-1)^2 + (0-2)^2 + (0-3)^2 \right]$$

$$= \frac{1}{6} \left[1+4+9 \right] \approx 2.3$$

$\theta_1 = 2$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{3} \left(h_\theta(x^i) - y^i \right)^2$$

$$\frac{1}{6} \left[(\right.$$

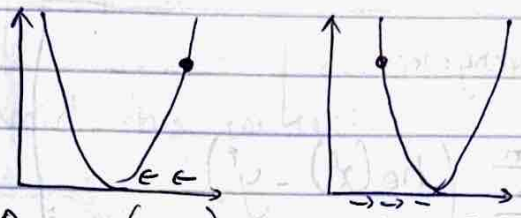
Convergence algorithm → Repeat untill convergence

loop
{

$$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \qquad \theta_1 = \theta_1 - \alpha\,(+ve)$$

}

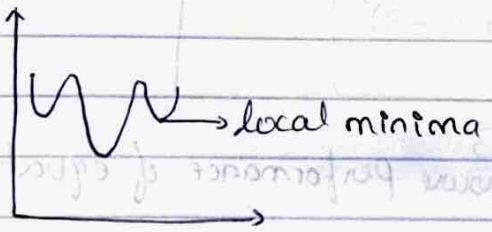
α → learning Rate



$\theta_1 = \theta_1 - \alpha\,(+ve)$ $\qquad$ $\theta_1 = \theta_1 - \alpha\,(-ve)$

α - learning rate → speed of coming to global minima
ideal value 0.01

Interview Question



Convergence algorithm might get stuck in local minima
But it is not occured in ML
cost function will not give local minima
In DL we have optimizers which will solve the problem

★ Gradient Decent algorithm

$$\frac{\partial}{\partial x} x^2 = 2x$$

repeat until convergence
{

$$\theta_j = \theta_j - \alpha \left(\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) \right\}$$

$$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right)^2$$

$$j=0 \Rightarrow \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)$$

$$j=1 \quad \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) x^i$$

Repeat until convergence

{

$$\theta_0 := \theta_0 - \alpha \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right)$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(h_\theta(x^i) - y^i \right) x^i$$

}

<u>Performance Matrix</u> -> measure performance of equation

R^2 and Adjusent R^2

$$R^2 = 1 - \frac{SS_{Res}}{SS_{Total}} = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2}$$

$h_\theta(x)$ → Predicted point

$\Downarrow$

$$= 1 - \frac{\text{low}}{\text{High}} \Big\} \text{ big no}$$

$\downarrow$

Small no

location Bedrooms Price Goals $R^2 = 85\%$

after adding corelated feature

$R^2 = 90\%$ — loc^n

$R^2 = 91\%$ → Gender

Non corelated feature should not increase R^2

To avoid this we use

Adjusted R^2

$$R^2_{adjusted} = 1 - \frac{(1-R^2)(N-1)}{N-P-1}$$

#100DAYSOFML

Machine Learning Algorithms

2.Ridge and Lasso Regression

Sameer Kulkarni
sameerk126

2

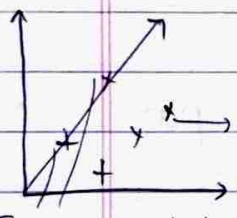
Ridge And Lasso Regression

Ridge & lasso regression
Assumption of linear regression
Logistic regression
Confusion Matrix
Practicals for linear, Ridge & lasso, logestic regression

Ridge and Lasso regression

Revision

$$\text{Cost function } J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta x^{(i)} - y^{(i)} \right)^2$$



$J(\theta_0, \theta_1) = 0$

New data point

Training data

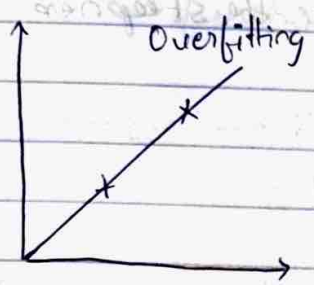
Over fitting - Even though model performed well with train data but it fails to perform well with test data

Model perform well → Train data → (low Bias)
Fails to perform well - test data → High variance

Underfitting

Model accuracy bad with Train data → High Bias
Model accuracy is also bad with Test data → High variance

Model-1	Model-2	Model-3
Training Acc = 90%	Training Acc = 92%	Training Acc = 70%
Test Acc = 80%	Test Acc = 91%	Test Acc = 65%
↓	↓	↓
Overfitting	Generalized model	Underfitting
Low Bias	low Bias	High bias
High variance	low variance	High variance



$$J(\theta_1) = 0$$

$$= \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 \qquad h_\theta(x^{(i)}) = \hat{y}$$

$$= \left(\hat{y}^{i} - y^{(i)}\right)^2$$

$$= 0$$

This should not be zero as the graph is overfitting

<u>Ridge</u> (l_2 Regularization) → it adds a unique parameter $\lambda(\text{slope})^2$

$h_\theta(x) = \theta_0 + \theta_1 x$

$h_\theta(x) = \theta_1 x$

↓

taking slope θ_1 & multipling tot

$$= \left(\hat{y}^{i} - y^{(i)}\right)^2 = 0$$

⇓

$$\left(\hat{y}^{i} - y^{(i)}\right)^2 + \lambda(\text{slope})^2$$

lets assume $\lambda = 1$ → $0 + 1(2)^2$ Assuming slope $= \theta_1 = 2$

$= 4$

Cost function again reduces the θ_1 to get generalized fitting

λ → Hyperparameter

$$\left(\hat{y}^{i} - y^{(i)}\right)^2 + \lambda(\text{slope})^2$$

$= (\text{Small value}) + 1(\text{decrease} \downarrow = 1.5)$ Value gets reduced from 4 to 3

$= (\text{Small value}) + 2.25$

Now we ~~cat~~ can get generalized model of low bias and low variance

We use Ridge to reduce overfitting

* Slope should not be steep we may get overfitting

$\lambda \rightarrow$ Hyper parameter

↳ Specifies how fast we want to reduce the steepness

Lasso regression (L1 Regularization)

$$= (\hat{y}^i - y^{(i)})^2 + \lambda |slope|$$

↳ Helps to do feature selection

lets consider i have many features

$$h_\theta(x) = \hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 + \cdots$$

$$|slope| = \cancel{x_1 + x_2 + x_3} + \cdots \theta_0' + \theta_1 + \theta_2 + \theta_3 + \cdots + \theta_n$$

Using lasso ① Preventing overfitting

② Feature selection → by ignoring less important features

$\lambda \rightarrow$ Hyper parameter

↳ Cross validation

Ridge regression (l_2 norm)

Cost function = $(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda (\text{slope})^2$

Purpose → preventing overfitting

Lasso regression (l_1 regularization)

Cost function = $(h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda |\text{slope}|$

Purpose → Prevent overfitting, feature selection

Assumption of linear regression

① If our feature are in Normal (Gaussian Dist → the model will train well

② Standardization (Scaling data using zsquare $\mu=0, \sigma=1$)
↳ Wherever gradient decent is involved it is good to use Standardization

③ Work w.r.t linearity

④ Check for multi colinearity → x_3 $\boxed{x_2 \quad x_1}$ y
↳ How much features are corelated if its high we can drop any one

(*) Variation Inflation factor

#100DAYSOFML

Machine Learning Algorithms

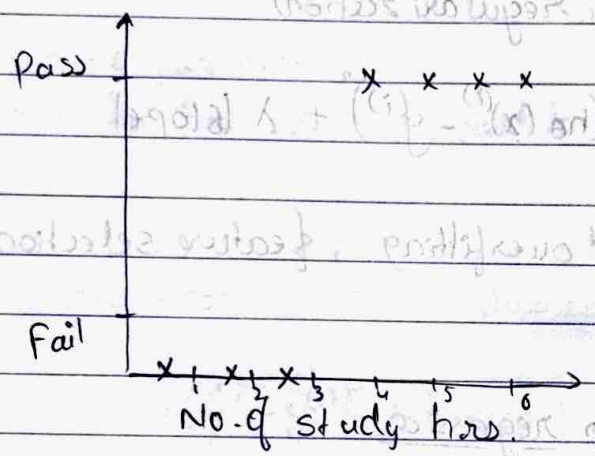
3.Logistic Regression

Sameer Kulkarni
sameerk126

Logistic Regression (classification algorithm)

No. of study hours	No. of play hrs	P/F
-	-	P
-	-	P

* Works very well with Binary classification

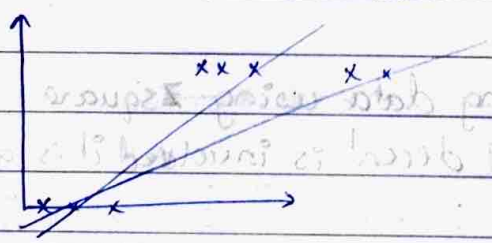


Can we solve with linear regression

$h_\theta(x) < 0.5 \Rightarrow 0$ - Fail

$h_\theta(x) \geq 0.5 \Rightarrow 1$ - Pass

Why we are not using linear regression
- When outliers are occured



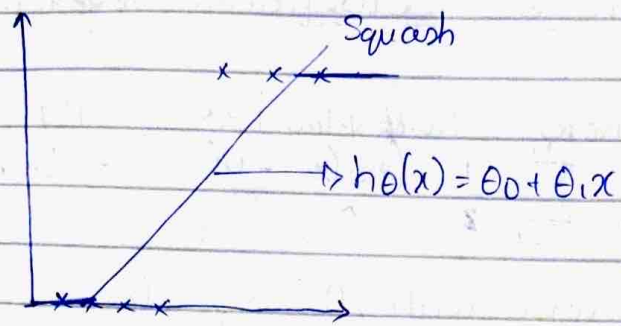
* Because of outlier entire line is changed ~~because~~ ~~of out~~
* Output can also be -ve

To squash f^n we use Sigmoid f^n bcz our values should always be between 0-1

Decision Boundary Logistic Regression

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$

$$h_\theta(x) = \theta^T x$$



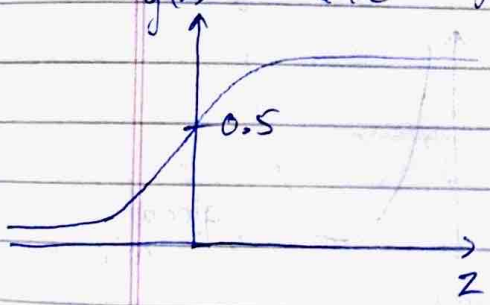
Squash

$h_\theta(x) = g(\theta_0 + \theta_1 x)$ Let $z = \theta_0 + \theta_1 x$

$= g(z)$

$h_\theta(x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}$ → This function is able to squash the line

$h_\theta(x) = \frac{1}{1+e^{-z}}$ } Sigmoid or logistic function



$g(z) \geq 0.5$ when $z \geq 0$

Problem Statement

Training Set → $\{(x^1, y^1), (x^2, y^2), (x^3, y^3) \ldots (x^n, y^n)\}$

$y \leftarrow \{0, 1\}$ 2 output

$h_\theta(z) = \frac{1}{1+e^{-z}}$ $z = \theta_0 + \theta_1 x$

lets consider $\theta_0 = 0$, $z = \theta_1 x$

Cost function

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{n} \frac{1}{2} \left(h_\theta(x) - y^{(i)}\right)^2$$ – Linear regression

Logistic regression

$$h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$$

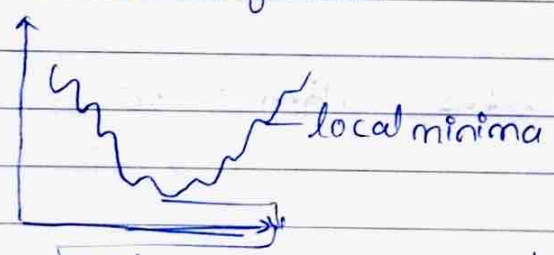
We cannot use this cost function for logistic

Cost function ⟶

$$\frac{1}{2m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

Because $h_\theta(x)$ of logistic Reg: is non convex function

Non convex function



↳ we will never reach global minima

Convex function

Logi

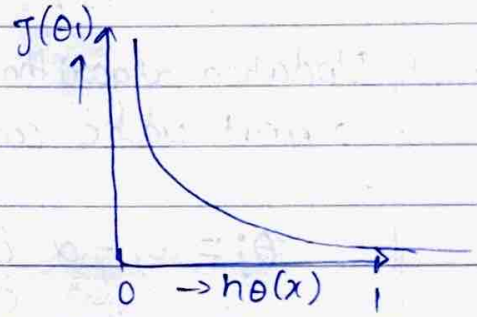
Logistic Regression cost function

$$J(\theta_1) = \begin{cases} -\log(h_\theta(x)) & y=1 \\ -\log(1-h_\theta(x)) & y=0 \end{cases} \qquad h_\theta(x) = \frac{1}{1+e^{-\theta_1 x}}$$

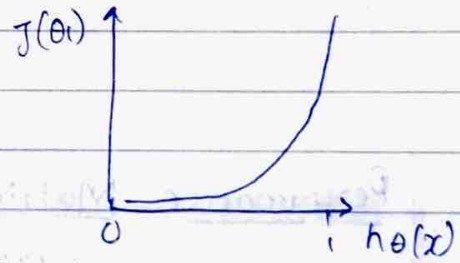
if $y=1$

Assumtions of curve

$y=1$ cost $=0$ if $h_\theta(x)=1$



if $y=0$



$$\text{Cost}(h_\theta(x^{(i)}), y^i) = \begin{cases} -\log(h_\theta(x)) \rightarrow y=1 \\ -\log(1-h_\theta(x)) \end{cases}$$

Combining values

$$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \underbrace{-y\log(h_\theta(x^i))}_{0 \text{ if } y=0} - \underbrace{(1-y)\log(1-h_\theta(x))}_{0 \text{ if } y=1}$$

if $y=1$

$\text{cost}(h_\theta(x^i), y) = -\log(h_\theta(x^i))$

if $y=0$

$\text{cost}(h_\theta(x^i), y) = -\log(1-h_\theta(x^i))$

$$\boxed{J(\theta) = \frac{-1}{2m} \sum_{i=1}^{m} \left(y^i \log(h_\theta(x^i)) + (1-y^i) \log(1-h_\theta(x^i)) \right)}$$

$$h_\theta(x^i) = \frac{1}{1+e^{-(\theta_1 x)}}$$

Updation algorithm

repeat until converge

$$\left\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial j} (J(\theta)) \right\}$$

<u>Performance Matrices</u> {Classification problem}

x_1	x_2	y	$\hat{y}$ (predicted value)
		0	1
		1	1
		1	1
		0	0
		1	1
		1	1
		0	1
		1	0

(Predict) \ (Actual)	1	0
1	3	1
0	1	1

P \ A	1	0
1	3	2
0	1	1

	1	0
1	TP	FP
0	FN	TN

gives right output

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$$= \frac{3+1}{3+2+1+1} = \frac{4}{7} = 0.57 \rightarrow 57\% \text{ accuracy}$$

Imbalance Data

0 → 900

1 → 100

↓

Model predicts → 0

Accuracy $= \frac{900}{1000}$

$= 90\%$

good accuracy but biased data

Balanced data

0 → 600

1 → 400

Imbalance data affects the algorithm

Predicted \ Actual	1	0
1	TP	FP
0	FN	TN

Precision

Recall

#IMP

① Precision

$\frac{TP}{TP+FP}$

out of all the predicted values how many are actual TP

② Recall

$\frac{TP}{TP+FN}$

Out of all the actual True values how many are predicted as TP

reduce FN

③ F-Score

{Spam classification} - Precision

{Cancer or not} → Recall

{Stock market crash}

F-Score (F-Beta) → $\frac{(1+\beta^2)\,\text{Precision} \times \text{Recall}}{\beta^2 \times \text{precision} + \text{Recall}}$

Whenever both FP & FN are important we use F-score

#100DAYSOFML

Machine Learning Algorithms

4.Decision Trees

Sameer Kulkarni
sameerk126

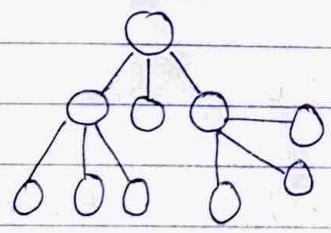
Decision Tree

1) Decision tree classifier
2) Decision tree regressor

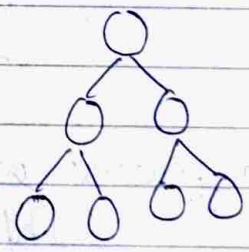
Decision tree classifier

Two types
- → ID3 (Iterative Dichotomiser 3)
- → CART (Classification & Regression tree)

ID3



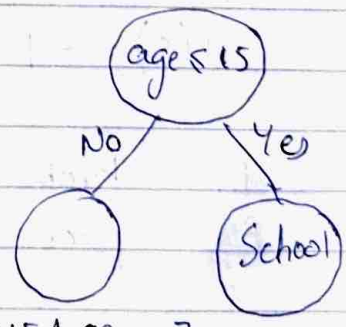
CART

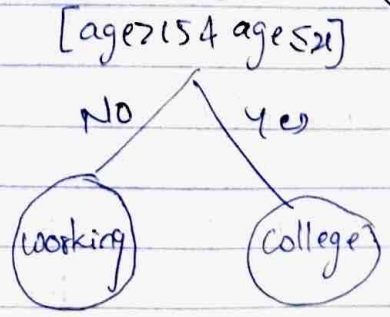


#only binary split

Decision tree - multinested if else conditions

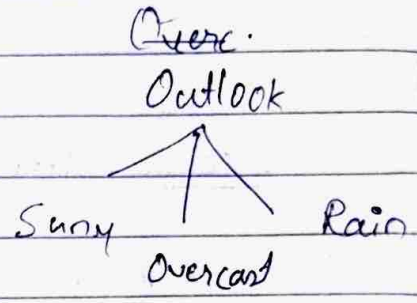
```
if age ≤ 15:
    print ("school")
elif (age > 15 & age ≤ 21)
    print ("college")
else
    print ("working")
```





Example dataset:

Outlook - [Sunny, overcast, rain]
Temp [Hot, mild, cold]
Humidity - [High, Normal
Wind - [weak, strong]
Play tennis [Yes, No]



* Purity check : pure split or impure split
  ↳ Measure of purity
    ↳ Entropy
    ↳ Gini impurity

* What feature we need to select to start the split
  ↳ Information gain technique

Entropy

$$H(S) = -P_+ \log_2 P_+ - P_- \log_2 P_-$$

P_+ = Prob. of +ve category
P_- = prob. of -ve category

Gini Impurity

$$G.I = 1 - \sum_{i=1}^{n} (P)^2$$

#100DAYSOFML

Machine Learning Algorithms

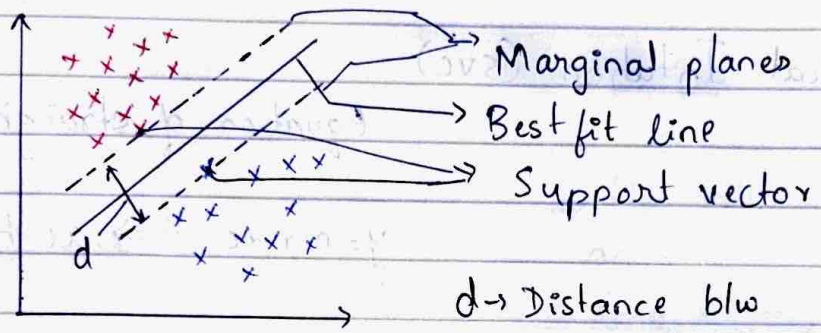
5.Support Vector Machines

Sameer Kulkarni
sameerk126

Support vetor Machines

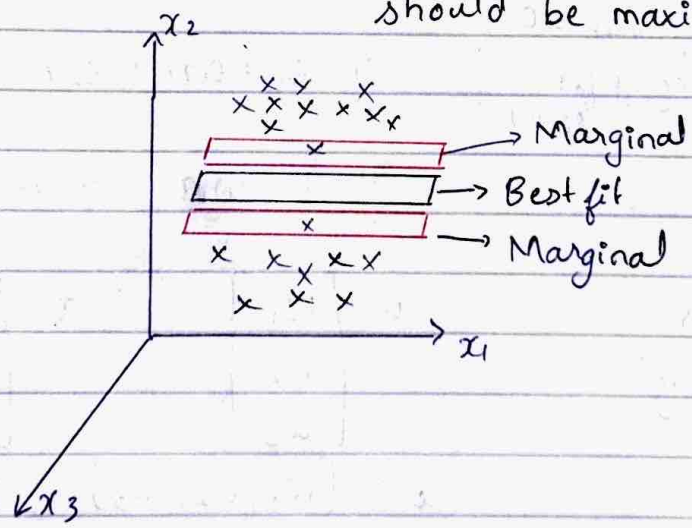
① SVC (Support vector Classifier)
② SVR (Support vector regressor)

1) Support Vector Classifier



[Marginal plane will pass through nearest point of those specific Category]

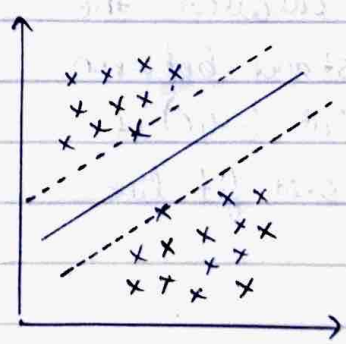
d → Distance b/w marginal planes should be maximum



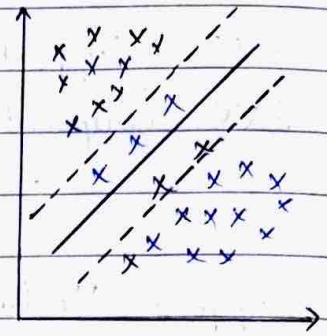
Soft margin and hard margin in SVC

1) Hard margin - No points are missclassified

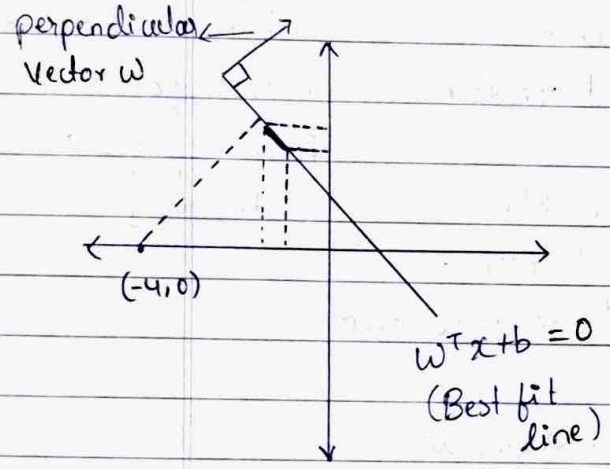
Data points are clearly saperated by hard margin



2) Soft margin : Some data points are misclassified



Mathematical Intution (SVC)



Equation of straight line

$y = mx + c \Leftrightarrow ax + by + c = 0$

$h_\theta(x) = \theta_0 + \theta_1 x_1$

$y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$

$y = b + w_1 x_1 + w_2 x_2 + w_3 x_3$

$\boxed{w^T x = 0}$

if the line is passing through Origin

$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}$ $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

$w^T = [w_1, w_2, w_3]$ $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

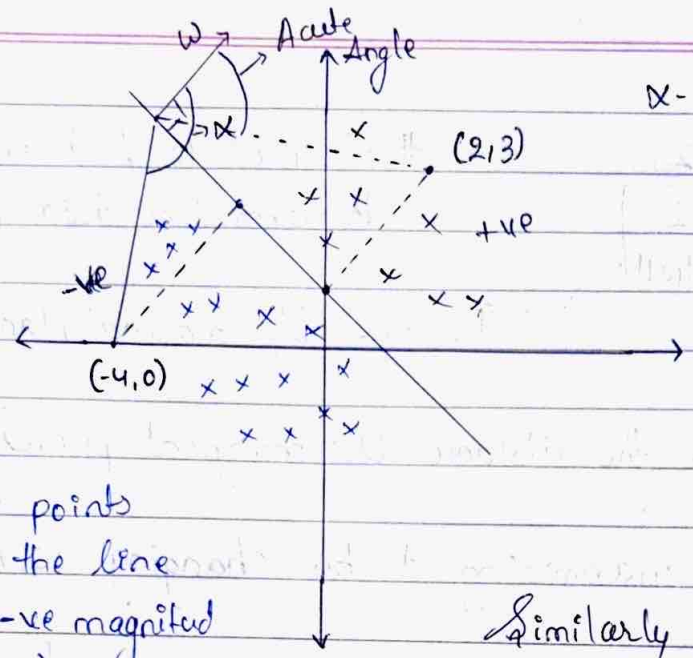
Use of perpendicular vector :

To calculate the distance between point (-4,0) & best fit line

$w^T * x$

$y = b + [w^T * x]$

$y = w^T x + b$



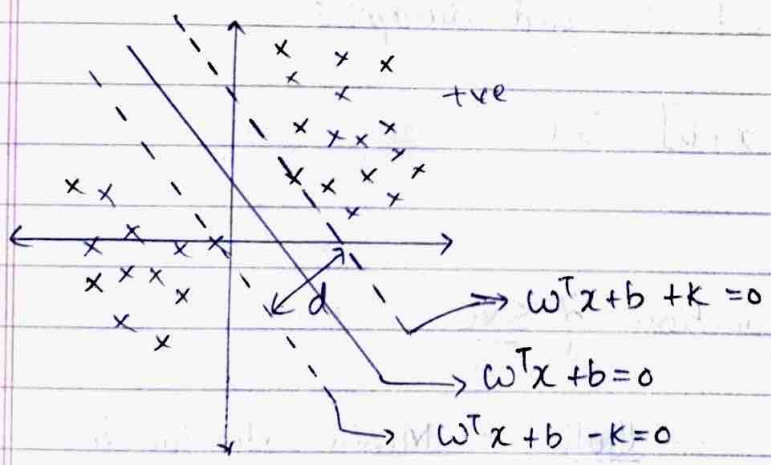
α- Angle b/w vector w & point (-4,0)

If angle > 90: magnitude of length b/w (-4,0) & best fit line is **Negative**

All the points below the line have -ve magnitud (Distance)

Similarly for point (2,3) the angle is <90° thus the distance (magnitude) is +ve (Pos

Ploting Marginal planes



points above the line have positive distance to the best fit line

points below the line have -ve distance

Calculation of d (distance b/w marginal planes)

Assuming k=1 & substracting eqn of two marginal planes

$$w^T x + b = +1$$
$$w^T x + b = -1$$
$$\oplus \quad \ominus \quad \oplus$$

$$\boxed{w^T (x_1 - x_2) = 2}$$

$$w^T(x_1 - x_2) = 2$$

We need to find unit vector to normalize data points

$$\frac{w^T}{\|w\|}(x_1 - x_2) = \boxed{\frac{2}{\|w\|}}$$

→ Distance b/w marginal planes (d)

Aim: Maximize the distance b/w marginal plane

Cost function: maximizing d by changing w, b

Constraint : such that

$$y_i = \begin{cases} +1 & \text{if } w^T x + b \geq 1 \\ -1 & \text{if } w^T x + b \leq -1 \end{cases}$$

For all the correct classified datapoint

$$y_i * [w^T x + b] \geq 1$$

<u>Modified cost function of SVC</u>

Minimize concept $\Longrightarrow \frac{\|w\|}{2}$ → Minimization function

Constraint such that

$$y_i \begin{cases} +1 & \text{if } w^T x \geq 1 \\ -1 & \text{if } w^T x \leq -1 \end{cases}$$

Cost function of SVC (soft margin)

↳ Misclassified datapoints

$$\min_{w,b} \frac{\|w\|}{2} + \boxed{C_i \sum_{i=1}^{n} \xi_i} \longrightarrow \text{Hinge Loss}$$

$C_i \rightarrow$ Hyper parameter
{How many points we can consider for misclassification}

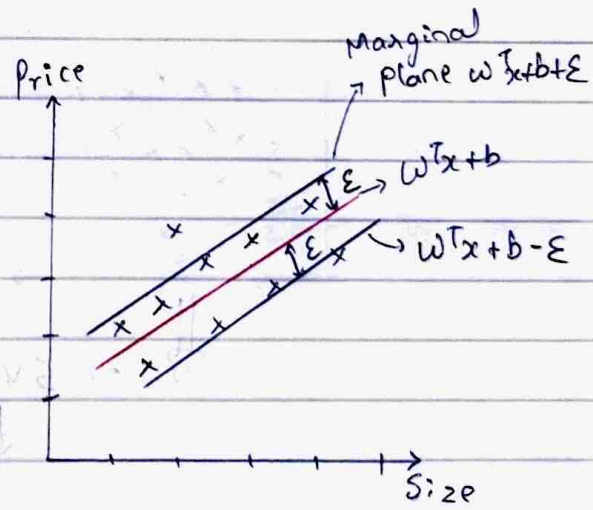
$\xi \rightarrow$ Eta
{Summation of distance of incorrect data points from marginal plane}

II Support vector regressor (SVR)

Best fit line $\rightarrow w^T x + b$

Marginal planes $\rightarrow w^T x + b + \varepsilon$

$w^T x + b - \varepsilon$



Cost function

$$\min_{w,b} \frac{\|w\|}{2} + \boxed{C \sum_{i=1}^{n} \xi_i} \rightarrow \text{Hinge loss}$$

classmate
Date
Page

<u>Constraint</u>

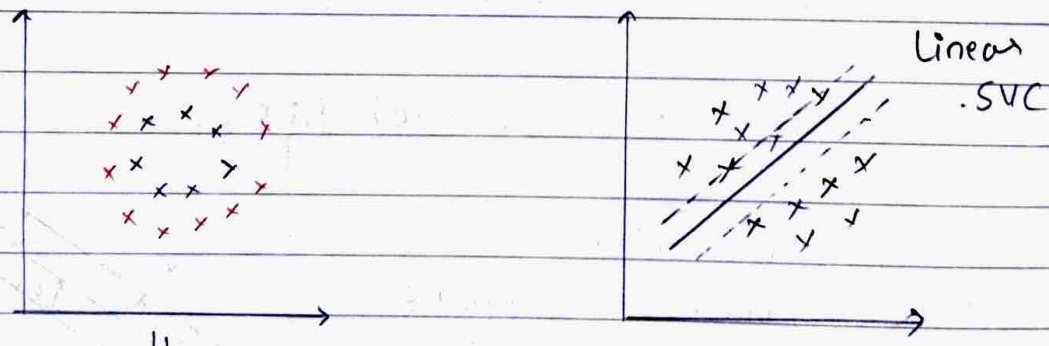
$\xi_i \rightarrow$ Summation of datapoint to marginal plane

$$|y_i - w^T x_i| \leq \varepsilon + \xi_i$$

* The diff. b/w real & predicted data point should be less than ε

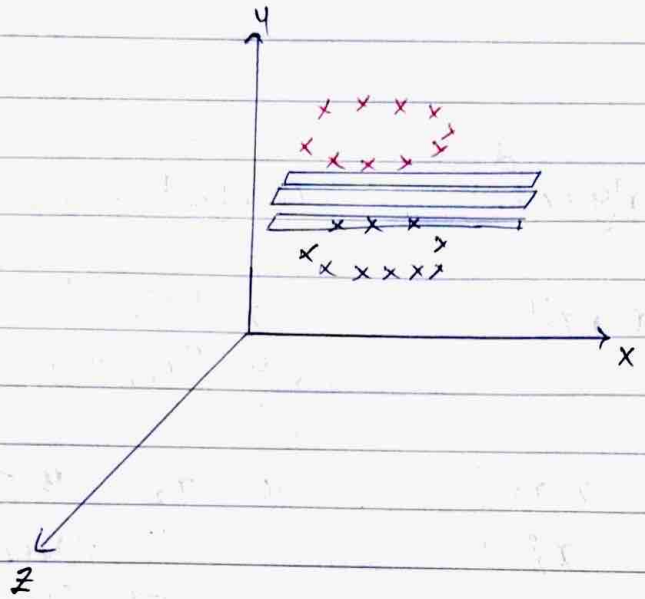
$\varepsilon \rightarrow$ Marginal error
$\xi_i \rightarrow$ Error above the margin

<u>SVM Kernels</u>



$\Downarrow \rightarrow$ SVM Kernels (transforms the datapoints
$\Downarrow$ Transforms (changing features

Mathematical formula
to make changes in features
2D $\rightarrow$ 3D
Two feature $\rightarrow$ 3 feature



Example :-

Dataset

	x	y
4	2	Yes
16	4	No
36	6	~~Yes~~ No
64	8	Yes

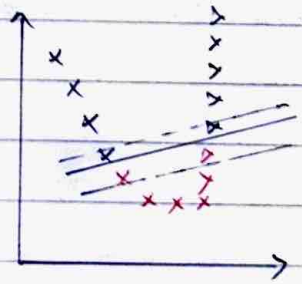
↓ ploting

After Transformation ⟹

SVM Kernel Transformation

Creating new feature

~~y = x~~ $z = x^2$



Types of Kernel

1. Polynomial kernel
2. RBF kernel
3. Sigmoid Kernel

i) Polynomial kernel

$$f(x,y) = (x^T y + c)^d \qquad c \to \text{Constant} \Rightarrow 1, 2, 3$$

$$= \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} * [x_1 \; x_2] \qquad x = [x_1, x_2] \quad y = [x_1, x_2]$$

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

$x_1 \quad x_2 \quad \cancel{y} \to 2D$

$\Downarrow$ Polynomial Transformation

$x_1^2 \quad x_1 x_2 \quad x_2^2$

#100DAYSOFML

Machine Learning Algorithms

6.Naïve Bayes

Sameer Kulkarni
sameerk126

Naive Baye's Algorithm

1. Probability [Independent & Dependent variable]
2. Baye's theorem
3. Naive Baye's Math intution

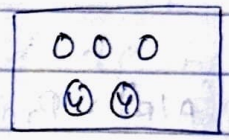
1. Probability

Independent event

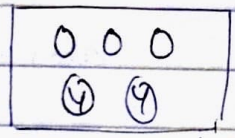
Rolling a dice [1,2,3,4,5,6]

$Pr(1) = 1/6$

Dependent events



What is probability of first removing then removing yellow marble



1. $p(0) = 3/5 \rightarrow 1^{st}$ event
2. $p(y) = 2/4 \rightarrow 2^{nd}$ event

$p(y|0) = \frac{2}{4} \longrightarrow$ Conditional probability

$$Pr(0 \text{ and } y) = p(0) * p(y|0)$$
$$= 3/5 * 2/4$$
$$= \frac{3}{10}$$

$$\boxed{Pr(A \text{ and } B) = Pr(A) * Pr(B|A)} \rightarrow \text{Bayes Theorem}$$

Baye's theorem

$$Pr(A \text{ and } B) = Pr(B \text{ and } A)$$
$$Pr(A) * Pr(B/A) = Pr(B) * Pr(A/B)$$

$$\boxed{Pr(A/B) = \frac{Pr(A) * Pr(B/A)}{Pr(B)}}$$

$P(A/B) \rightarrow$ Probability of event A given B has occured
$P(B/A) \rightarrow$ probability of event B given A has occured
$P(A) \rightarrow$ Probability of event A
$P(B) \rightarrow$ Probability of event B

Dataset

x_1	x_2	x_3	o/p = Y
-	-	-	Yes
-	-	-	NO

$$P(Y/(x_1,x_2,x_3)) = \frac{P(Y) * P((x_1,x_2,x_3)/Y)}{Pr(x_1,x_2,x_3)}$$

$$P(Y/(x_1,x_2,x_3)) = \frac{Pr(Y) * P((x_1,x_2,x_3)/Y)}{Pr(x_1,x_2,x_3)}$$

$$= \frac{Pr(Y) * Pr(x_1/y) * Pr(x_2/y) * Pr(x_3/y)}{Pr(x_1) * Pr(x_2) * Pr(x_3)}$$

$$Pr(Y=Yes \mid (x_1, x_2, x_3)) = \frac{Pr(Yes) * Pr(x_1 \mid Yes) * Pr(x_2 \mid Yes) * Pr(x_3 \mid Yes)}{Pr(x_1) * Pr(x_2) * Pr(x_3)}$$

$$Pr(Y=No \mid (x_1, x_2, x_3)) = \frac{Pr(No) * Pr(x_1 \mid No) * Pr(x_2 \mid No) * Pr(x_3 \mid No)}{Pr(x_1) * Pr(x_2) * Pr(x_3)}$$

Basically we calculate probability of o/p being yes and No. Which o/p has highest probability. then the o/p of new test data will be respective output

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Sunny yes / Total no of yes

Outlook

	Yes	NO	P(E|Yes)	P(E|NO)
Sunny	2	3	2/9	3/5
Overcast	4	0	4/9	0/5
Rain	3	2	3/9	2/5

Temperature

	Yes	NO	P(E|Yes)	P(E|NO)
Hot	2	2	2/9	2/5
mild	4	2	4/9	2/5
Cool	3	1	3/9	1/5

Yes \| NO → Yes = 9 P(Y) = 9/14

NO = 5 P(N) = 5/14

(Sunny, hot) → o/p ?

$$Pr(Yes \mid (Sunny, Hot)) = Pr(Yes) * Pr(Sunny \mid Yes) * Pr(Hot \mid Yes)$$

$$= \frac{9}{14} * \frac{2}{9} * \frac{2}{9}$$

$$= \frac{2}{63} = 0.031$$

$$Pr(No / (Sunny, Hot)) = Pr(No) * Pr(Sunny / No) * Pr(hot / No)$$

$$= \frac{5}{14} * \frac{3}{5} * \frac{2}{5}$$

$$= 0.085$$

<u>Finally</u>

$$Pr(Yes / (Sunny, hot)) = \frac{0.031}{0.031 + 0.085} = 0.27 = 27\%$$

$$Pr(No / (Sunny, Hot)) = \frac{0.085}{0.031 + 0.085} = 0.73 = 73\%$$

New Data [Sunny, Hot] $\Rightarrow$ 73% NO $\Rightarrow$ 0
27% Yes

$\Downarrow$

Then the person will not play Tennis

<u>Variants of Naive Bayes</u>

1. Bernoulli Naive Bayes
2. Multinomial Naive Bayes
3. Gaussian Naive Bayes

i) Bernoulli Naive Bayes

Whenever features are following Bernoulli distribution then we use Bernoulli Naive Bayes

Bernoulli Distribution Dataset → 0,1 {Binary variables}

Dataset

f_1	f_2	f_3	o/p
Yes	Pass	M	Yes
NO	Fail	F	NO

$P(\text{Success}) = 1 = P$

$P(\text{Fail}) = 0 = 1-P$

ii) Multinomial Naive Bayes

Could be used when input data is in text format

~~Data~~ Dataset example:

Sentiment analysis

Review message	o/p
Product is good	Positive
Product is bad	Negative

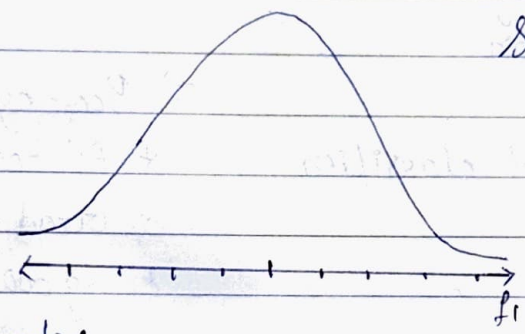
⇓ Convert text to numerical values using NLP

NLP techniques

1. BOW - Bag of words
2. TF-IDF
3. Word2Vec

iii) Gaussian Naive Baye's

Used when features are following Gaussian or normal distribution to solve classification problem



Suitable for continous features

Dataset example:

Age	Height	weight	Yes/No (over weight)
25	170	78	0
35	179	75	1

If we get a senario of mixture of continous & binary features then we should majority feature & proceed with respective algorithm

---

In case of Binary data (Bernoulli dataset)

f_1	f_2	f_3	o/p
Yes	Pass	M	Yes
NO	fail	F	NO
Yes	Fail	M	NO

$\Longrightarrow$

f_1	f_2	f_3	o/p
1	1	1	1
0	0	0	0
1	0	1	0

Whenever we get sparse matrix we can use
+ Bernoulli naive bayes
+ Multinomial naive baye's

This type of dataset also called as <u>sparse matrix</u>

#100DAYSOFML

Machine Learning Algorithms

7.Random Forest Algorithm

Sameer Kulkarni
sameerk126

Ensemble techniques

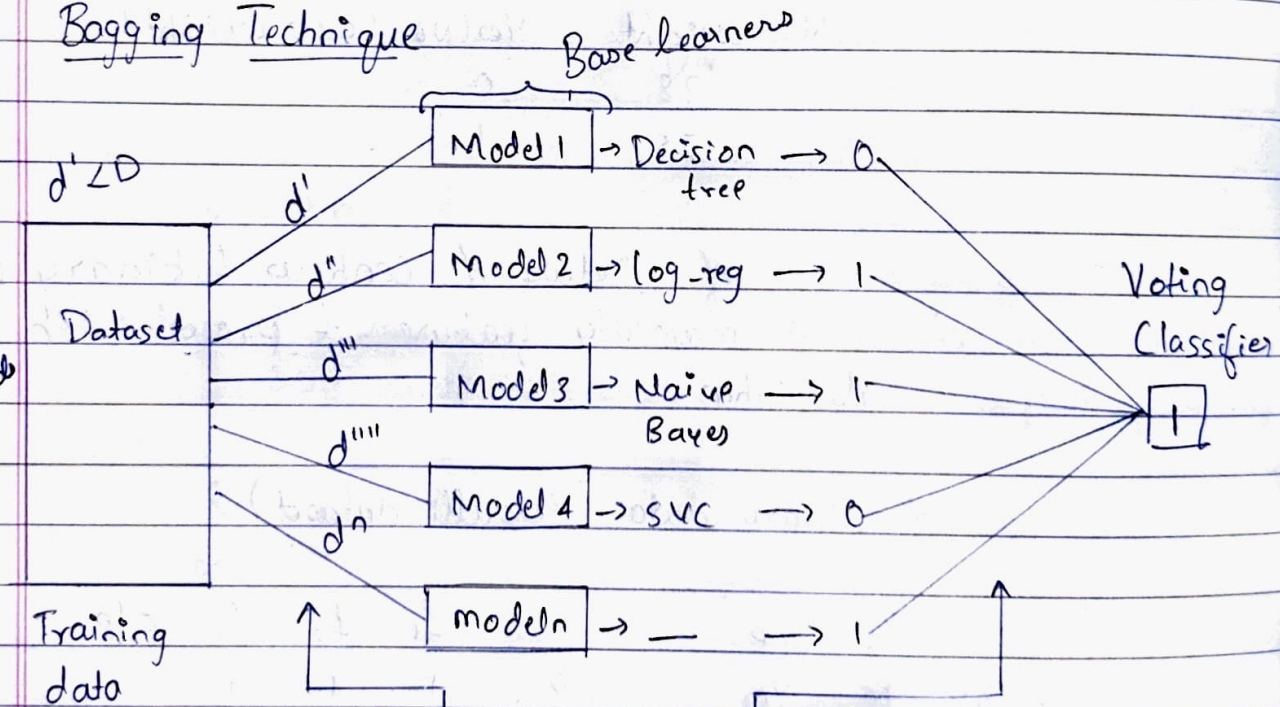
* Ensemble - Combining multiple models & predict using all models

Types of Ensemble techniques
- → Bagging
- → Boosting

i) Bagging
* Random forest classifier & regressor

ii) Boosting
* Adaboost
* Gradient boost
* Xg boost

Bagging Technique



[Bootstrap Aggregration
↓
Dividing Sample data

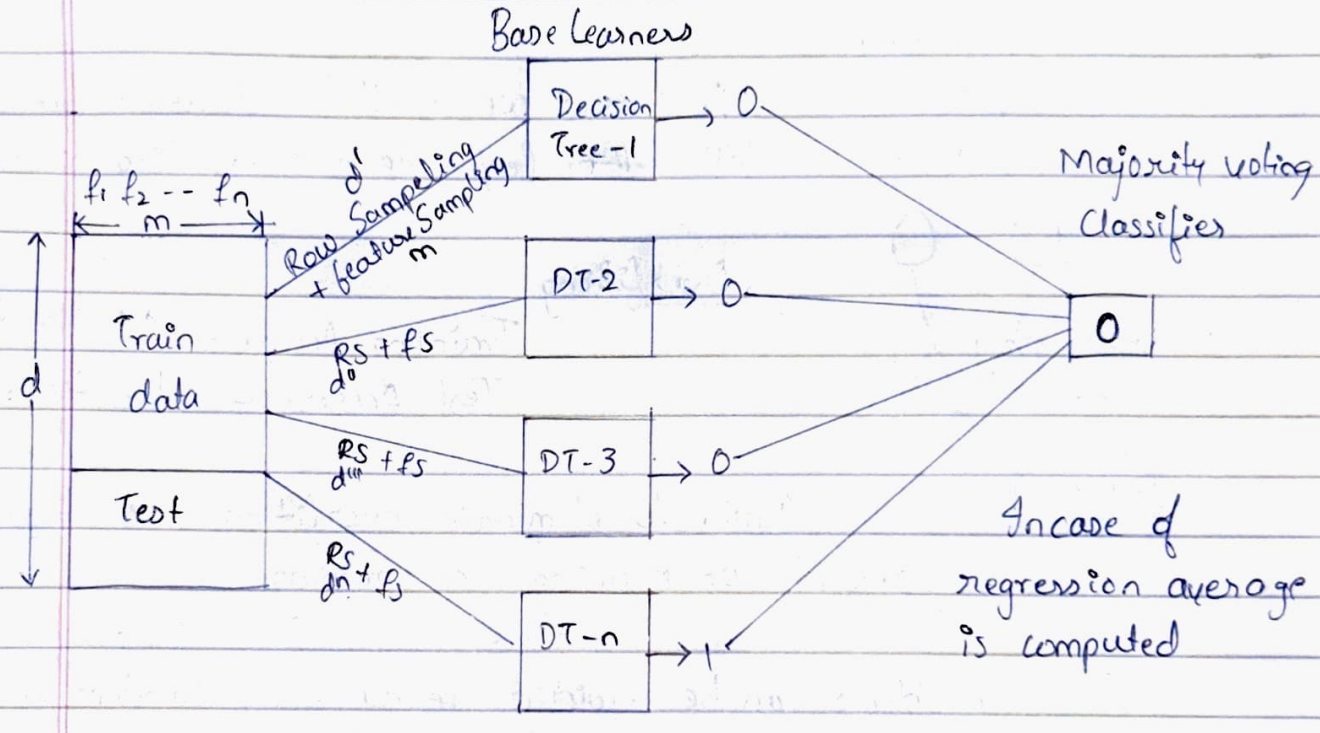
↳ Get aggregate & get most ~~freqy~~ appropriate output

Classification
↳ Custom bagging or custom ensemble technique

In case of regression we take average of all outputs

Random forest Classifier & Regressor

Similar to bagging technique but all the base learners are decision trees



Row Sampling → Selecting sample of records

Feature sampling → Selecting n features (where n<m)

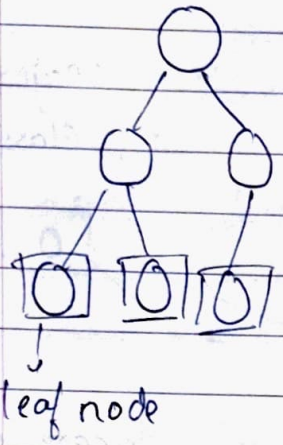
Note

* Classification - Majority voting classifier
* Regression - Average of models

Why should we use random forest insted of decision tree?

Why should we use Random forest insted of Decision tree?

Decision tree



Whenever we split the decision tree till ~~left~~ no leaf node → overfitting occurs

Overfitting

Training Accuracy ↑↑ low Bias
Test Accuracy ↓↓ High variance

Again to eliminate overfitting we should perform prepruning post pruning

These problems can be avoided ~~in~~ by using Random forest as we have trained many decision trees with sample features thus increases test data accuracy

Out of Bag Data

Those rows which are missed during row sampeling of random forest

Parameter → oob_score = True

↓

Out of bag data = validation data = validation accuracy

↓

oob_score

#100DAYSOFML

Machine Learning Algorithms

8. Ada Boost

Sameer Kulkarni
sameerk126

Random forest Parameters

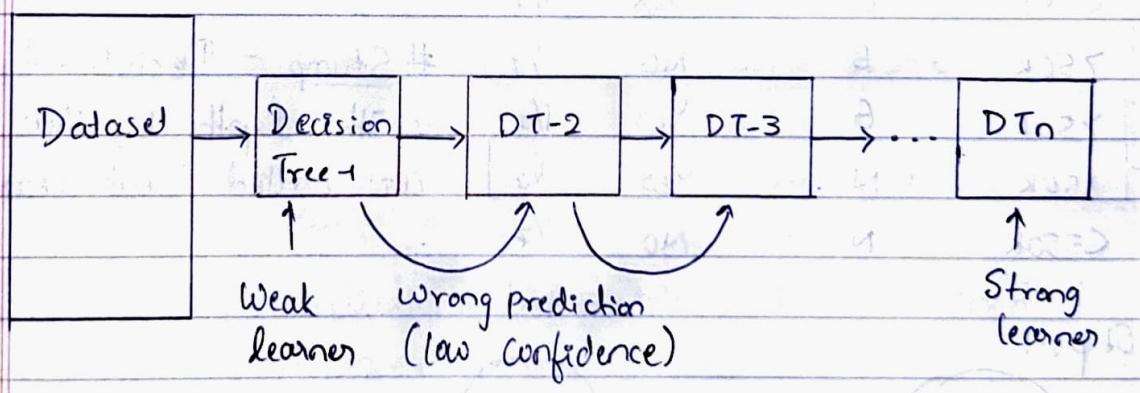
* n-estimators = 100 → No. of decision trees
oob-score = False

Boosting Algorithm

1) Adaboost
2) Gradient boost
3) Xgboost

Boosting Techniques

* Decision trees are sequentially connected



* weak learners :- Haven't learnt much about training dataset

In case of bagging:
Bagging → Random forest → Majority voting

Boosting → Assignment of weights to weak learners

In case of boosting

$f = \alpha_1 (\text{model-1}) + \alpha_2 (M2) + \alpha_3 (M3) + \cdots + \alpha_n (M_n)$ → Classification / → Regression

$\{\alpha_1, \alpha_2 \cdots \alpha_n\}$ → Confidence / weights

Adaboost Classifier → Boosting technique

Dataset [Loan approval]

Sample weight → Step 2

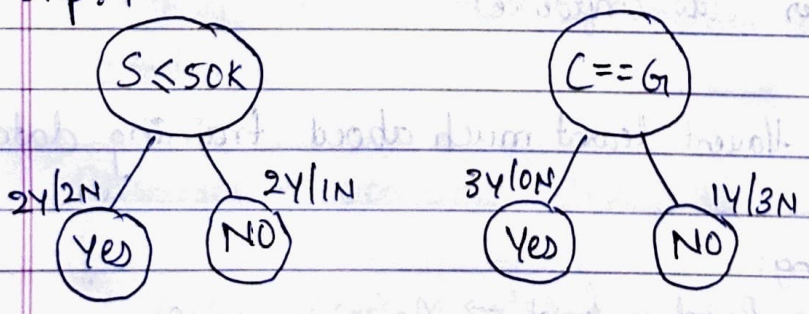
Salary	Credit	Approval	s/w
<=50K	B	No	1/7
<=50K	G	Yes	1/7
<=50K	G	Yes	1/7
>50K	B	No	1/7
>50K	G	Yes	1/7
>50K	N	Yes	1/7
<=50K	N	No	1/7

Step-3 ←

We create a decision tree Stump and we select the best stump

Stump - Decision tree with depth 1, thus these are called weak learners

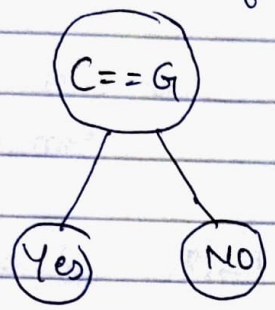
Step :-1



To select the strong stump we should consider gini impurity, Entropy & information gain

Step:2 we should assign sample weight to records

Step:3 Sum of total errors & performance of stump



Sum of total error = 1/7
Because we can observe the 6th record always will be predicted with wrong output

Step:4 performance of stump $= \frac{1}{2} \ln\left[\frac{1-TE}{TE}\right]$

$= \frac{1}{2} \ln\left[\frac{1-1/7}{1/7}\right]$

$= \frac{1}{2} \ln[6]$

≈ 0.896

The performance is used as α in boosting formula

$\alpha_1 = 0.896$

Step 4: Update the weights for correctly & incorrectly Classified points

The wrong predictions from M1 will be trained by model 2 and so on..

Salary	Credit	Approval	s/w	updated weight
<=50K	B	NO	1/7 ↓	0.058
<=50K	G	Yes	1/7 ↓	0.058
<=50K	G	Yes	1/7 ↓	0.058
>50k	B	NO	1/7 ↓	0.058
>50K	G	Yes	1/7 ↓	0.058
>50K	N	Yes	1/7 ↑	0.349
~~>50K~~	N	NO	1/7 ↓	0.058
~~<=50K~~				

$\frac{6}{7}$

$\frac{1}{7} = 0.1428$

Updating weights for correctly classified points

$= \text{weight} * e^{-(\text{Performance of stump})}$

$= \frac{1}{7} * e^{-(0.876)}$

$= 0.058$

For incorrect classified point

$= \text{weight} * e^{(\text{Performance of stump})}$

$=$

$= \frac{1}{7} * e^{(0.876)}$

$= 0.349$

Step: 5 (a) Normalize weight computation & assigning Bin

Salary	Credit	Approval	SW	updated s/w	Normalized wt	Bin Assigned
<=50K	B	No	1/7	0.058	0.083	0 - 0.083
<=50K	G	Yes	1/7	0.058	0.083	0.08 - 0.16
<=50K	G	Yes	1/7	0.058	0.083	0.16 - 0.24
>50K	B	No	1/7	0.058	0.083	0.24 - 0.32
>50K	G	Yes	1/7	0.058	0.083	0.32 - 0.40
>50K	N	Yes	1/7	0.349	0.500	0.40 - 0.90
<=50K	N	No	1/7	0.058	0.083	0.90 - 0.98
				0.697	1	

$\text{Normalized weight} = \frac{\text{Updated s/w}}{0.697}$

The bin assignment creates larger range for in correct data points thus the incorrect record has more chances to be selected in next model

Bins assignments creates datapoints to be selected for next model

Step: 6 Select datapoint for next model

Salary	Credit	approval	Bin assignment
<=50K	B	No	0 - 0.08
<=50K	G	Yes	0.08 - 0.16
<=50K	G	Yes	0.16 - 0.324
>50K	B	NO	0.32 - 0.40
>50K	G	Yes	0.32 - 0.40
>50K	N	Yes	0.40 - 0.90
<=50K	N	NO	0.90 - 0.98

Iteration process for selecting random values from 0 to 1 (Bin)

Step : 7 Final prediction

Test data (<= 50k, G)

f = Test data → DT-1 → DT-2 → DT-3 -- DT-4

DT-1	DT-2	DT-3	DT-4
⇓	⇓	⇓	⇓
Yes	No	Yes	No
$\alpha = 0.896$	0.650	0.24	$\alpha_n = -0.30$

$= 0.896\,(\text{Yes}) + 0.650\,(\text{No}) + 0.24\,(\text{Yes}) - 0.30\,(\text{No})$

$= 1.136\,(\text{Yes}) + 0.350\,(\text{No})$

Final o/p

= Yes

performance of say (Yes) = 1.136

performance of say (No) = 0.350

#100DAYSOFML

Machine Learning Algorithms

9.Gradient Boost

Sameer Kulkarni
sameerk126

Gradient Boosting Algorithm

Regression & Classification

Dataset

x_1	x_2	y	$(y-\hat{y})$ → Step 2
Exp	Degree	Salary	R_1
2	BE	50K	-25K
3	Masters	70K	-5K
5	Masters	80K	5K
6	P.HD	100K	25K

Avg salary = 75 K

Step : 1 Create a base model

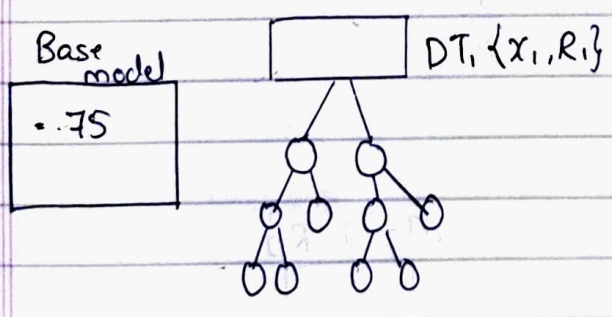
I/P → [ ] → O/P

Avg Sal = 75K = $\hat{y}$

Step : 2 Compute residuals, Error

↳ $R_1 = y - \hat{y}$

Step : 3 Construct a decision tree, consider inputs x_i and outputs as R_1



Decision tree 1 gives Predicted R_2

Exp	Deg	Sal	R_1	Predicted R_2	$\hat{y}$
2	BE	50	-25	-23	74.77
3	Master	70	-5	-3	74.97
5	Mas	80	5	3	
6	P.hD	100	25	23	

final output is calculated by combining prediction of base model and result of decision tree

Predicted output calculation:

$75 + (-23) = 75 - 23 = 52$ {overfitting}

Because 52 is much closer to actual o/p 50

thus,

$= 75 + \alpha(-23)$

$= 75 + -0.23$

$= 74.77$

α = learning rate = {0 to 1} = 0.01

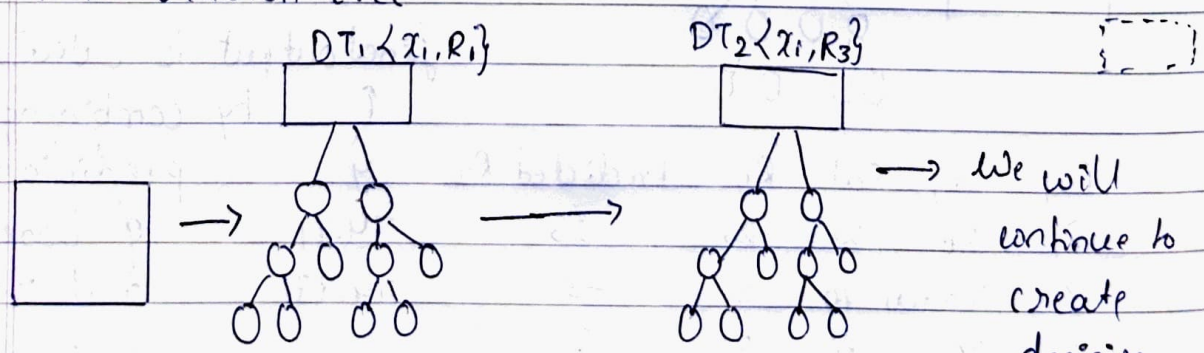
record-2 $= 75 + \alpha(-3)$

$= 75 + -0.03$

$= 74.97$

Exp	Degree	Salary	R_1	Predicted R_2	$\hat{y}$	R_3
2	B.E	50	-25	-23	74.77	-24.77
3	Master	70	-5	-3	74.97	-4.97
5	Mas	80	5	3	-	
6	P.hD	100	25	20	-	

Next decision tree



→ We will continue to create decision tree

#100DAYSOFML

Machine Learning Algorithms

10.XG Boost

Sameer Kulkarni
sameerk126

Mathematical representation $\boxed{\alpha_0 = 1}$

$$F(x) = \alpha_0 h_0(x) + \alpha_1 h_1(x) + \alpha_2 h_2(x) + \cdots + \alpha_n h_n(x)$$

$h_0(x)$ = Base model $\{\alpha_0, \alpha_1 \cdots, \alpha_n\}$ - learning rate

$h_1(x)$ = Decision tree $\alpha = [0 \text{ to } 1]$

Xgboost Classification Algorithm

Dataset

x_1	x_2	y	Error $(y - 0.5)$ ← Step 1
Salary	Credit	Approval	R_1
<=50K	B	0	-0.5
<=50K	G	1	0.5
<=50K	G	1	0.5
>50K	B	0	-0.5
>50K	G	1	0.5
>50K	N	1	0.5
<=50K	N	0	-0.5

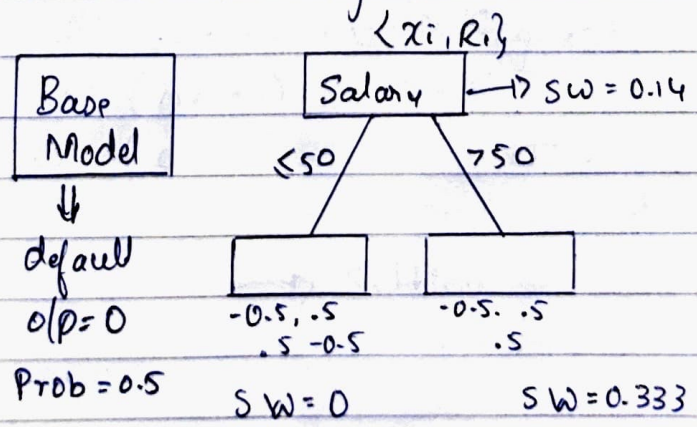
Steps

1. Construct a base model
2. Construct a decision tree with root node
3. Calculate similarity weight

$$S.W = \frac{\left(\sum \text{Residual}\right)^2}{\sum \text{Prob}(1 - \text{Prob})}$$

4. Calculate gain

Step : 1 - Constructing base model



Similarity weight (left child)

$$S.W\ (LC) = \frac{\left(\sum residual\right)^2}{\sum P_r(1-P_r)} = \frac{[-0.5+0.5+0.5+(-0.5)]}{0.5(0.5)+0.5(0.5)+0.5(0.5)+0.5(0.5)} = 0$$

S.W of Right child

$$SW\ (RC) = \frac{\left(\sum residual\right)^2}{\sum P_r(1-P_r)} = \frac{[-0.5+0.5+0.5]^2}{0.5(0.5)+0.5(0.5)+0.5(0.5)}$$

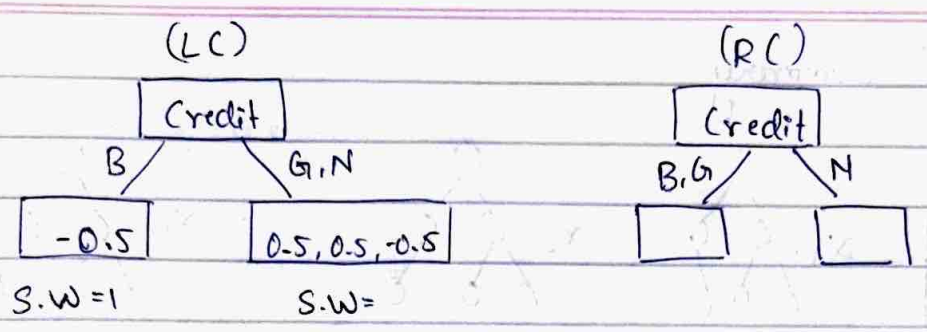
$$= \frac{0.25}{0.75} = 0.333$$

Similarity weight of root

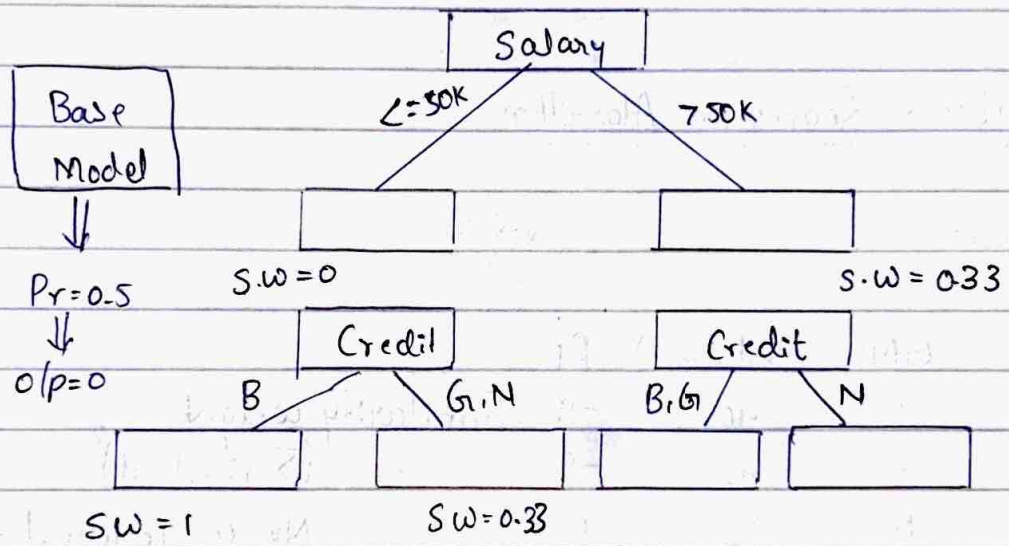
$$S.W\ (root) = \frac{0.25}{1.75} = 0.14$$

Calculating gain

$$gain = 0 + 0.33 - 0.14 = \cancel{0.21}$$
$$= 0.19$$



Final output for classification problem



log odds formula

$$\log(\text{odds}) = \log\left(\frac{p}{1-p}\right) = \log\left(\frac{0.5}{0.5}\right) = 0$$

$\log(1) = 0$

$\alpha \rightarrow$ learning rate

$\alpha = [0 \text{ to } 1]$

Base model $\rightarrow$ Test data o/p $= 0 + \alpha(1)$

$\sigma(0 + \alpha(1)) \longrightarrow$ Sigmoid activation function

$\sigma(0 + 0.1)$

$\sigma(0.1)$

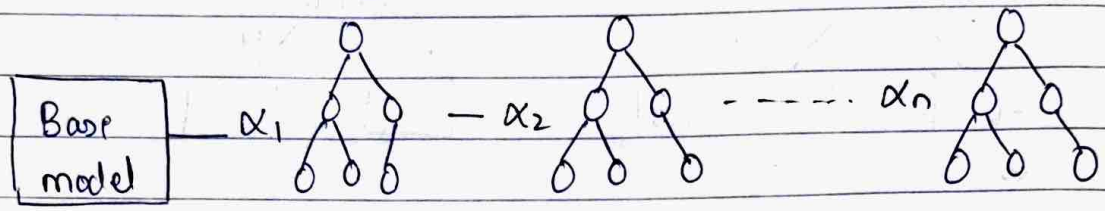
$$\sigma = \frac{1}{1 + e^{-z}}$$

o/p $= 0.52 \Longrightarrow$ Setting threshold $= 0.6$

threshold is set by Domain expert

$\boxed{0.52 < 0.6} \Longrightarrow 0$

Xgboost Summary



$$o/p = \sigma\left(\text{Base learner} + \alpha_1 (DT_1) + \alpha_2 (DT_2) + \cdots + \alpha_n (DT_n)\right)$$

Xgboost Regressor Algorithm

Dataset

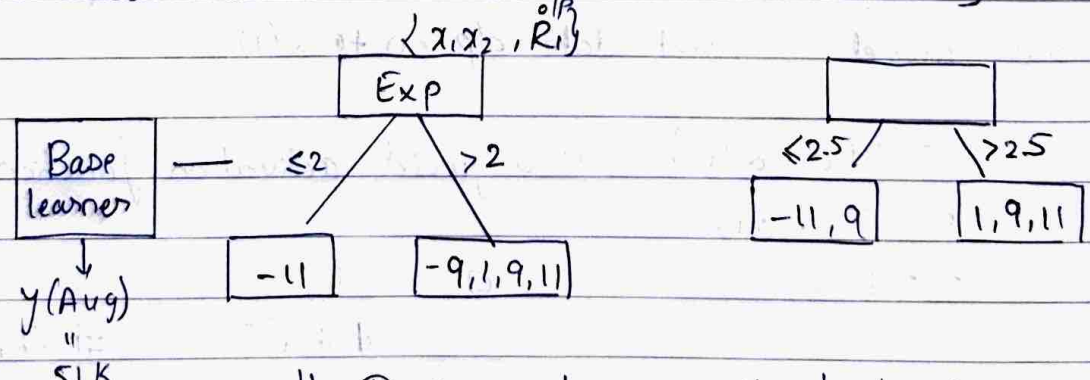
Exp	GAP	Salary(k)	R_1
2	Y	40	-11
2.5	Y	42	-9
3	N	52	1
4	N	60	9K
4.5	Y	62	11
		Avg = 51	

Similarity weight $= \dfrac{(\sum \text{residual})^2}{\text{No. of residual} + \lambda}$

$\lambda \rightarrow$ Hyperparameter

Steps

① Create a base model $[-11, -9, 1, 9, 11]$



Decision tree with high info gain is selected

Similarity Weight $= \dfrac{(\sum \text{Residual})^2}{\text{No. of residual} + \lambda} = \dfrac{121}{1+1} = 65.5$

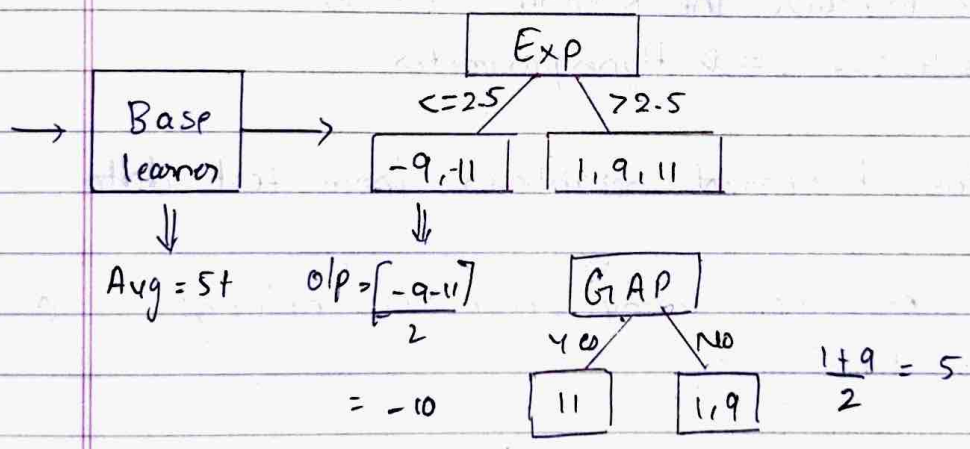
S.w (Right child)

$= \dfrac{(-9+1+9+11)^2}{4+1} = \dfrac{144}{5} = 28.5$

LC RC Root

Calculating gain $= 65.5 + 28.5 - 0.16 = 98.34$

Similarly gain of second DT is also calculated, DT with high gain is selected



Test data [2.1, Yes]

$51 + \alpha(-10)$

$= 51 + 0.1$

$= 50.9$

α - learning rate $= 0.1$

New data [2.7, No]

$= 51 + 0.1(5) = 51.5$

This process continues till the predicted value is close enough to real value

#100DAYSOFML

Machine Learning Algorithms

11.K Nearest Neighbour

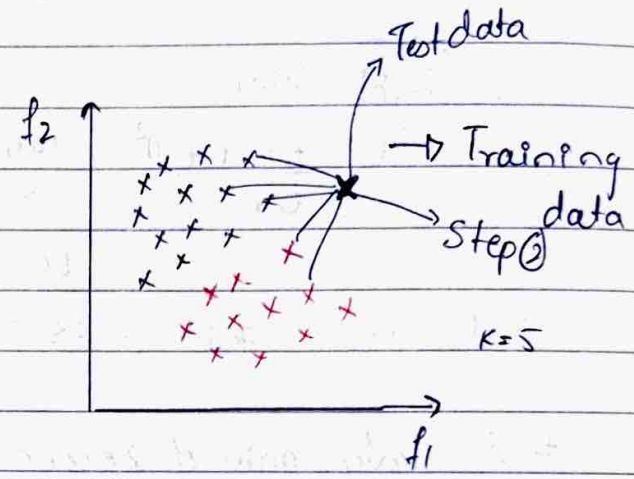
Sameer Kulkarni
sameerk126

K Nearest Neighbour (KNN)

Classification | Regression

I Classification

f_1	f_2	y
-	-	0
	-	1
	-	1
	-	0



① We have to initialize the K value $(K>0)$
$K = 1, 2, 3, 4, 5 \ldots \Rightarrow$ Hyperparameter

② We find the K-nearest neighbour from test data

③ From those $K=5$ how many neighbours belongs to 0 category & 1 category

3 1's 2 0's $\Rightarrow$ Predicted value is 1

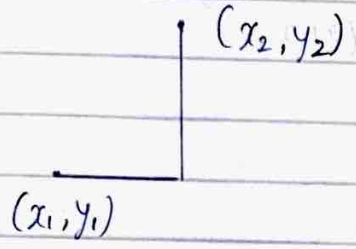
Distance Matrix → ① Eucledian distance
→ ② Manhatten distance

① Eucledian distance

2 D (x_1, y_1) — (x_2, y_2) $\quad D = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

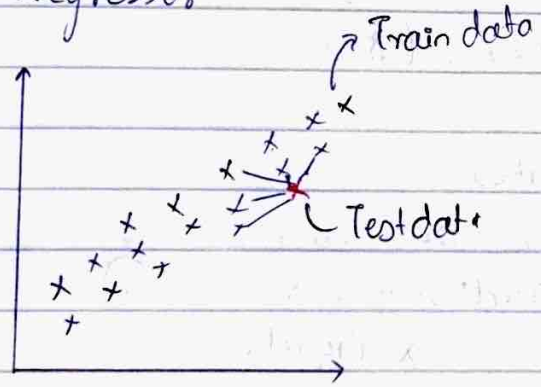
3D Distance $= \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$

② Manhatten Distance



$$M\,Dist = |x_1 - x_2| + |y_1 - y_2|$$

II Regressor



Find n nearest points of test data and find the mean of them

Variants of KNN

As we calculate distance from all points and then choose closest n points thus the time complexity is high

Time complexity = $O(n)$

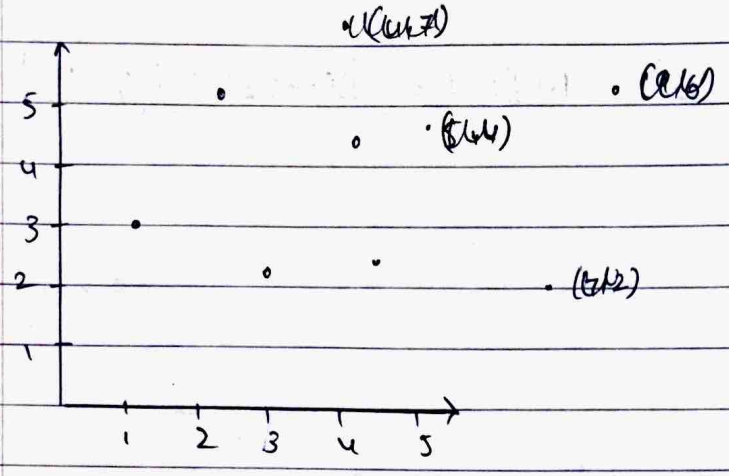
causes problems with large dataset

Alternatives

KD Tree
Ball tree } work on the concept of binary tree

Time complexity of binary tree → $O(\log(n))$

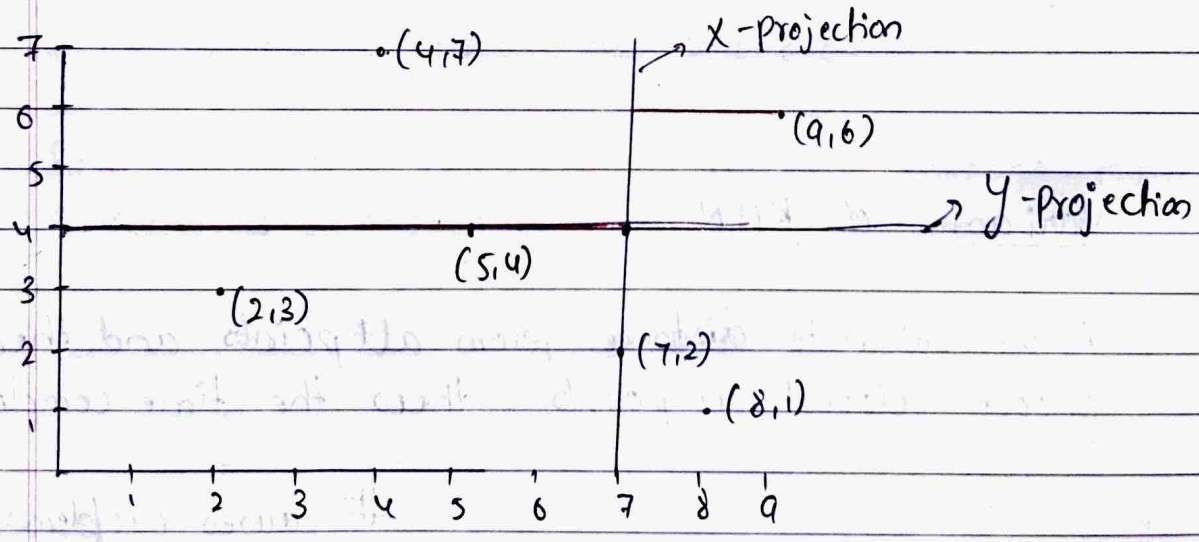
1] KD Tree (K-Dimensions Tree)



(2,3), (4,7), (5,4), (7,2), (8,1)

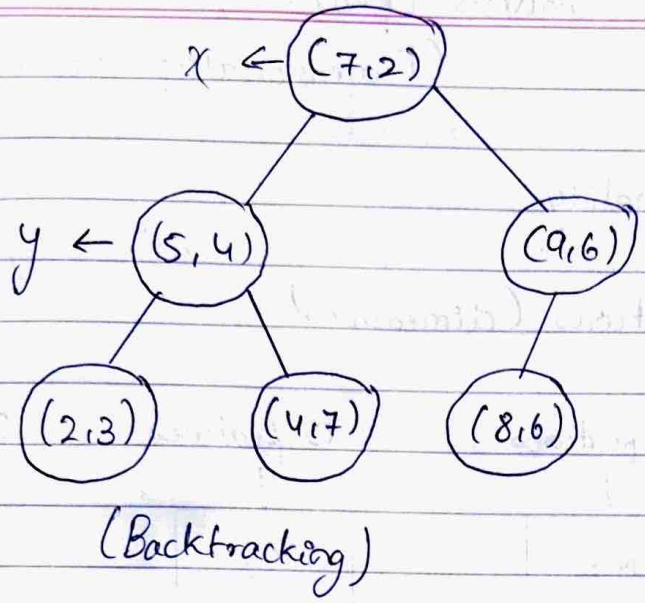
① Median of x coordinates

2, 4, $\boxed{5, 7}$, 8, 9 median → 6.5 → 7



② Median of y axis

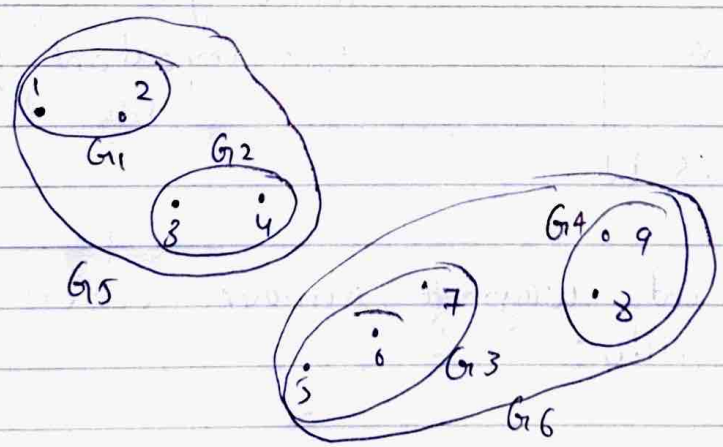
1, 2, $\boxed{3, 4}$, 6, 7 median = 3.5 → 4

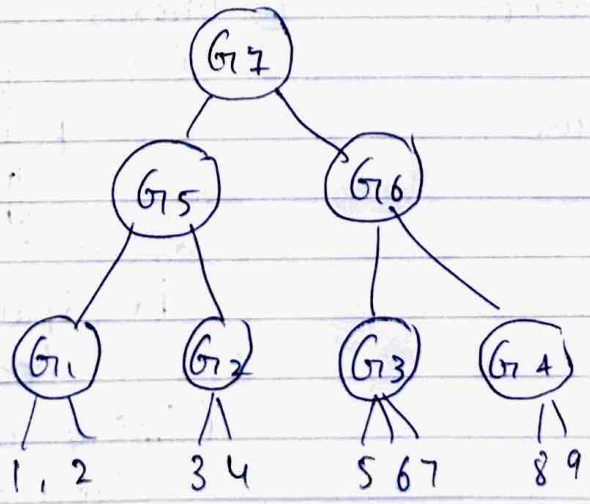


Reduces no. of comparisions
Reduces Time complexity

2] Ball tree

* Groups elements into smaller





#100DAYSOFML

Machine Learning Algorithms

12.Principle Component Analysis (PCA)

Sameer Kulkarni
sameerk126

Principal Component Analysis [PCA]

(Dimensionality Reduction)

I Curse of Dimensionality

Dataset = 500 features (dimensions)

{Model is Overfeeded}

3 features	6 features	15 features	50 features
M_1 (Model 1)	M_2	M_3	M_4
Acc_1	Acc_2 ↑↑	Acc_3 ↑↑	Acc_4 ↓↓

$Acc_1 < Acc_2 < Acc_3 > Acc_4$

100 features → M_5 → Acc_5 ↓↓

As no. of features increase model gets confused due to low important features

* Two different ways to remove curse of Dimensionality:

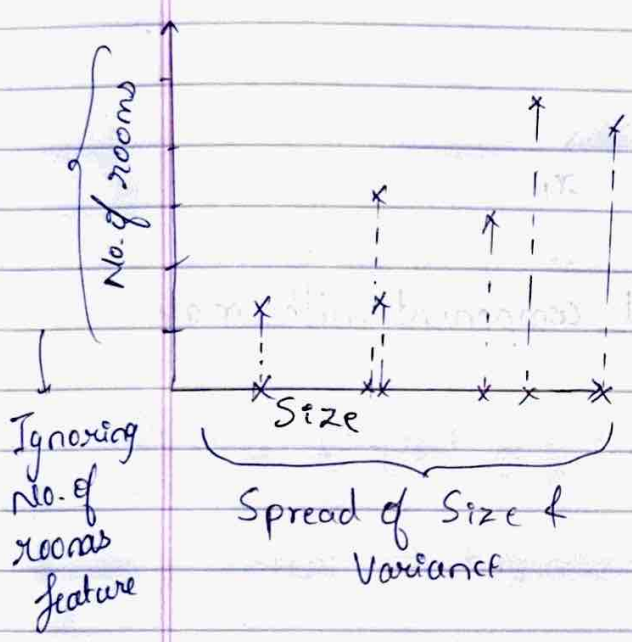
① feature selection ⇓ Imp features

② Dimensionality Reduction (PCA) ⇓ Feature extraction ⇓ Principal Component Analysis (Using eigen values & vectors)

PCA Geometric Intution [Dimensionality Reduction]

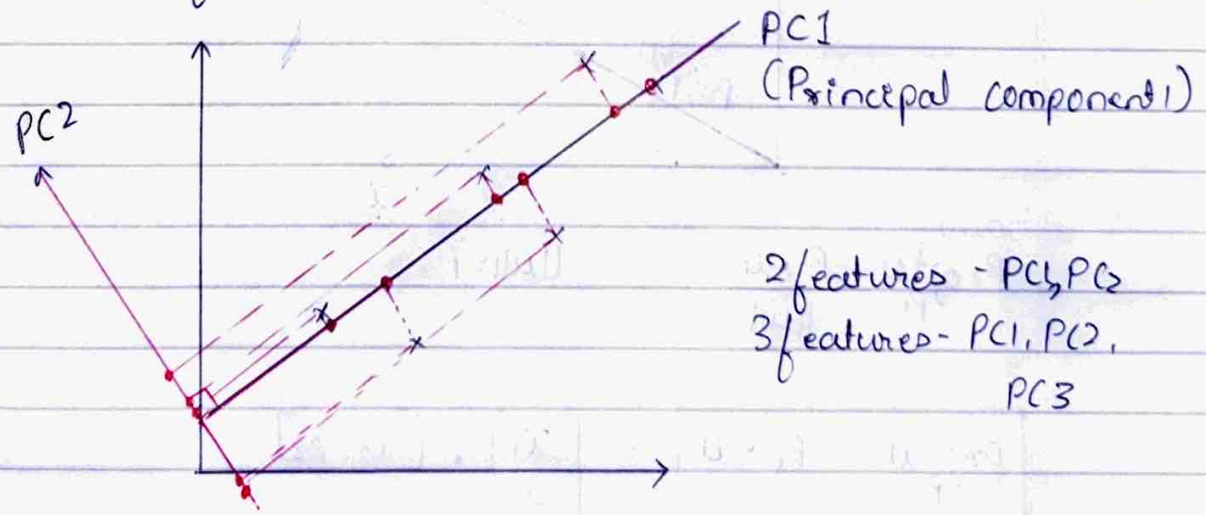
Aim → 2D → 1 dimension

x_1	x_2	o/p
Size of ~~room~~ house	No. of rooms	price



① Feature selection
⇓
2D → 1D

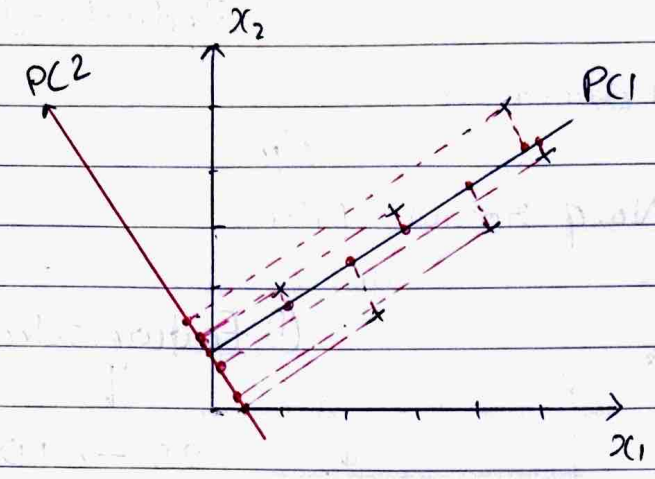
II PCA technique



Variance of PC_1 > Variance of PC_2 > Variance of PC_3 ? --

* We are extracting ~~f~~ info from ~~fre~~ features & create new features

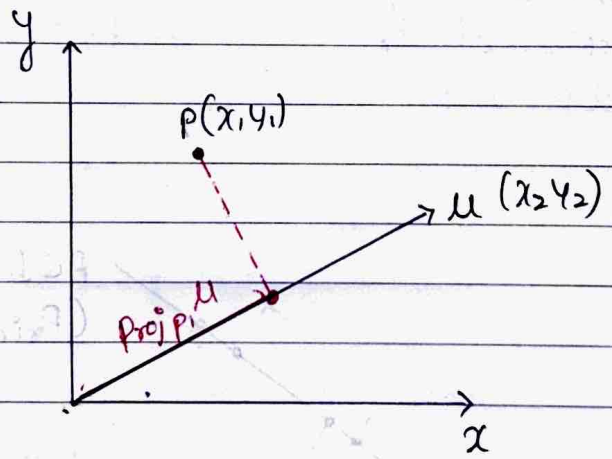
Mathematical Intution behind PCA



Variance
PC1 > PC2

① Projection
② Optimization → find principal component with max variance

① Projection



$$Proj_{p_1} u = \frac{p_1 \cdot u}{\|u\|} \qquad \|u\| = 1$$

$$\boxed{Proj_{p_1} u = p_1 \cdot u} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \cdot \begin{bmatrix} x_2 & y_2 \end{bmatrix}$$

$$= x_1 x_2 + y_1 y_2$$

$\boxed{p_0', p_1', p_2' \ldots}$ ⟹ $\boxed{\text{Scaler Value}}$ ⟹ Calculating Variance

↓

all the projections of data points

Calculating Variance

$P_0', P_1', P_2', P_3' \text{ --- } P_n'$

$x_0, x_1, x_2, x_3 \text{ -- } x_n$

Max. variance $= \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n} \Rightarrow$ Cost function

Goal = { find the best unit vector which capture maximum variance }

Q How to find the vectors?

Concept → Eigen values decomposition

Eigen values Eigen vectors

Step1 → Covariance matrix b/w features

Step 2 → Eigen value & eigen vectors will be found out using this covariance matrix

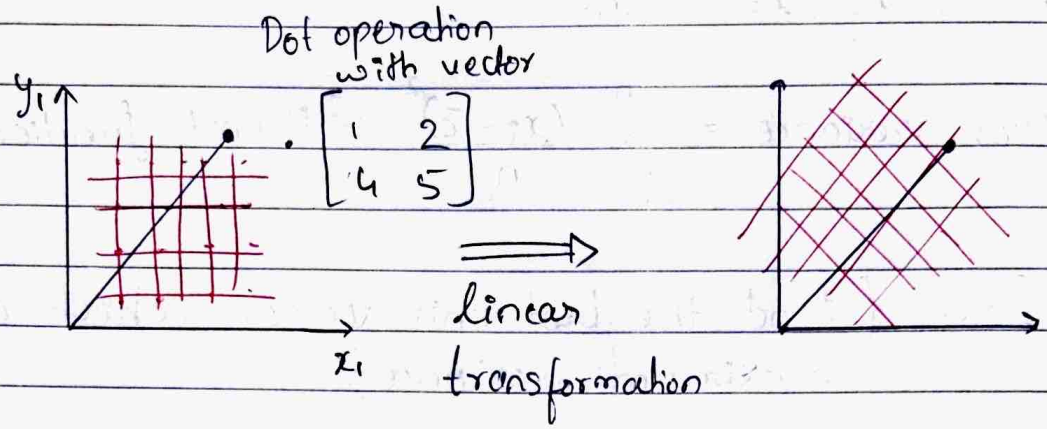
$A\lambda =$ $A v = \lambda v$

Covar matrix Covar mat

Step:3 Eigen vector → Higest Eigen value → Capturing max variance

classmate
Date
Page

* How to calculate eigen vectors & eigen values
[Linear transformation]



* Eigen vectors direction donot change after linear transformation {2D plane has 2 eigen vectors
3D plane has 3 eigen vectors -- }

$$\begin{bmatrix} \text{covariance} \\ \text{matrix} \end{bmatrix} * [\text{vector}] = \lambda * v$$

$\lambda \Downarrow$ Eigen value magnitude

Eigen vector → Max magnitude (eigen value)
⇓
PCA!

Steps to calculate eigen values & eigen vectors 2d → 1d

① Covariance of features $\quad Cov(x,y) = \sum_{i=1}^{n} \frac{(x_i-\bar{x})(y_i-\bar{y})}{N-1}$

	x	y
x		
y		

Var(x)	Cov(x,y)
Cov(x,y)	Var(y)

* Before applying PCA we have to <u>standardise</u> the dataset

③ Find out the eigen values & eigen vectors

$$A \cdot v = \lambda \cdot v$$

variance ↑↑ ⇒ PC1, PC2, PC3

#100DAYSOFML

Machine Learning Algorithms

13.K - Means Clustering

Sameer Kulkarni
sameerk126

Unsupervised Machine Learning

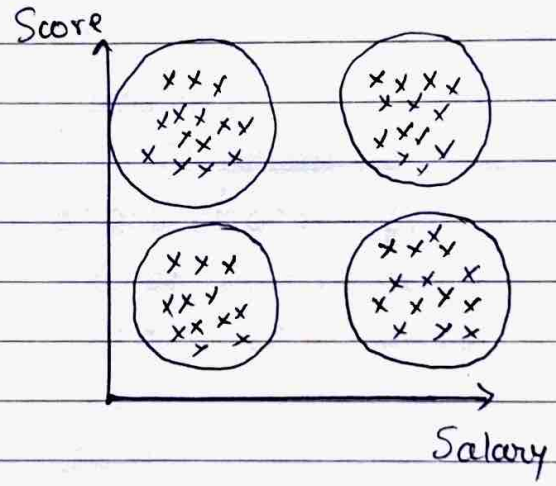
Clustering Algorithm

Unsupervised
↳ No o/p feature

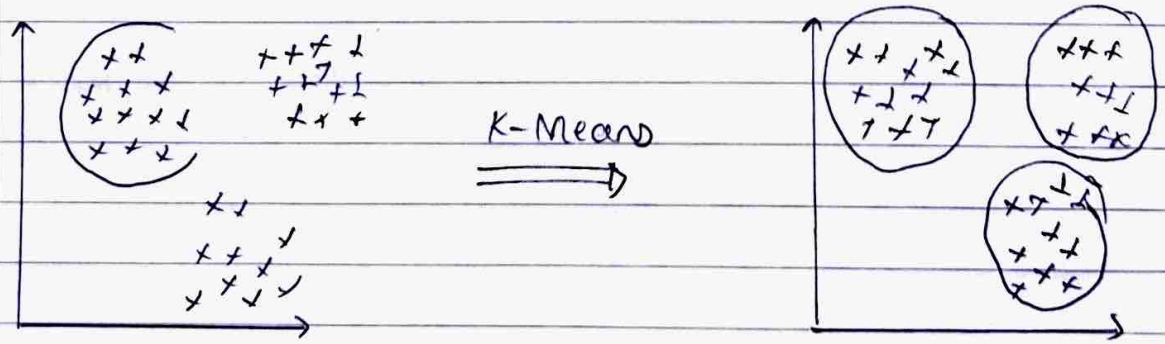
1. K-Means clustering
2. Hierarichal clustering
3. DBSCAN clustering

Dataset

Salary	Spending Score
–	–
–	–
–	–



K-Means Clustering [Geometric Intution]



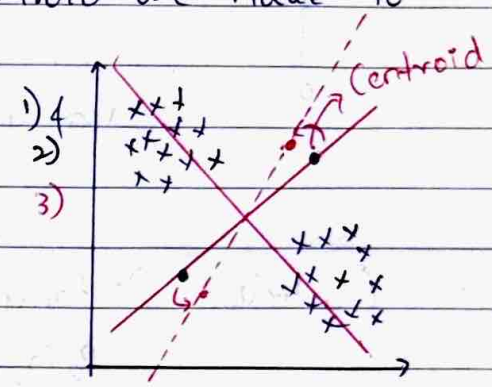
Steps:

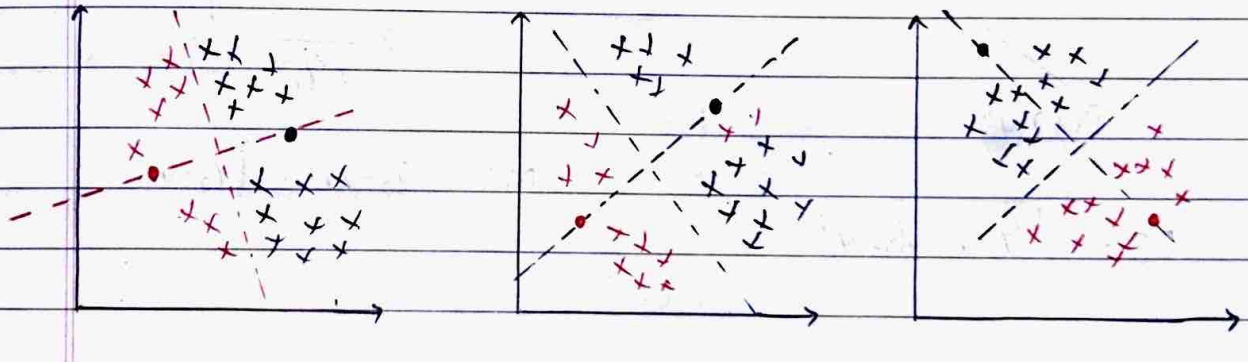
① Initialize some k- centroids {k = no. of groups}

② Points that are nearest to the centroid we have to group them

③ Move the centroids by calculating the mean.

4.) Repeat step 2

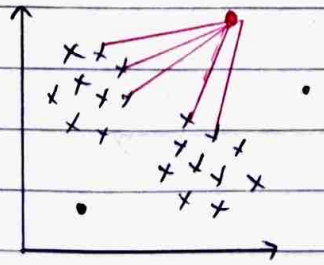




How do we select the K-value?

WCSS = Within cluster sum of squares

Initialize k = 1 to 20



$$WCSS = \sum_{i=1}^{k} \left(\text{Distance b/w points to nearest centroid}\right)$$

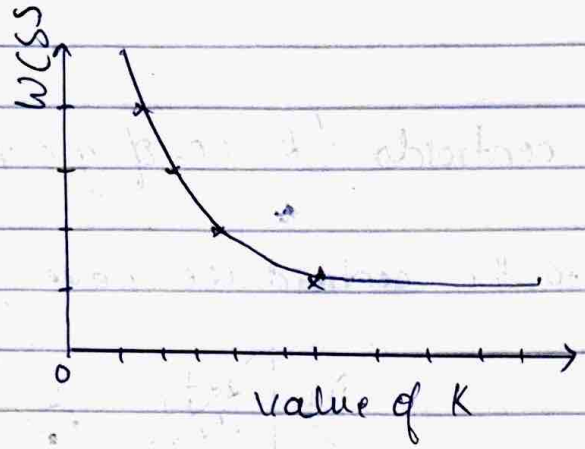
As no. of centroid increases WCSS value decreases

#100DAYSOFML

Machine Learning Algorithms

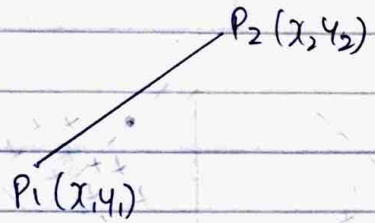
14. Hierarchial Clustering

Sameer Kulkarni
sameerk126



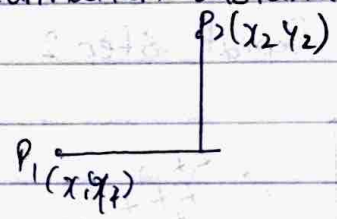
Elbow method

Euclidean Distance



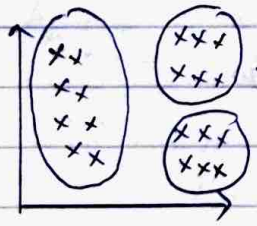
$$ED = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$$

Manhatten distance



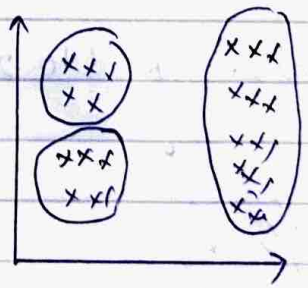
$$MD = |x_2 - x_1| + |y_2 - y_1|$$

<u>Random Initilization Trap</u> (k means)



(K means)
K=3

Problem



Problem occors due to because centroids are nearly initialized

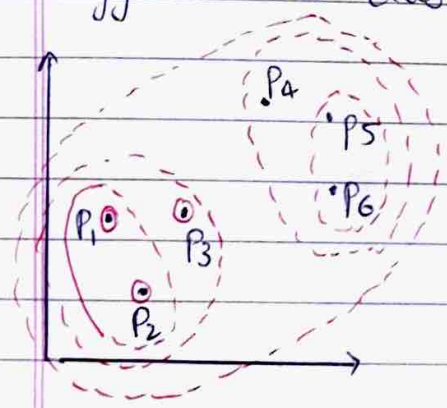
solution - k means ++

Kmeans ++ → if K=3, it makes sure the centroids are initialized with large distance

Hierarichal Clustering

① Agglomerative (combine)
② Divisive (divide)
} Geometric Intution

① Agglomerative clustering



* For each point initially we will consider it as a saperate cluster

② Find the nearest point & create a new cluster

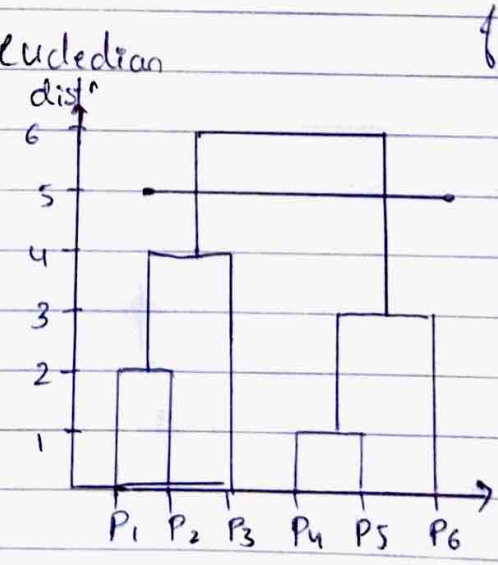
③ We keep on doing step 2 untill we get a single cluster

How many clusters?

Imp **Dendogram** :-

(connection medium)

Threshold of eucledian distance ex = 5



{threshold is 5 no. of clusters is 2

* Select the longest vertical line such that no horizontal line passes through it

K Means Vs Hierarichial Clustering

(Scalability & flexibility)

① Dataset size
- → Huge ⇒ K Means
- → Small ⇒ Hierarichial Clustering

② Type of data
- → Numerical data → Kmeans or Hierachial
- → Variety of data → Hieraichial

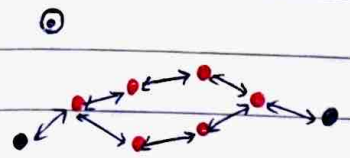
#100DAYSOFML

Machine Learning Algorithms

15. DBSCAN

Sameer Kulkarni
sameerk126

DBSCAN Clustering



• → Core point

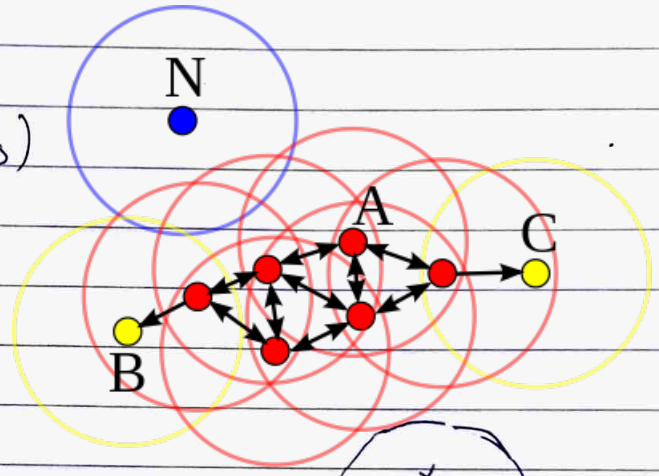
• → Border point

⊙ → Noise / outlier

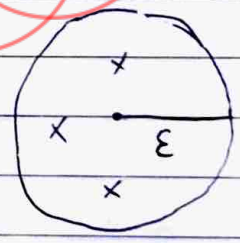
Hyperparameters

i) minimum points (minpts)

ii) ε - epsilon = radius

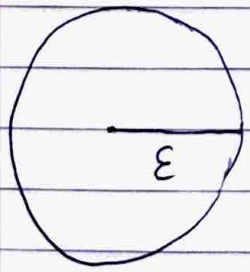


* Core point



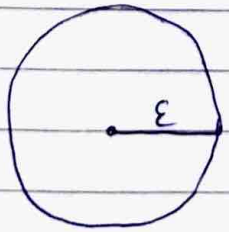
No of points within ε should be greater than or equal to minimum points

* Border points



* No. of points within radius will be less than minimum point

* Outliers [DBSCAN is Robust to outliers]



* No points within the radius