Unit – IV
Learning Machines
Decision Tree, Naïve Bayes, KNN algorithm, SVM algorithm, Logistic Regression

- All these can be used for both classification and regression.
- All these are supervised learning methods
- Training algorithm is simple
- All these machines are sensitive to errors in data as well as missing data
- Data cleaning is assumed before training
- Data is assumed to be normalized
- However, these algorithms are more popular for classification
- We focus on classification
- Different algorithms are used for each of these Learning Machines
- We study simple use case

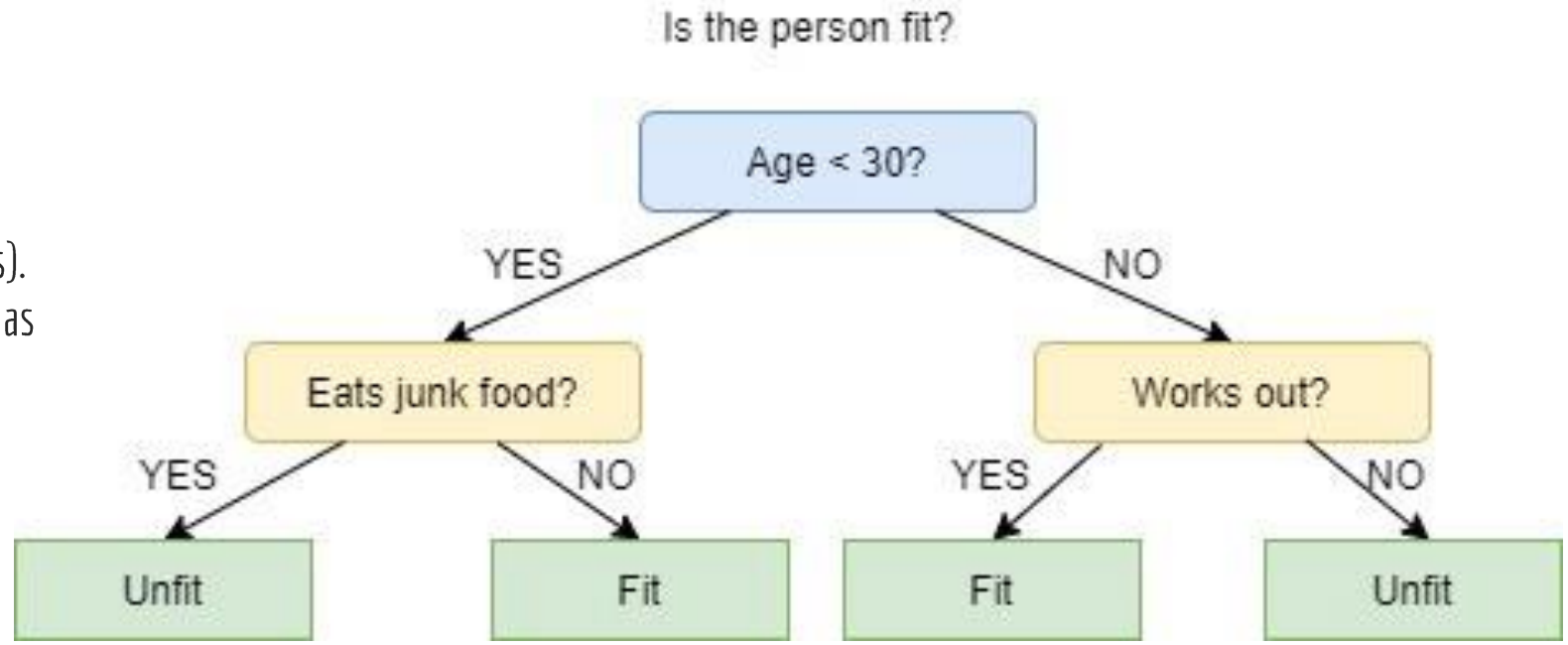
Decision Tree - Theory

Q. What is Decision Tree

A.

- is a structure that contains nodes and edges
- built from a dataset (table of columns representing features/attributes and rows corresponds to records).
- Each node is either used to ***make a decision*** (known as decision node)
- or ***represent an outcome*** (known as leaf node).
- Each Node represents a random variable

Ex. Decide if a person is fit or not



- DT nodes are questions
- "Is the person less than 30 years of age?',
- 'Does the person eat junk?',
- the leaves are one of the two possible outcomes viz. Fit and Unfit.

DT is a supervised ML algorithm

DT can be used for both Classification and Regression problerms

- The initial node is called the root node
- the final nodes are called the leaf nodes
- the rest of the nodes are called intermediate or internal nodes.
- The root and intermediate nodes represent the decisions
- the leaf nodes represent the outcomes.

Q. What is entropy in DT?

- Entropy is a measure of randomness in the given data.
- Higher the entropy, harder to draw any conclusion from data

Formula for entropy:

$H(X) = -\sum_1^n P(x_i) log_2(x_i)$

Q. What is Information Gain in DT?

- Measure of amount of information gained about a random variable from observing another random variable.
- Can be considered the difference between the entropy of parent node and weighted entropy of child nodes

Formula for IG:

IG(S,A) = E(S) - -$\sum_1^n P(x)$E(x)

- Information Gain calculates the reduction in the entropy and measures how well a given feature separates or classifies the target classes
- The feature with the highest Information Gain is selected as the best one.
- Q. What is ID3 algorithm?
- Iterative Dichotomiser 3
- The algorithm iteratively dichotomizes(divides) features into two or more groups at each step.
- ID3 uses a top-down greedy approach to build a decision tree.
- the top-down approach means that we start building the tree from the top
- greedy approach means that at each iteration we select the best feature at the present moment to create a node.
- Most generally ID3 is only used for classification problems with nominal features only.

Q.How is a node added in DT?

- the ID3 algorithm selects the best feature at each step while building a Decision tree.
- 'How does ID3 select the best feature?'
- Uses a metric called Information Gain or just Gain to find the best feature.
- The feature with the highest Information Gain is selected as the best one.

- In the case of binary classification (where the target column has only two types of classes) entropy is 0
- if all values in the target column are homogenous(similar) and will be 1
- if the target column has equal number values for both the classes, either is ok.

ID3 Steps

1.Calculate the Information Gain of each feature.
2.Considering that all rows don't belong to the same class, split the dataset **S** into subsets using the feature for which the Information Gain is maximum.
3.Make a decision tree node using the feature with the maximum Information gain.
4.If all rows belong to the same class, make the current node as a leaf node with the class as its label.
5.Repeat for the remaining features until we run out of all features, or the decision tree has all leaf nodes.

Decision Tree - Example

Fever	Cough	Breathing	Covid
N	N	N	N
Y	Y	Y	Y
Y	Y	N	n
Y	N	Y	Y
Y	Y	Y	Y
N	Y	N	N
Y	N	Y	y
Y	N	Y	y
N	Y	Y	y
Y	Y	N	y
N	Y	N	n
N	Y	Y	y
N	Y	Y	n
Y	Y	N	n

Data has three random variables , Fever, Cough, Breathing
Given a combination of values for these three random variables, find if a person has Covid?

first step : calculate the entropy of **S of the whole data**

$E(S) = — (8/14) * \log_2(8/14) — (6/14) * \log_2(6/14) = 0.99$

If all the values in target column are same the entropy will be zero (meaning that it has no or zero randomness).

Second : calculate IG of each random variable

Consider Fever; corresponding Sub Table will be

Fever	Covid			
		Y	N	Total
	Y	6	2	8
	N	2	4	6
Total				14

E(S,FEVER) = (8/14)[*E(6,2) +(6/14)*E(2,4)
= (8/14)*[-(6/8)log(6/8-)-(2/8)log(2/8)] + (6/14)[(-2/6)*log(2/6)-(4/6)*log(4/6)]
= **(8/14) * 0.81 - (6/14) * 0.91 = 0.8574**

IG(S, Fever) = E(S) – E(S,FEVER) = 0.99 - (8/14) * 0.81 - (6/14) * 0.91 = 0.132

E(S,Cough)

Fever	Covid			
		Y	N	Total
	Y	5	5	10
	N	3	1	4
Total				14

E(S,Cough) = (10/14)[*E(5,5) +(4/14)*E(3,1)
= (10/14)*[-(5/10)log(5/10)-(5/10)log(5/10)] + (4/14)[(-3/4)*log(3/4)-(1/4)*log(1/4)]
= **(10/14) * 1 + (4/14) * 0.60875 = 0.96671**
IG(S,Cough) = E(S) – E(S,Cough) = 0.04

E(S,Breathe)

Fever	Covid			
		Y	N	Total
	Y	7	1	8
	N	1	5	6
Total				14

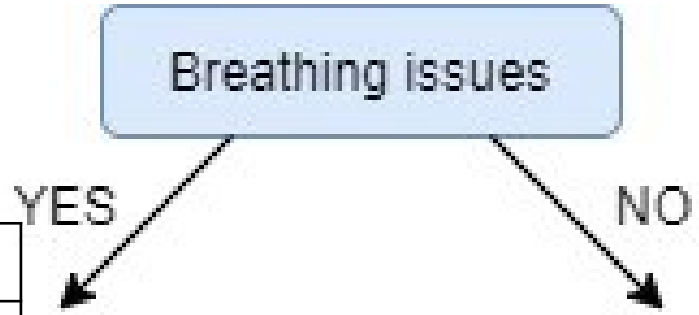
E(S,Breathe) = (8/14)[*E(7,1) +(6/14)*E(1,5)
= (8/14)*[-(7/8)log(7/8)-(1/8)log(1/8)] + (6/14)[(-1/6)*log(1/6)-(5/6)*log(5/6)]
= **(8/14) * [7/8*0.19265+3/8]+(6/14)[1/6*2.585+5/6*0.263] = 0.5891**

IG(S,Breathe) = E(S) – E(S,breathe) = 0.4

Since the feature **Breathing issues** have the highest Information Gain it is used to create the root node.

Since the feature **Breathing issues** have the highest Information Gain it is used to create the root node.

Sub Table to be considered for left branch of root node



Fever	Cough	Breathing	Covid
Y	Y	Y	Y
Y	N	Y	Y
Y	Y	Y	Y
Y	N	Y	y
Y	N	Y	y
N	Y	Y	y
N	Y	Y	y
N	Y	Y	n

Fever	Cough	Breathing	Covid
N	N	N	N
Y	Y	N	n
N	Y	N	N
Y	Y	N	y
N	Y	N	n
Y	Y	N	n

Start over again with data from left branch of breathing by first calculating entropy of breathing and information gain of the parameter fever under breathing yes.

Fever	Cough	Breathing	Covid
Y	Y	Y	Y
Y	N	Y	Y
Y	Y	Y	Y
Y	N	Y	y
Y	N	Y	y
N	Y	Y	y
N	Y	Y	y
N	Y	Y	n

E(Breathe) = — (7/8) * $\log_2$(7/8) — (1/8) * $\log_2$(1/8) = 0.5435

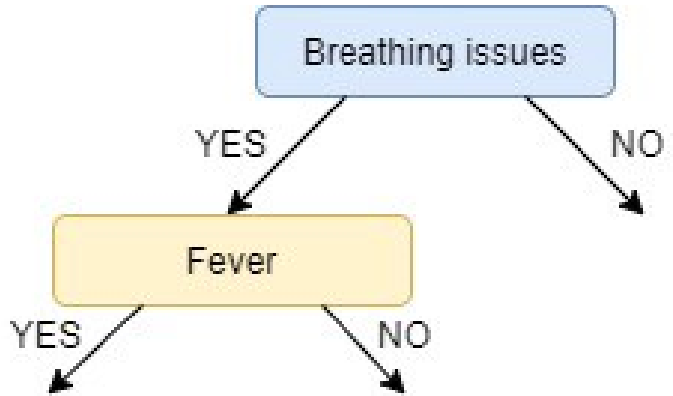
Now consider information gain of fever under breathe node

Fever	Covid			
		Y	N	Total
	Y	5	0	5
	N	2	1	3
Total				8

E(Breathe,Fever) = (5/8)[E(5,0) +(3/8)*E(2,1)
= (5/8)*[-(5/5)log(5/5)-(0/8)log(0/5)] + (3/8)[(-2/3)*log(2/3)-(1/3)*log(1/3)]
= **(3/8) * [2/3*0.585+1/3*1.5585] = 0.344**
IG(Breathe,Fever) = E(Breathe) – E(Breathe,Fever) = 0.5435 – 0.199125 = 0.2(appr.)

Now consider information gain of cough under breathe node

Fever	Covid			
		Y	N	Total
	Y	4	1	5
	N	3	0	3
Total				8



E(Breathe,cough) = (5/8)[E(4,1) +(3/8)*E(3,0)

= (5/8)*[-(4/5)log(4/5)-(1/8)log(1/5)] + (3/8)[(3/3)*log(3/3)]

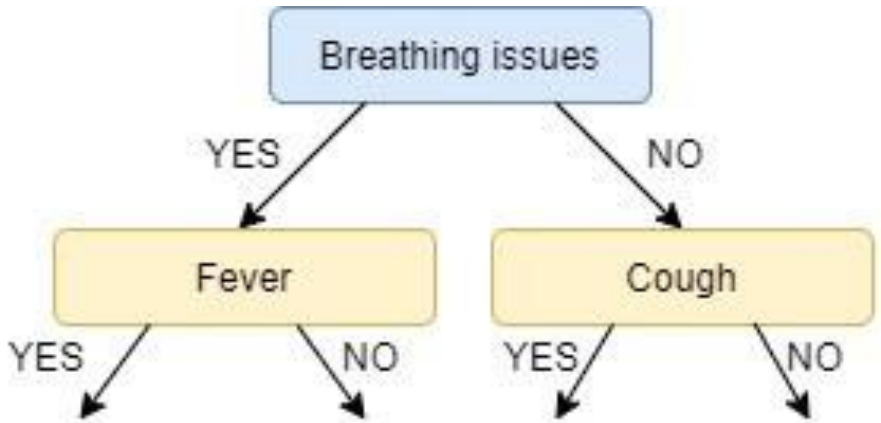
= **(5/8) * [4/5*0.3219+1/5*2.322] = 0.4512**

IG(Breathe,cough) = E(Breathe) – E(Breathe,cough) = 0.5435 – 0.4512 = 0.09(appr.)

IG(Breathe,Fever)>IG(Breathe,Cough)->Therefore nxt node is Fever

Next feature with the maximum IG for the right branch of Breathing Issues.
However, there is only one feature left; hence make it the right branch of the root node.
So the tree now looks like this:

Fever	Cough	Breathing	Covid
Y	Y	Y	Y
Y	N	Y	Y
Y	Y	Y	Y
Y	N	Y	y
Y	N	Y	y



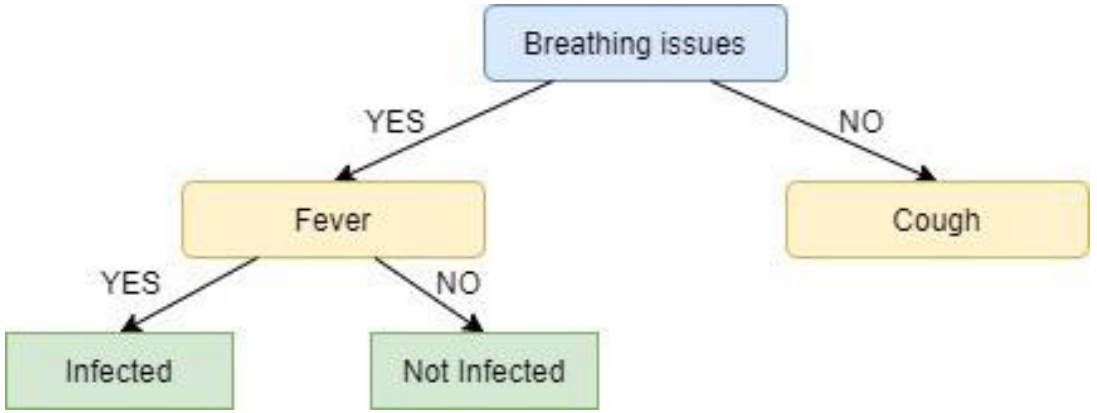
As the table shows Covid infection true whenever Fever is true,
conclude label it **Infected.**

Table for Fever not there is as below

Fever	Cough	Breathing	Covid
N	Y	Y	N
N	Y	Y	N
N	Y	Y	N

As the table shows Covid infection not there whenever Fever is not there, conclude label it **not infected** under node fever

Therefore the tree looks like



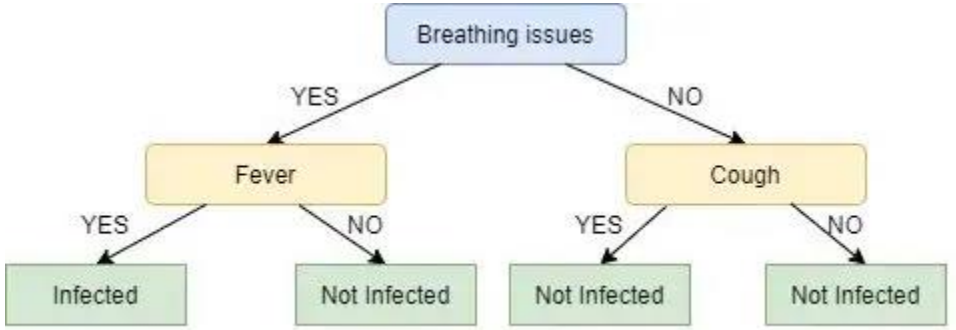
Finally, Cough node need to be expanded. From parent table, cough and breathing no results in

Cough	Breathing	Covid
N	N	N

Finally, Cough node need to be expanded. From parent table, cough and breathing yes results in

Cough	Breathing	Covid
Y	Y	Y
Y	Y	Y
Y	Y	y
Y	Y	y

The shows that whenever there is no cough and no breathing problem there is no covid infection and whenever there is cough and breathing problem there is covid. Hence final Decision tree is



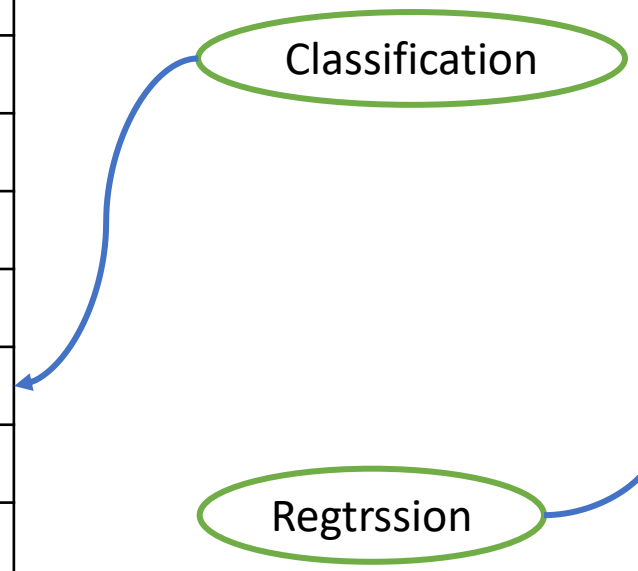
Q. Use cases of DT classifier

- Customer's willingness to purchase a given product in a given setting, i.e. offline and online both
- Product planning; for example, Gerber Products, Inc. used decision trees to decide whether to continue planning PVC for manufacturing toys or not
- Loan approval

KNN Algorithm - Theory

- K Nearest Neighbor algorithm is a Supervised Learning category
- used for classification and regression.
- General uses are for classification and imputing missing values, resampling data sets
- KNN considers K Nearest Neighbors (Data points) to predict the class or continuous value for the new Datapoint.

Age	Likes Apple Phone
30	Y
25	Y
35	N
40	Y
35	Y
40	N
50	Y
55	N
30	Y
40	Y
32	N
28	Y



Ht(ft)	Wt(kgs)
5.15	57.5
5.3	63.0
5.5	65
5.45	67.0
4.85	55.0
4.5	50.0
5.6	61.5
5.75	68.5
5.65	65.75
5.2	61.35
5.38	62.75
5.15	59.75

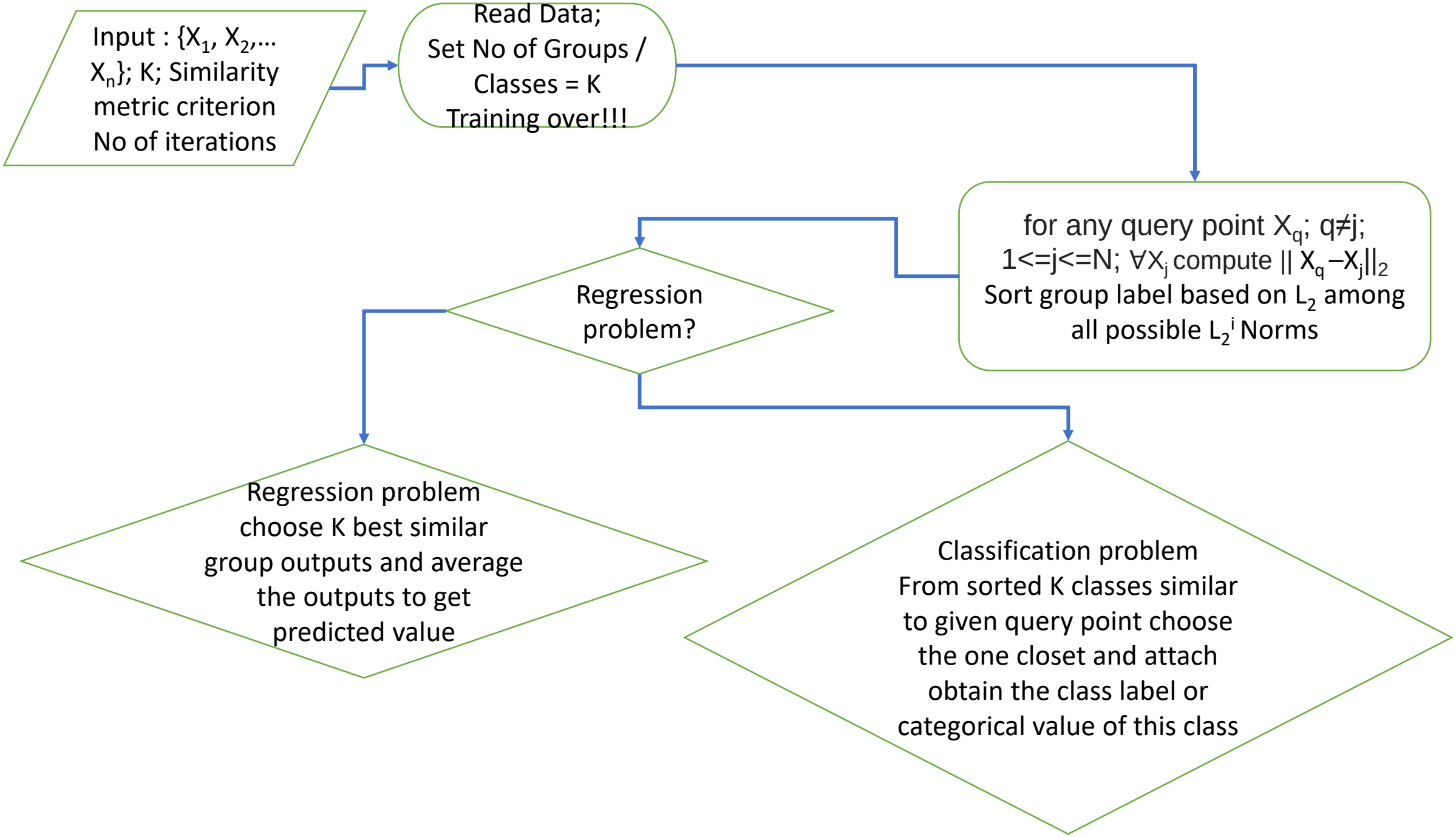
- KNN algorithm works on the principle of grouping datapoints which are similar exist in close proximity.
- Similarity metrics like Euclidean, Manhattan etc., is used.
- The algorithm is iterative
- No of classes/ groups K into which the given data is to be separated is given at start
- Testing involves just storing the data
- Testing Involves finding similarity between each training dat point and finding the nearest points similar to query point
- In case of regression average the output value belonging to k neigbhours
- In case of classification choose a class label closet to query point.

Advantages

1.The algorithm is simple and easy to implement.
2.There's no need to build a model, tune several parameters, or make additional assumptions.
3.It can be used for classification, regression, and search

Disadvantages

1.The algorithm gets significantly slower as the number of examples and/or predictors/independent variables increase.


Input : {X_1, X_2,… X_n}; K; Similarity metric criterion No of iterations
Read Data; Set No of Groups / Classes = K Training over!!!
for any query point X_q; q≠j; 1<=j<=N; ∀X_j compute || X_q –X_j||$_2$ Sort group label based on L_2 among all possible L_2^i Norms
Regression problem?
Regression problem choose K best similar group outputs and average the outputs to get predicted value
Classification problem From sorted K classes similar to given query point choose the one closet and attach obtain the class label or categorical value of this class

The KNN Algorithm

1.Load the data
2.Initialize K to your chosen number of neighbors
3. For each example in the data
3.1 Calculate the distance between the query example and the current example from the data.
3.2 Add the distance and the index of the example to an ordered collection
4. Sort the ordered collection of distances and indices from smallest to largest (in ascending order) by the distances
5. Pick the first K entries from the sorted collection
6. Get the labels of the selected K entries
7. If regression, return the mean of the K labels
8. If classification, return the majority rule of K labels

KNN Algorithm - Example

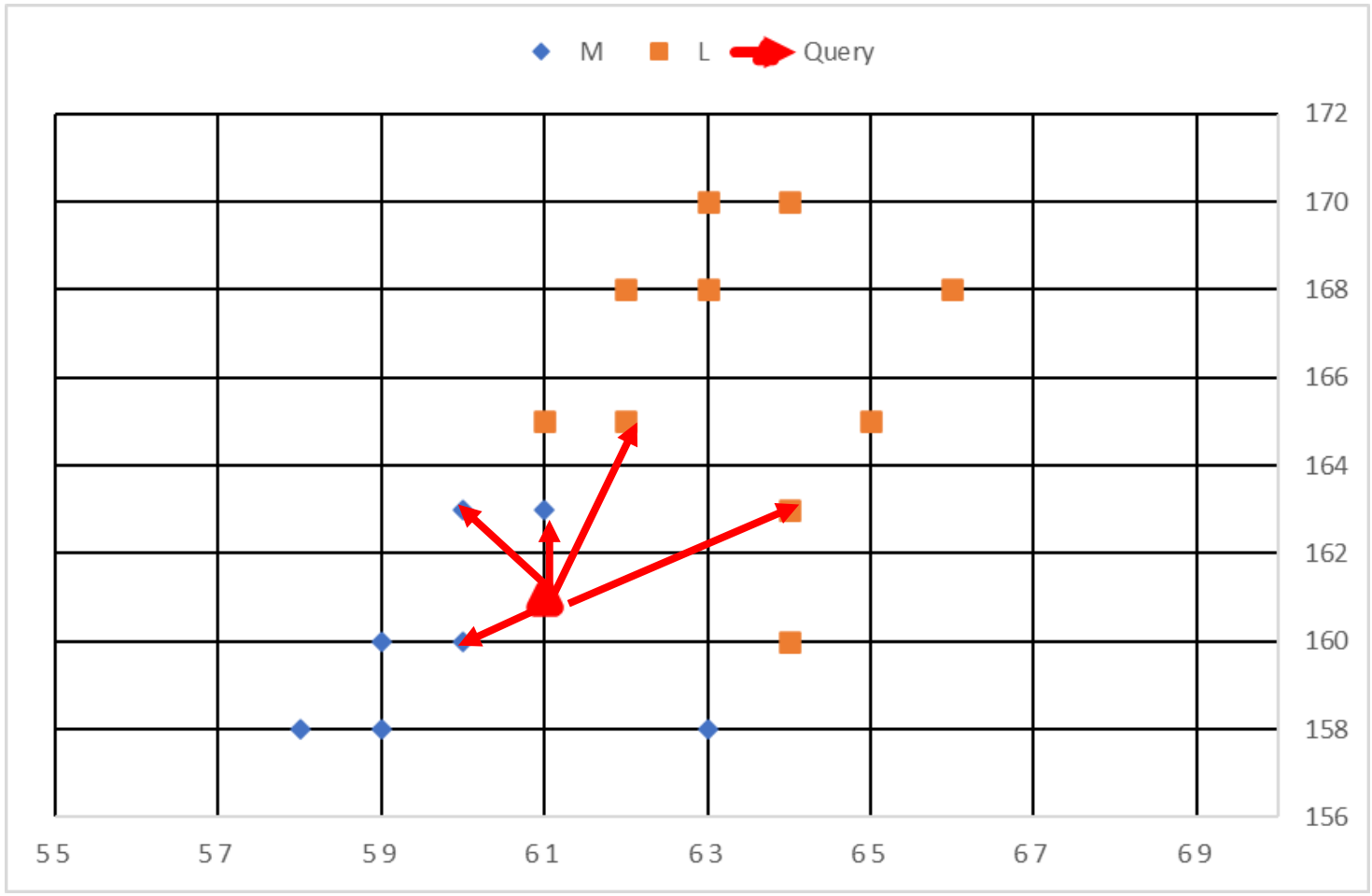
Height(cms)	Weight (kgs)	T Shirt Size
158	58	M
158	59	M
158	63	M
160	59	M
160	60	M
163	60	M
163	61	M
160	64	L
163	64	L
165	61	L
165	62	L
165	65	L
168	62	L
168	63	L
168	66	L
170	63	L
170	64	L

Query 'Geetha' height 161cm and weight 61kg, what is T shirt size
Given data table and K = 5

Height(cms)	Weight (kgs)	T Shirt Size	Distance	Nearest Neighbor
158	58	M	4.24	
158	59	M	3.61	
158	63	M	3.61	
160	59	M	2.24	3
160	60	M	1.41	1
163	60	M	2.24	3
163	61	M	2	2
160	64	L	3.16	5
163	64	L	3.61	
165	61	L	4	
165	62	L	4.12	
165	65	L	5.66	
168	62	L	7.07	
168	63	L	7.28	
168	66	L	8.6	
170	63	L	9.22	
170	64	L	9.49	

there are four data points nearest to query. Use majority rule for classification; hence the right answer is M T shirt.

KNN Algorithm – Example - Visualization



Naïve Bayes Classifier - Theory

- Naive Bayes classifier is a probabilistic machine learning model
- used mainly for classification task.
- The classifier is based on the Bayes theorem.
- It assumes the predictors/features are independent.
- presence of one particular feature does not affect the other.
- Hence it is called naive.

$$P(x|y) = \frac{P(y|x)}{P(y)}P(x)$$

Bayes Theorem

Posterior = Likelihood * Prior / Evidence

Bayes Vs Naive Bayes :

- Naïve bayes assumes conditional independence of predictor variables or evidences
- where Bayes theorem does not.

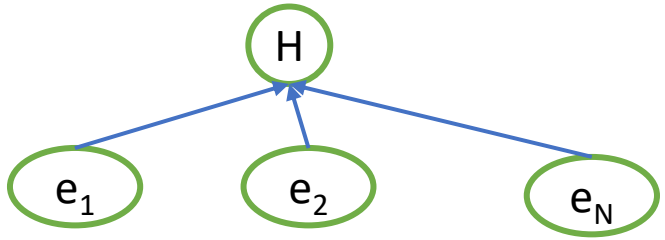
$$P(H|e_1, e_2, \ldots e_N) = \frac{P(e_1,e_2,\ldots eN \,|H)\, P(H))}{P(e_1,e_2 \ldots eN)}$$

Bayes theorem with multiple evidences and single hypothesis

$$P(H|e_1, e_2, \ldots e_N) = P(H|e_1)P(H|e_2), \ldots P(H|eN)$$
$$P(H))$$

Naïve Bayes theorem with multiple evidences and single hypothesis

- Each feature is conditioned on class label only !!!
- For each feature, need to only to know how it depends on class label
- Total number of parameters is linear in N



Algorithm

- Start probability distribution table data consisting of $P(e_i|H)$ and $P(H)$
- Compute $P(H|e_1,e_2..e_N) = P(e_1|H)\, P(e_2|H) \ldots P(e_N|H)\, P(H)$

Naïve Bayes Classifier - Example

Example : Naïve Bayes Classifier

Model : $P(H|e_1, e_2,..e_N) = P(H|e_1)P)H|e_2), ... P(H|eN)\ P(H))$

Consider sentences and the associated category to which these belong

Goal : A new sentence "A very close game"is tested to what category this belongs.

Training Data

Text	Tag
"A great game"	Sports
"The election was over"	Not sports
"Very clean match"	Sports
"A clean but forgettable game"	Sports
"It was a close election"	Not sports

- calculate the probability that the sentence "A very close game" is Sports
- and the probability that it's Not Sports.
- i.e. calculate P (Sports | a very close game)
- convert this text into numbers
- use word frequencies.

- P(sports | a very close game) = P(a very close game | sports)P(sports)
- P(sports | a very close game) = P(a very close game | -sports)P(-sports)

- P(sports | a very close game) = P(a very close game | sports)P(sports)/P(a very close game)

As per Bayes Theorem

- "A very close game" doesn't appear in training data
- so the probability is zero.

Example : Naïve Bayes Classifier – contd.

- Naïve assumption

- P(a very close game | sports)P(sports) =P(a | sports) P(very | sports) P(close | sports) P(game | sports) P(sports)
- each word can appear several times in the training data, and it is possible to calculate the individual conditional
- Step-1
- calculate the a priori probability of each word
- for a given sentence in the training data, P (Sports) = ⅗; P (Not Sports) = ⅖.
- Step-2
- P (game | Sports) = no of times the word "game" appears in Sports texts divided by the total number of words in sports = 2/11
- Step-3
- P(close | game) = 0 as close doesn't appear in sports
- To overcome 0; we have to use smoothing function; if not 0 we can straight use the approach

- Note : if all the conditional probability terms are not zero, no need of applying smoothing.
- This example is shown to demonstrate smoothing requirement.
- Smoothing can be applied in different ways, caution should be exercised that whenever numerator is adjusted to avoid 0, ratio has to be ‹ 1 due to definition of probability is in the range (0-1)

Smoothing function

- add 1 to every count so it's never zero.
- we should also add the number of possible words to the divisor, so the division will never be greater than 1 !!

'a', 'great', 'very', 'over', 'it', 'but', 'game', 'election', 'clean', 'close', 'the', 'was', 'forgettable', 'match'

- number of words = 14;
- apply smoothing
- P(game | sports) = (2+1)/(11+14)

Word	P (word | Sports)	P (word | ~Sports)
a	(2 + 1) ÷ (11 + 14)	(1 + 1) ÷ (9 + 14)
very	(1 + 1) ÷ (11 + 14)	(0 + 1) ÷ (9 + 14)
close	(0 + 1) ÷ (11 + 14)	(1 + 1) ÷ (9 + 14)
game	(2 + 1) ÷ (11 + 14)	(0 + 1) ÷ (9 + 14)

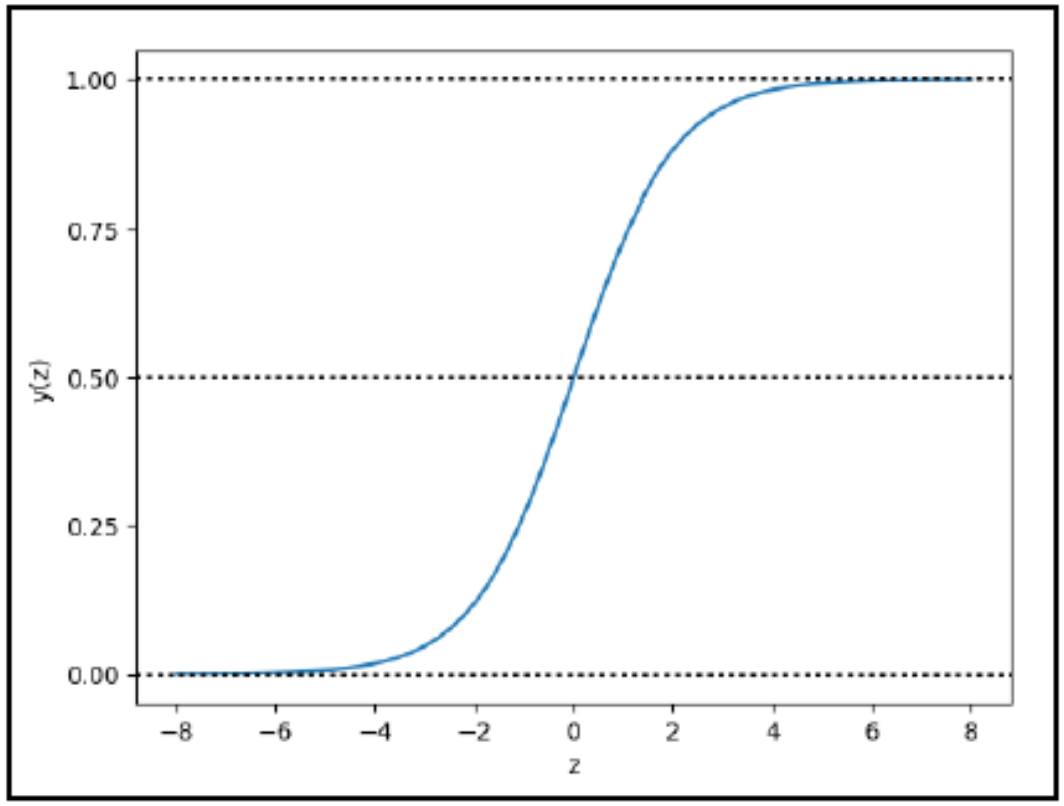
- P(a | sports) P(very | sports) P(close | sports) P(game | sports) P(sports) = (3/25)(2/25)(1/25)(3/25)(3/5) = $2.76*10^{-5}$
- P(a | ~sports) P(very | ~sports) P(close | ~sports) P(game | ~sports) P(~sports) = (2/23)(1/23)(2/23)(1/23)(2/5) = $0.572*10^{-5}$
- classifier identifies "A very close game" the Sports tag !!

Logistic Regression - Theory

$$\tilde{Y} = \frac{1}{(1 + e^{-X})}$$

Sigmoid Function

- Input range : $(-\infty - +\infty)$
- Output range (0-1)
- +ve inputs result in an output (0.5 – 1)
- -Ve inputs result in an output (-0.5 – 0.0)



Consider data points $X = (x_1, x_2 \ldots x_n)$; x_i is i^{th} feature

Consider a model $z = \theta_1 + \theta_2 x_1 + \theta_3 x_2 + \ldots + \theta_{N+1} x_N = X^T \theta$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_i \\ \vdots \\ x_N \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \vdots \\ \vdots \\ \theta_{N+1} \end{bmatrix}$$

If we define $\tilde{Y} = \frac{1}{(1+e^{-z})} = \frac{1}{(1+e^{-(X^T \theta)})}$

$\tilde{Y}$ shall be in the range (0 -1)

$\tilde{Y} = P(y|X)$

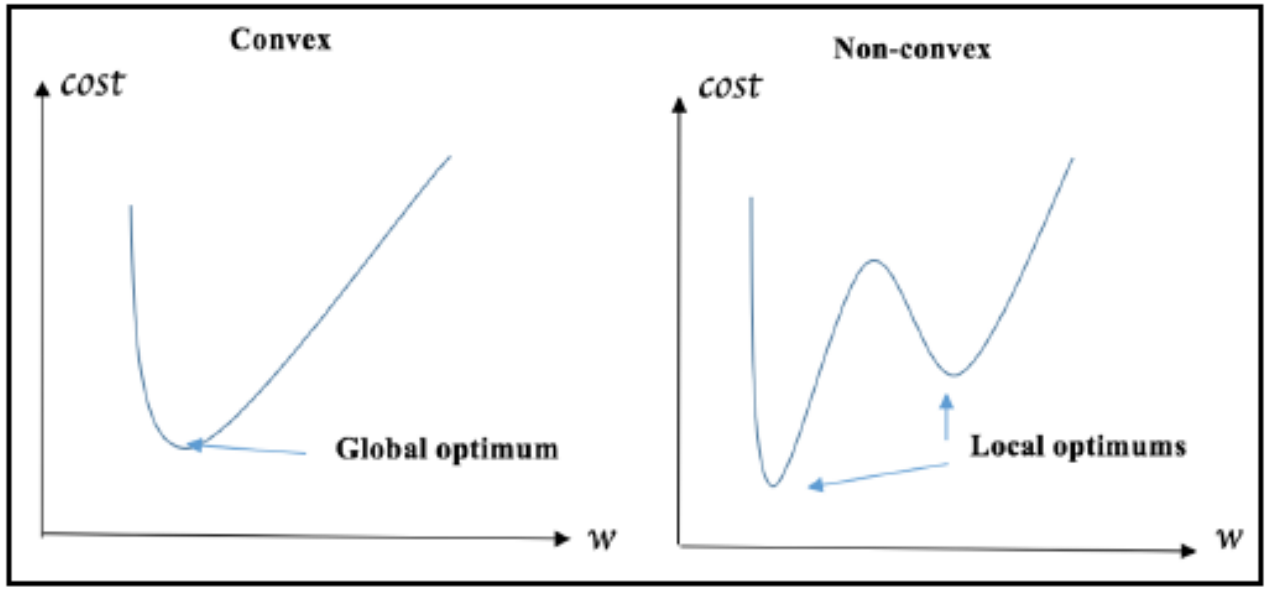
If we consider $\tilde{Y} > 0.5$ is 1 and $\tilde{Y} < 0.5 = 0$; then this expression can be used as a binary classifier

- In general the above expression for $\tilde{Y}$ is similar to conditional probability and similar to Naïve bayes classifier
- This is called logistic regression for classification.

- How do we get model parameter θ in logistic regression for classification?
- Consider binary classification, then $Y = \begin{bmatrix} 1 \\ 0 \\ . \\ . \\ 1 \end{bmatrix}$
- 1 represents Class A and 0 represents Class B
- As seen in Linear regression Model the cost / error function between Model prediction and actual target class is

$$\varepsilon = (\tilde{Y}\text{-}Y) \text{ and } J = \varepsilon^T\varepsilon = (\tilde{Y} - Y)^T(\tilde{Y}\text{-}Y)$$

- However, J can result in non-convex function resulting in inaccurate classification
- One way to overcome the problem of non-convexity is to use different cost function!!



- Use binary cross entropy loss function instead of MSE
- This loss function works only for binary classification

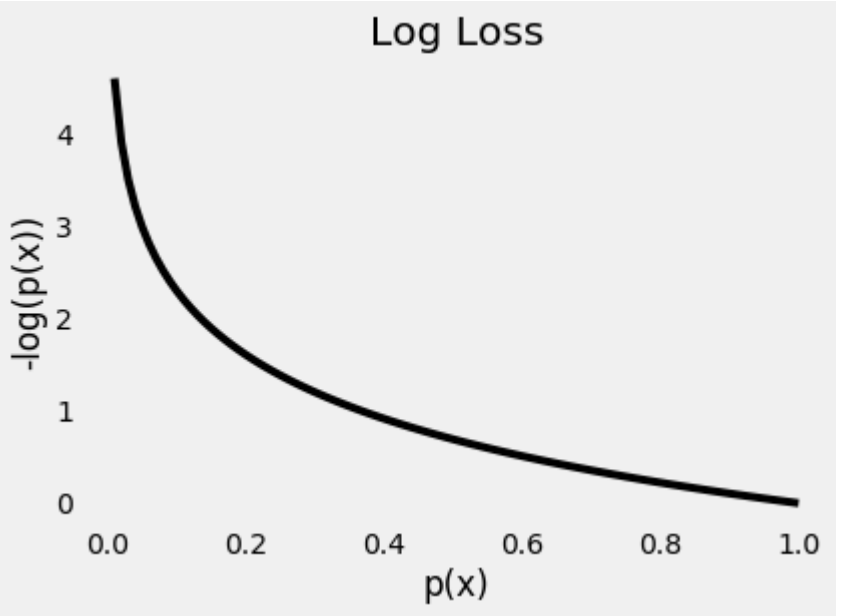
- $J = -[y_i \log(\tilde{Y}_i) + (1-y_i)\log((1-\tilde{Y}_i)]$

- The binary cross entropy has two components

$-[y_i \log(\tilde{Y}_i)]$

$-(1-y_i)\log((1-\tilde{Y}_i)]$

- Each of these one class!!



$$X = \begin{bmatrix} 1\ x_{11} .. x_{1n} \\ 1\ x_{21} .. x_{2n} \\ \\ 1\ \dot{x}_{i1} .. \dot{x}_{in} \\ 1\ x_{m1} .. x_{mn} \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \cdot \\ \cdot \\ \dot{\theta}_n \end{bmatrix} \quad Z = (\theta_0 + \theta_1 x_{i1} + \theta_2 x_{i2} + \cdots \theta_n x_{in}) = X\theta \qquad \tilde{Y}_i = \frac{1}{(1+e^{-Z})} = \frac{1}{(1+e^{-(\theta_0+\theta_1 x_{i1} + \theta_2 x_{i2} + \cdots \theta_n x_{in})})}$$

$$\frac{\partial Z}{\partial \theta} = \frac{\partial X\theta}{\partial \theta} = XT$$

$$J = [y_i \log(\tilde{Y}_i) + (1-y_i)\log((1-\tilde{Y}_i)]$$

$$\frac{\partial y_i \log(\tilde{Y}_i)}{\partial \theta_0} = \frac{y_i}{\tilde{Y}_i}\frac{\partial(\tilde{Y}_i)}{\partial \theta_0} = \left(\frac{y_i}{\tilde{Y}_i}\right)\frac{\partial\left(\frac{1}{(1+e^{-Z})}\right)}{\partial \theta_0} = \left(\frac{y_i}{\tilde{Y}_i}\right)\frac{1}{(1+e^{-Z})^2}\frac{\partial(1+e^{-Z})}{\partial \theta_0} = y_i \frac{(1+e^{-Z})}{(1+e^{-Z})^2}\frac{\partial(1+e^{-Z})}{\partial \theta_0} =$$

$$y_i \frac{1}{(1+e^{-Z})} e^{-Z}\frac{\partial(Z)}{\partial \theta_0} = y_i \frac{1}{(1+e^{-Z})} e^{-Z}\frac{\partial(\theta_0+\theta_1 x_{i1} + \theta_2 x_{i2} + \cdots \theta_n x_{in})}{\partial \theta_0} = y_i \frac{1}{(1+e^{-Z})} e^{-Z}$$

$$\frac{\partial(1-y_i)(1-\log(\tilde{Y}_i))}{\partial \theta_0} = \frac{(1-y_i)}{(1-\tilde{Y}_i)}\frac{\partial(\tilde{Y}_i)}{\partial \theta_0} = \left(\frac{(1-y_i)}{(1-\tilde{Y}_i)}\right)\frac{\partial\left(\frac{1}{(1+e^{-Z})}\right)}{\partial \theta_0} = \frac{-(1-yi)}{(1+e^{-Z})}$$

$$\frac{\partial y_i \log(\tilde{Y}_i)}{\partial \theta_1} = x_1 y_i \frac{1}{(1+e^{-Z})} e^{-Z} \qquad \frac{\partial(1-y_i)(1-\log(\tilde{Y}_i))}{\partial \theta_1} = \frac{-(1-yi)x_1}{(1+e^{-Z})}$$

$$\frac{\partial J}{\partial \theta_0} = \frac{\partial[yi\log(\tilde{Y}_i) + (1-yi)\log((1-\tilde{Y}_i)]}{\partial \theta_0} = y_i \frac{1}{(1+e^{-Z})} e^{-Z} \quad - \frac{(1-yi)}{(1+e^{-Z})} = \frac{y_I\, e^{-Z} - 1 + y_i}{(1+e^{-Z})} = \frac{y_I\,(e^{-Z}+1) - 1}{(1+e^{-Z})} = y_I - \frac{1}{(1+e^{-Z})} = (y_I - \tilde{Y}_i)$$

$$\frac{\partial J}{\partial \theta_0} = (y_I - \tilde{Y}) \qquad \frac{\partial J}{\partial \theta_1} = (y_I - \tilde{Y}_i)\, x_1 \qquad \frac{\partial J}{\partial \theta} = (y - \tilde{Y})^T X$$

- Parameter update rule during training the model
  - $\theta^{new} = \theta^{old} - \eta \frac{\partial J}{\partial \theta} = \theta^{old} - \eta(y - \tilde{Y})^T X$
- Final classification used for testing and further prediction
  - $\tilde{Y}_i = \frac{1}{(1+e^{-(\theta_0+\theta_1 x_{i1}+\theta_2 x_{i2}+\cdots+\theta_n x_{in})})}$
  - $if \tilde{Y}_i < 0.5$ class A else if $\tilde{Y}_i > 0.5$ $Class\ B$
- The above works for binary classification only

Logistic Regression - Example
Binary classification

Given text :

It's *boring* . There are virtually no surprises , and the writing is **second-rate** . So why was it so *enjoyable* ? For one thing , the cast is *great* . Another *nice t*ouch is the music . *I* was overcome with the urge to get off the couch and start dancing . It sucked *me* in , and it'll do the same to *you* .

Features in Text

- Positive sense word count (X1-Green) = 3
- Negative sense word count (X2-red) = 2
- 'No' word Count (X3-yellow) = 1
- Pronoun word count (X4-blue) = 3
- Symbols like !,", etc. (X5) = 0
- Log of toral word count (X6)= ln(57) = 4.04
- Feature vector x = (1,X1,X2,X3,X4,X5,X6)

- Assume we are given parameter values $[\theta_0\ \theta_1\ \theta_2\ \theta_3\ \theta_4\ \theta_5\ \theta_5]$ = [0.1,2.5,-5.0,-1.2,0.5,2.0,0.7]
- These values are usually obtained by training the model using gradient descent

- Given these 6 features and the input Text,
- P(+|x) and P(-|x) is computed using

$P(y = 1) = 1 / (1+\exp(-(\theta \cdot x)))$

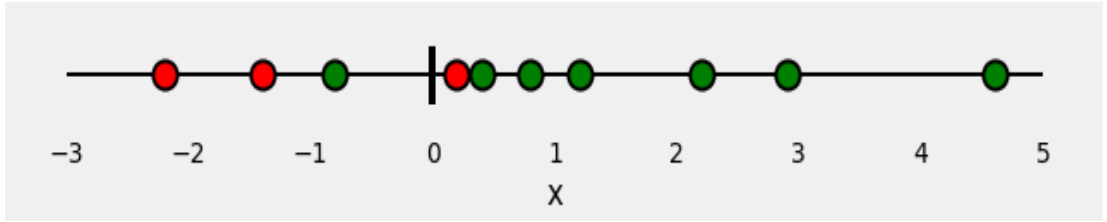
$P(y = 0) = 1 - (1 / (1+\exp(-(\theta \cdot x))))$

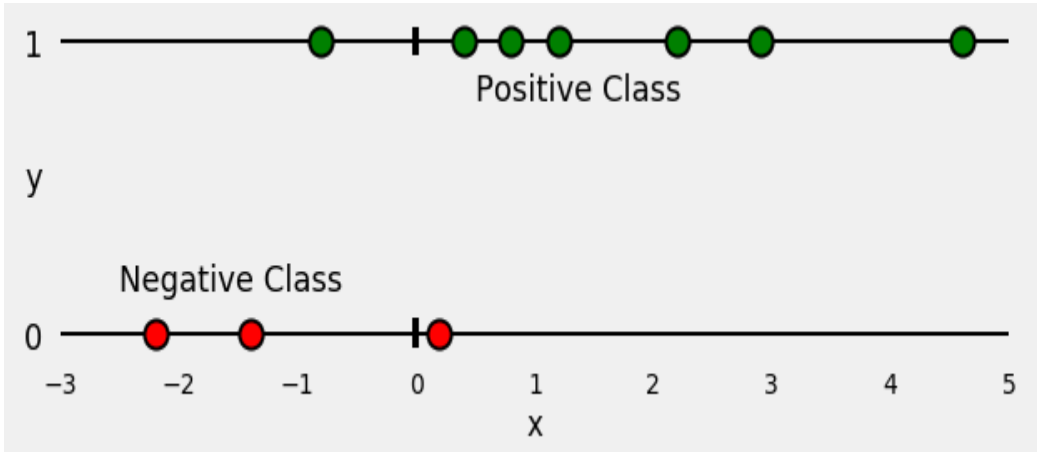
$$\text{decision}(x) = \begin{cases} 1 & \text{if } P(y = 1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

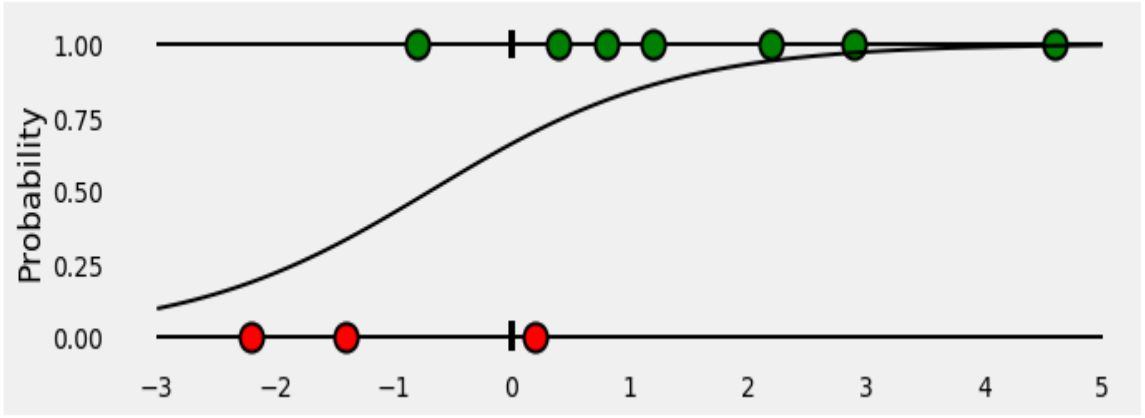
$$p(+|x) = P(y = 1|x) = \sigma(\theta \cdot x) = \sigma([0.1\ 2.5,-5.0,-1.2,0.5,2.0,0.7] \cdot [1,3,2,1,3,0,4.04]) = \sigma(.833) = 0.70$$
$$p(-|x) = P(y = 0|x) = 1-\sigma(\theta \cdot x) = 0.30$$

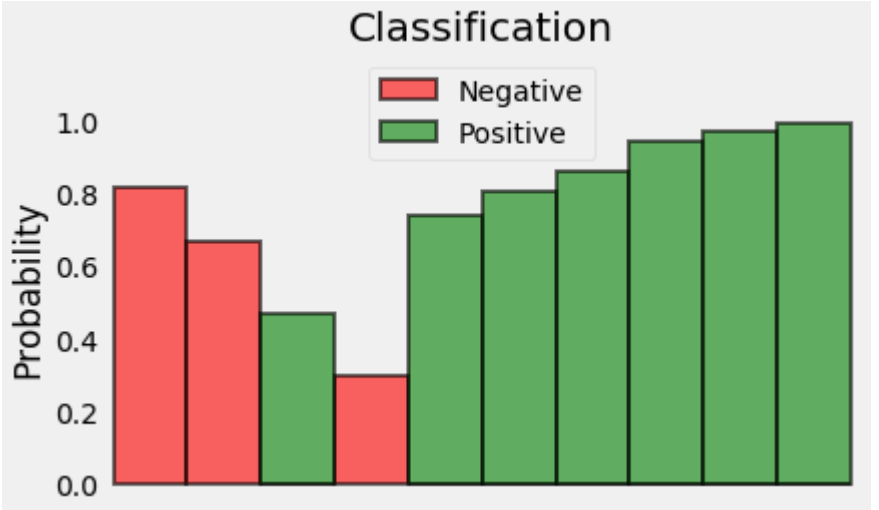
- Therefore the given text implies positive sentiment or positive feed back
- In the above example we are given data as text
- We converted the text into feature vector using lexicon of words
- We are given the model parameters after training
- Then we checked if the given text displays positive sentiment or negative sentiment

x = [-2.2, -1.4, -0.8, 0.2, 0.4, 0.8, 1.2, 2.2, 2.9, 4.6]









Support Vector Machines for classification - Theory

Q.What is SVM?

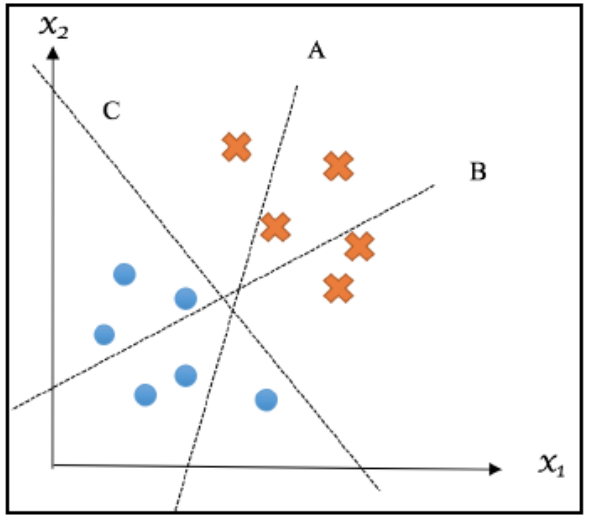
A.

- A leaning Machine which finds an optimal hyperplane that separates input space (observations) belonging to different classes.
- A hyperplane is a plane of n -1 dimension for a given n dimensional feature space(observations)
- The hyperplane separated the given input space into two sub spaces (classes)
- into two spaces.
- the hyperplane in a 2_D feature space is a line
- surface in a 3_D feature space.
- There can be many hyperplanes which can separate the input data into two classes.
- Finding the optimum plane which efficiently separates is the goal of SVM

Q. How SVM works?

A.

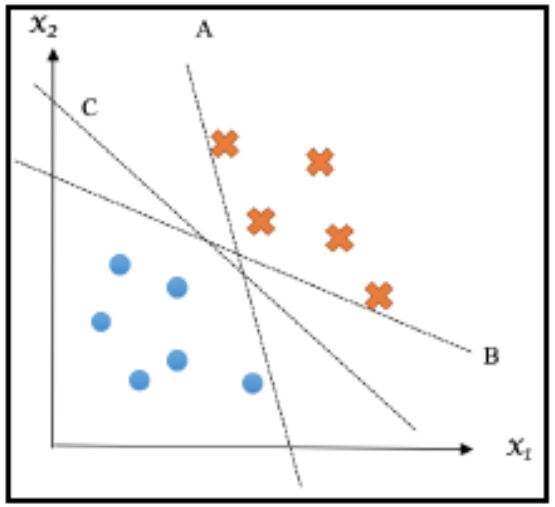
- A hyperplane is chosen and the distance from the nearest points in the two classes is obtained.
- The distance is maximized so that there is least overlap between the classes.
- The nearest points are called support vectors.
- SVM is also called maximum margin classifier
- In the figure three separating hyperplanes A,B and C are shown
- A and B are not good choices as there one point in each class which is not correctly classified
- C separates all points belonging to both classes.
- Hence hyperplane C is good choice for classification
- Points from blue and yellow classes closet to C
- are candidates for support vectors

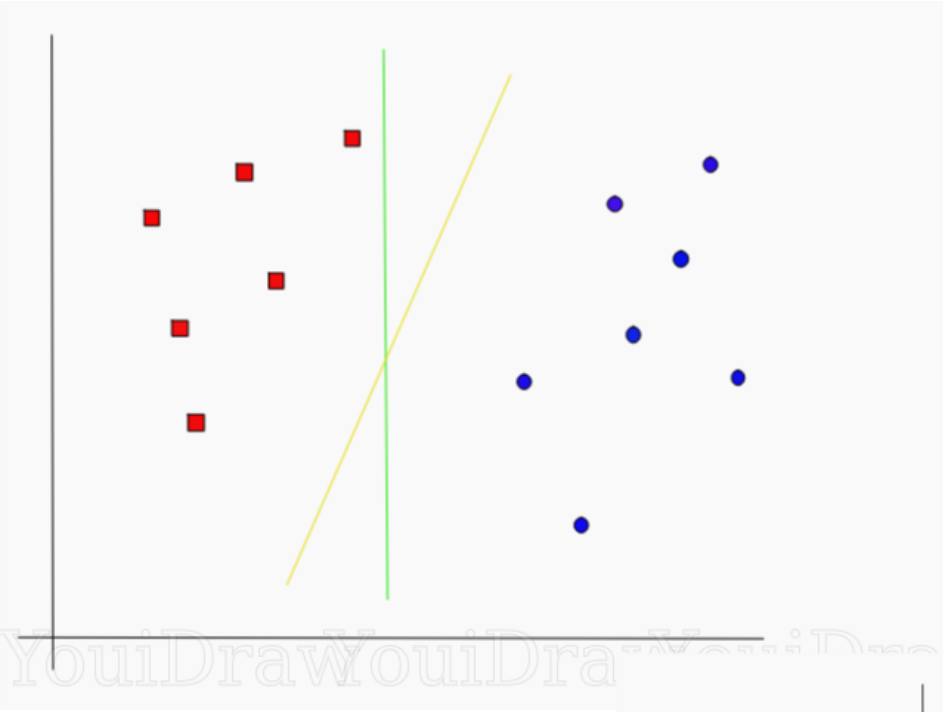


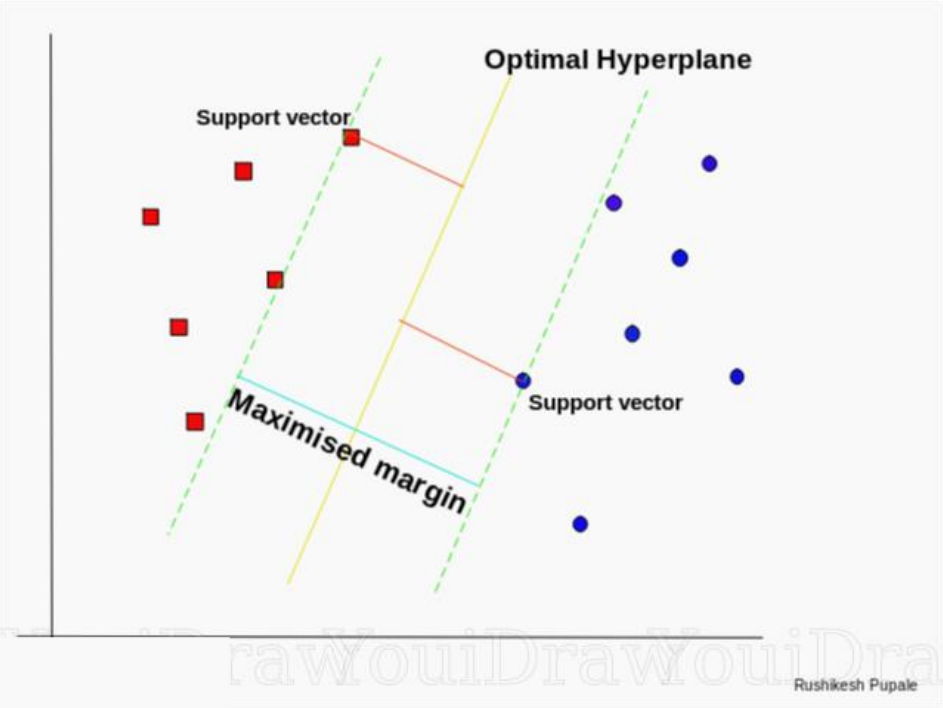
Q. How does SVM work?

A.

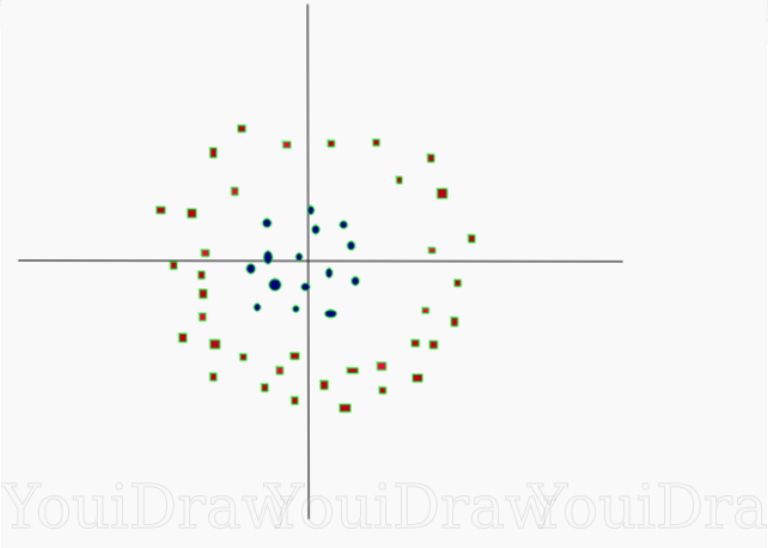
- Equation of a Hyperplane in N dimensions is $\theta X + \theta_0 = 1$(why not 0?)
- θ – parameters and θ_0 is constant term or bias term representing Hyperplane C
- For any given data point X find Y using above equation
- If Y › 0 -› label that point to class A
- Else if Y ‹ 0 -› label that to class B
- By using translation, rotation on C -› a new hyperplane can be generated still correctly classifying data!!!
- Goal is to find which optimum (A or B or C)?
- Difference between perpendicular distance between nearest points belonging to two classes is called margin





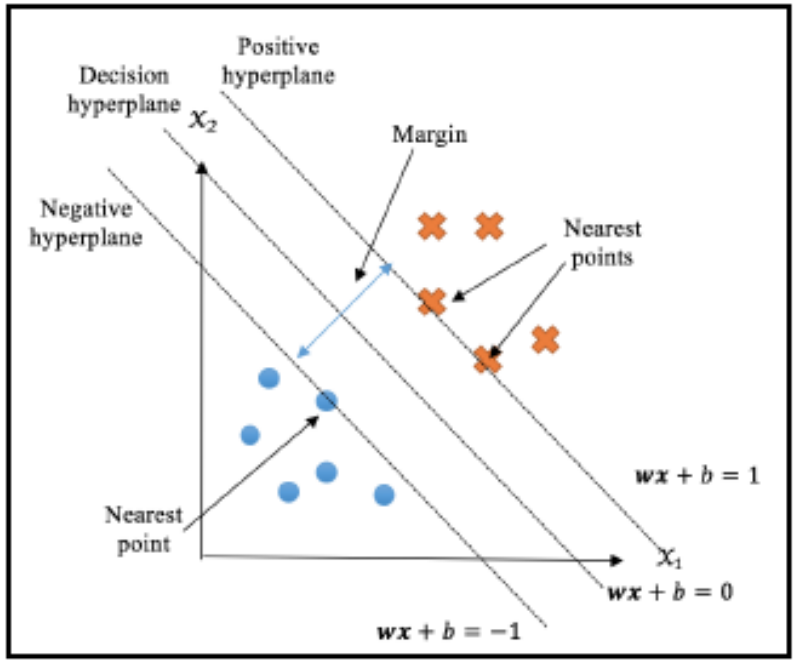


Linear separation



Q. How does SVM work?

A.



- Consider a hyperplane class A hyperplane as $\Theta X^A + \Theta_0 = 1$
- Consider another hyperplane nearer to class B as $\Theta X^B + \Theta_0 = -1$
- Perpendicular distance from a point belonging to class A and hyperplane A is

- $\frac{\|\Theta XA + \Theta_0\|}{\|\Theta\|} = \frac{1}{\|\Theta\|}$ and
- $\frac{\|\Theta XB + \Theta_0\|}{\|\Theta\|} = \frac{1}{\|\Theta\|}$
- Margin between the two perpendicular distances is $\frac{1}{\|\Theta\|} + \frac{1}{\|\Theta\|} = \frac{2}{\|\Theta\|}$
- To maximize distance, we should minimize $\|\Theta\|$
- Consider any point (x^i, y^i)
- Also we need to ensure if $y^i = 1$ (belongs to class A) $(\Theta X^A + \Theta_0) >= 1$
- if $y^i = -1$ (belongs to class B) $(\Theta X^B + \Theta_0) <= -1$

- The above expressions can be combined to obtain a single expression $y^i(\theta X^i + \theta_0) >= 1$
  - The above minimization results into a
  - Constrained optimization problem

  - For a given data set (X,Y) minimize $||\theta||$ subjected to $y^i(\theta X^i + \theta_0) >= 1$

Clustering - Discussion

Q. What is Clustering?

A.

- A Process in which similar data points are grouped
- Different groups , each having similarity among data points within the group, differ in similarity w.r.to data points in other groups
- Clustering is an optimization problem

Q. What are desirability of Clustering algorithm?

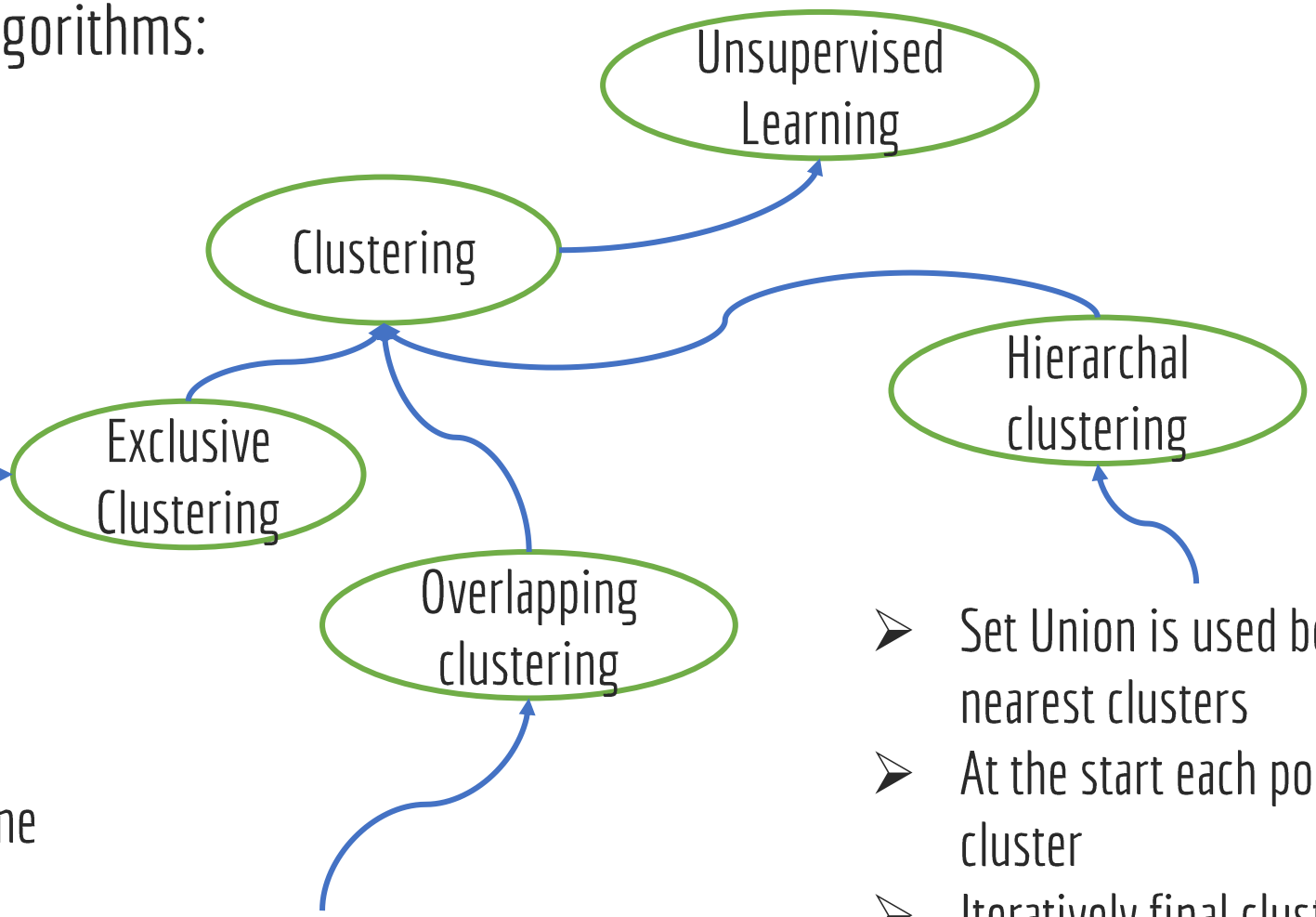
A.

- Clustering algorithm should be capable to form groups by separating them in any boundary / arbitrary shape
- Good degree of Similarity within a group and equally well dissimilarity among different groups is central to clustering
- Should have minimum domain knowledge to determine input features
- Scalable
- Not much influenced by noise in data
- Order of input data should not be matter
- Should deal with high dimensionality

Q. What are Types of clustering algorithms:

A.

- Exhaustive clustering
- Overlapping clustering
- Hierarchal Clustering



- Objects belonging one group are excluded from all other groups
- A data point can belong to only one group

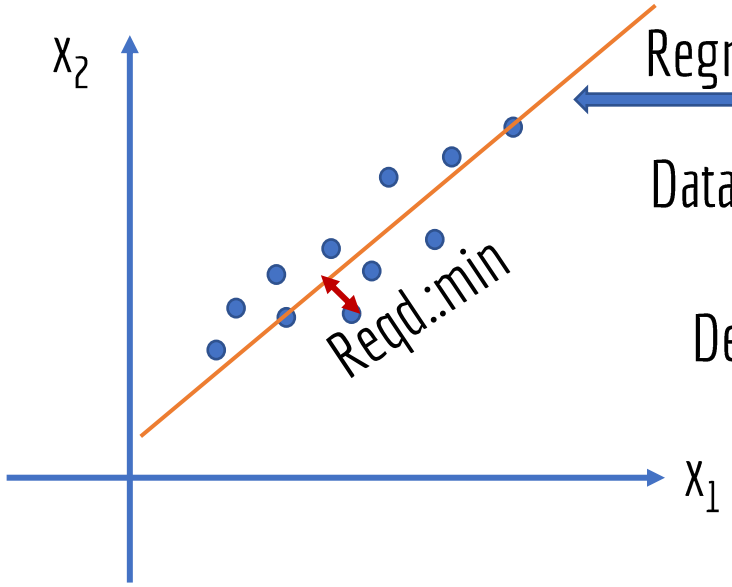
- A data point can be included in more than one group
- If a data point is represented in different groups, degree of membership for each group is differs
- Fuzzy set theory used

- Set Union is used between two nearest clusters
- At the start each point is cluster
- Iteratively final clusters are arrived.

Q. How similarity is defined in clustering algorithms?

A. $||L||_1$, $||L||_2$S, Cosine , $||L||_\infty$ etc.

Q. Differentiate between Regression, Classification & Clustering.

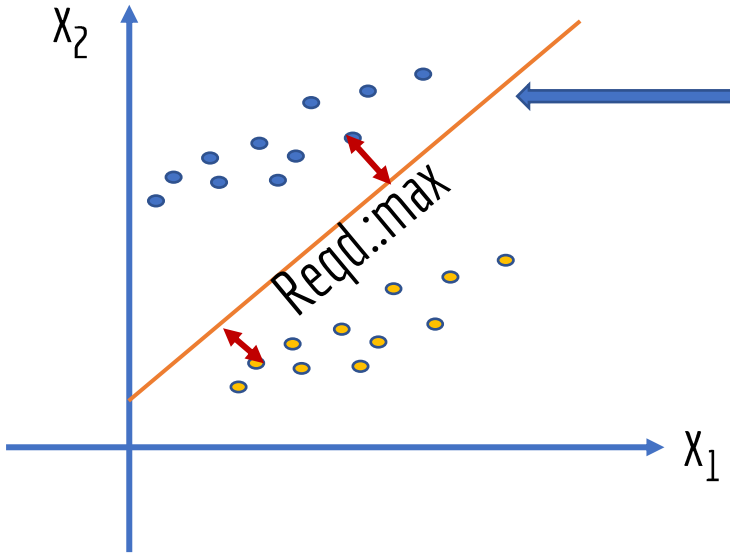


Regression

Data points represented by Regression Line/Plane/curve/surface

Desirability : Line must be close to maximum no of data points

Classification : Binary/Multi Classes



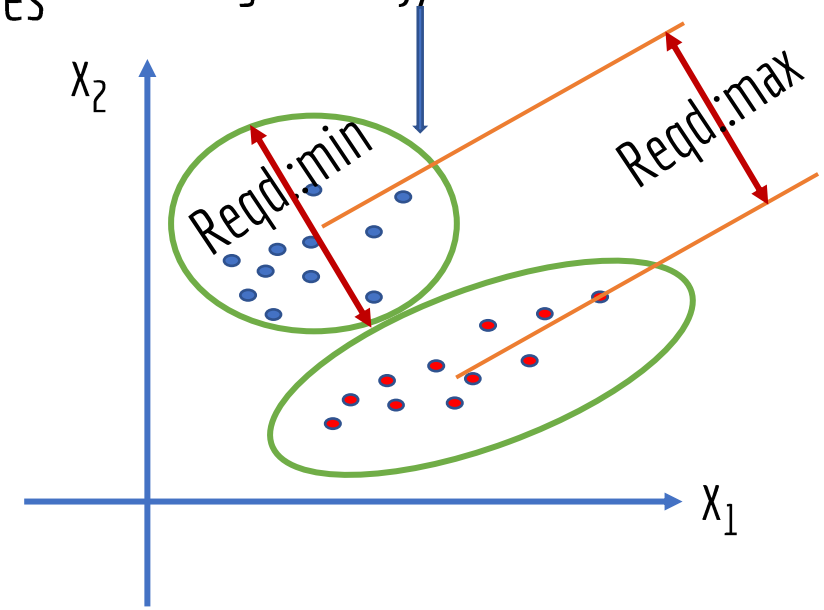
Line/plane/curve/surface represents Decision boundary

Desirability : Line must be as far as possible to data points

data for which the groups are known

Objective : try to learn what differentiates these groups to properly classify future data

Clustering : Binary/Multi Clusters



Line/plane/curve/surface represents Decision boundary

Desirability : high degree of similarity within a group and high degree of dissimilarity between different groups

Q. What are some Use Cases of Clustering

A.

- news articles or web pages by topic
- protein sequences by function, or genes according to expression profile •
- users of social networks by interest •
- customers according to purchase history

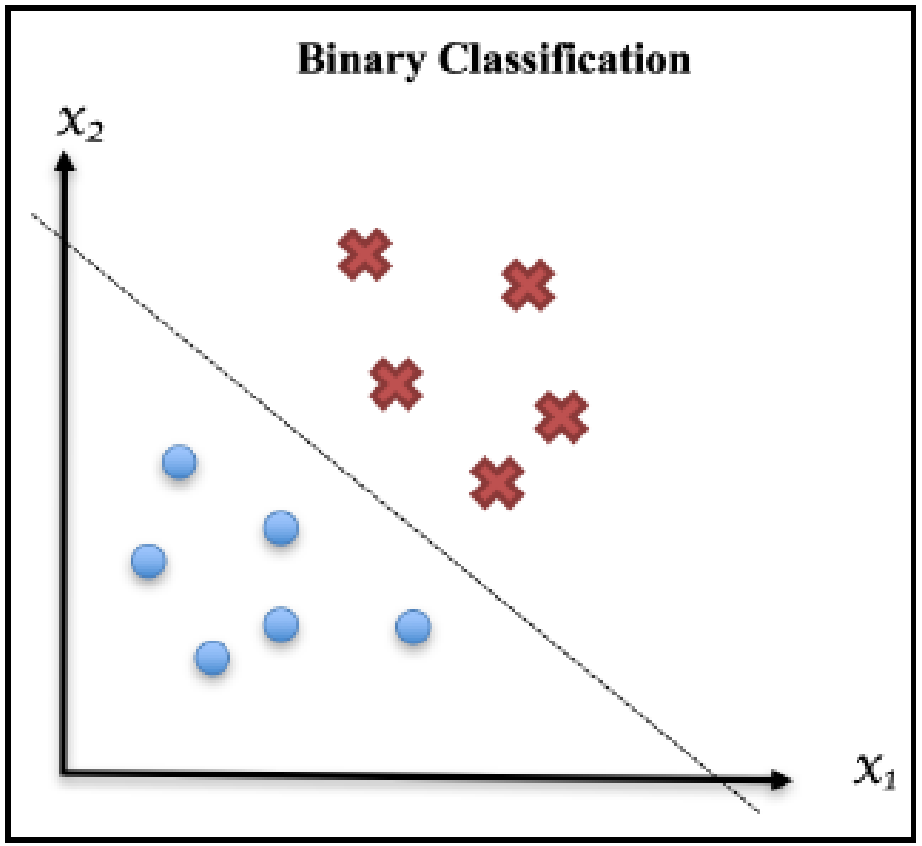
Decision Boundaries – Classification & Clustering

Q. What is Decision Boundary ?

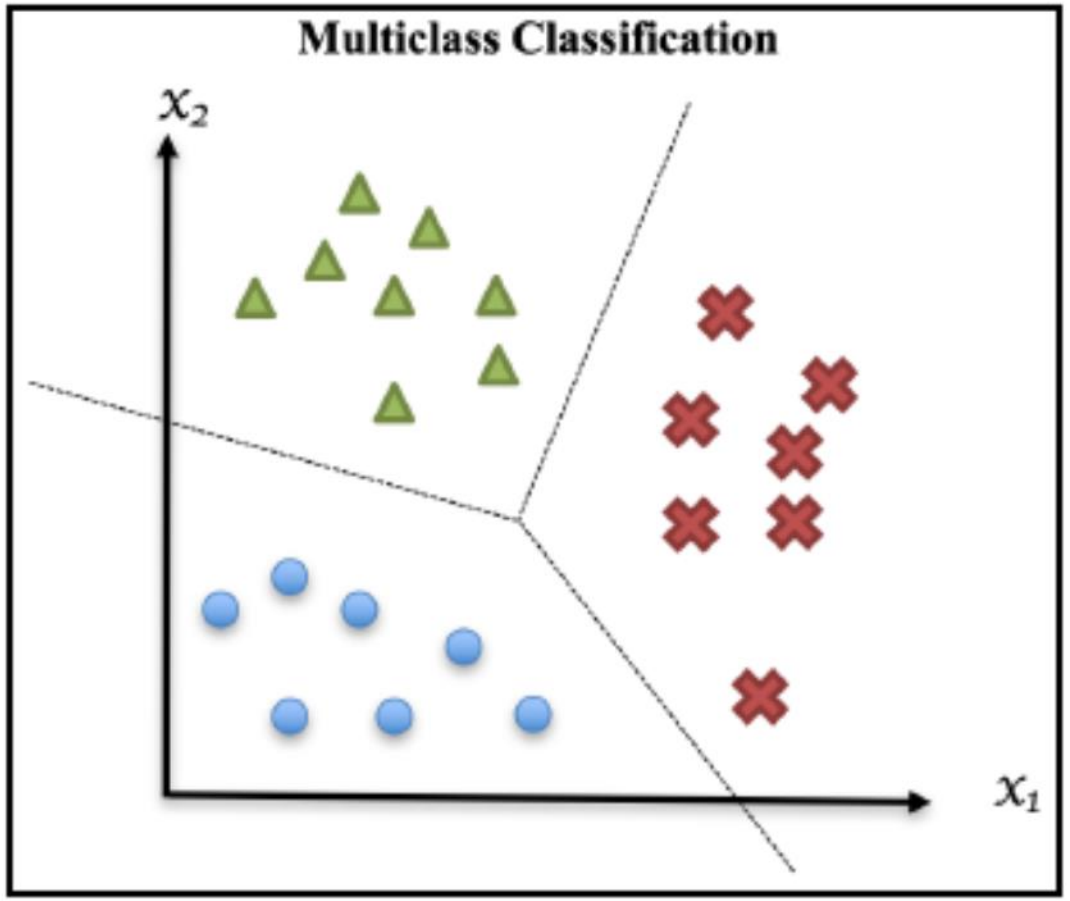
A.

- is a curve/surface that *separates* one class from the other
- Near the separating line/curve/surface in the problem space, the class label is ambiguous.
- Data points away from the separating line/curve/surface can be clearly identified with a label.
- If the decision surface is a line/hyperplane, then the classification problem is linear, and the classes are linearly separable.
- Majority of data samples of one class are on one side of the decision boundary & all other data samples of the other class are on the opposite side of the decision boundary

Q. Binary vs multi class classification?

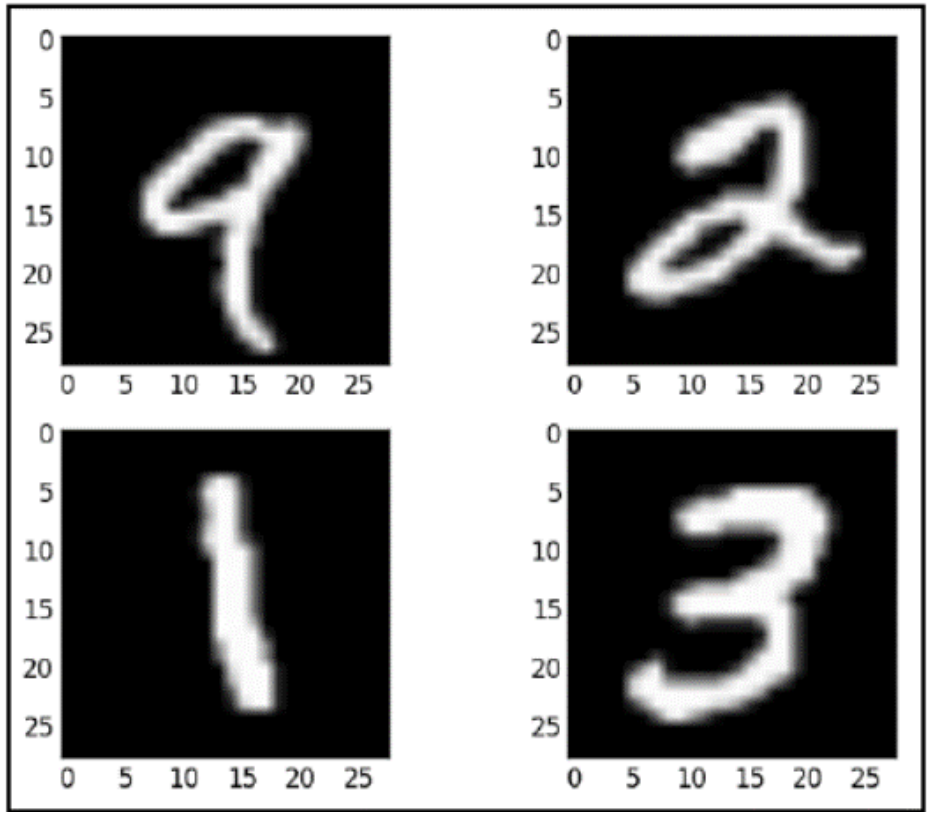


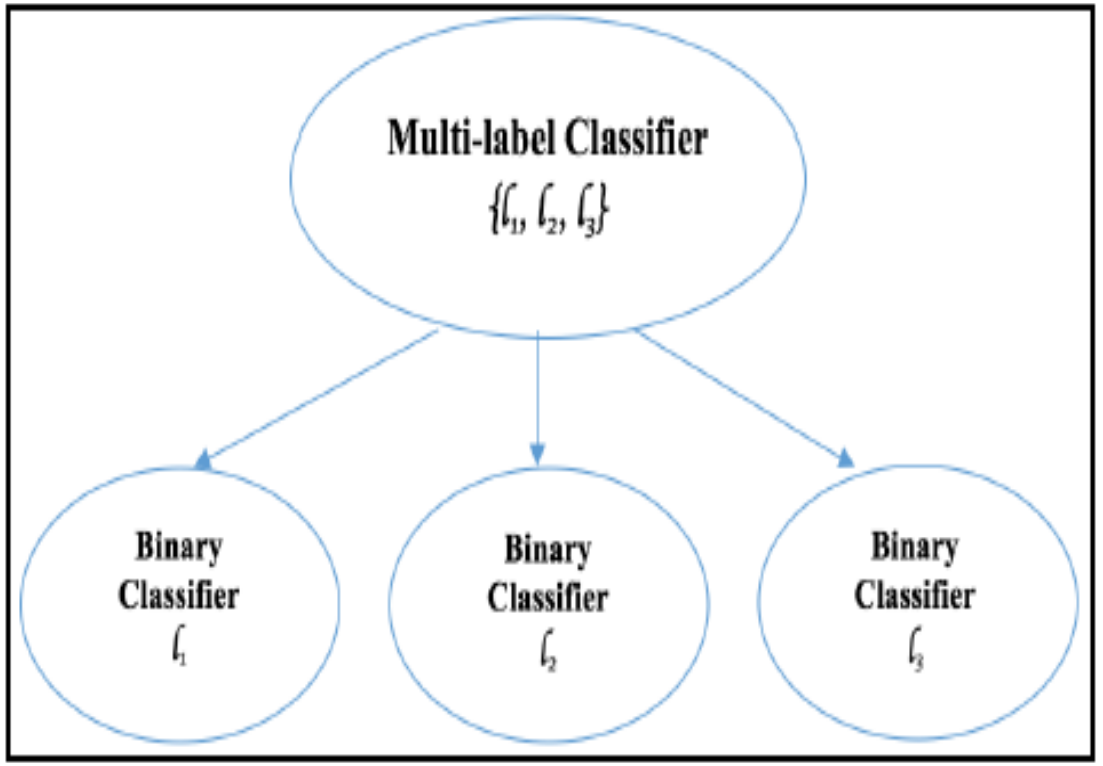
- allows only two classes



- allows more than two possible classes, as opposed to only two in binary cases.

Q. Multi class classification

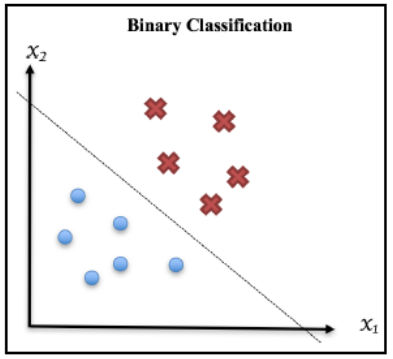


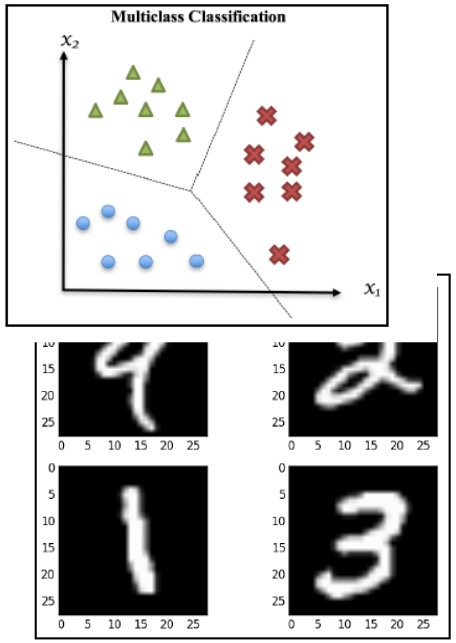


- solve an n label classification problem
- transform it into a set of n binary classifications problems
- handled by individual binary classifiers.

Binary classification: Use cases

- classify observations into one of two possible classes.
- Ex. spam email filtering
- Input : email messages (input observations) as
- Customer segment data and activity data from CRM systems
- identifies which customers are likely to churn.
- Ex. Given user's cookie info and browsing, Check if an Ad will be clicked or not.
- Ex. early cancer diagnosis, classifying patients into high or low risk groups based on MRI images.

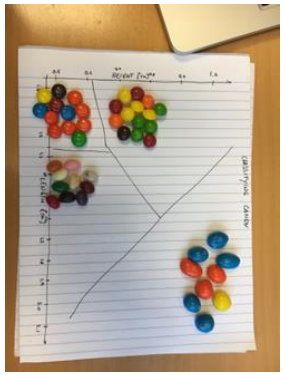


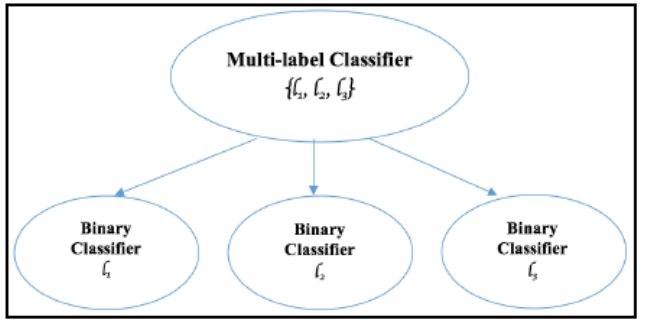


- **Multiclass classification (multinomial classification)**:
- allows more than two possible classes

Ex. Handwritten digit : ZIP codes -› envelopes are automatically sorted

- Number plate identification
- Data : Modified National Institute of Standards and Technology,(**MNIST)**, is a benchmark dataset
- evaluate multiclass classification of hand written digits



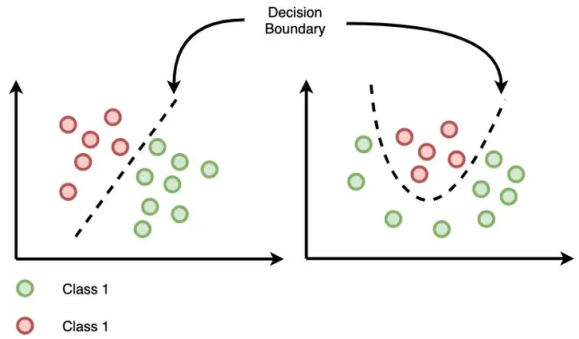


Multi-label classification:

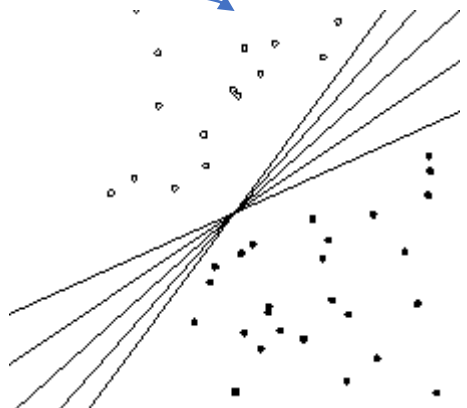
- a picture of a sea and sunset can simultaneously belong to both conceptual scenes
- an image of either cat or dog in a binary case
- one type of fruit among oranges, apples, and bananas in a multiclass case.
- adventure films are often combined with other genres, such as fantasy, science fiction, horror, and drama.
- protein function classification -› protein may have more than one function—storage, antibody, support, transport, etc.
- *n* label classification problem is to transform it into a set of *n* binary classifications problem, then handled by individual binary classifiers.

What is the decision boundary?

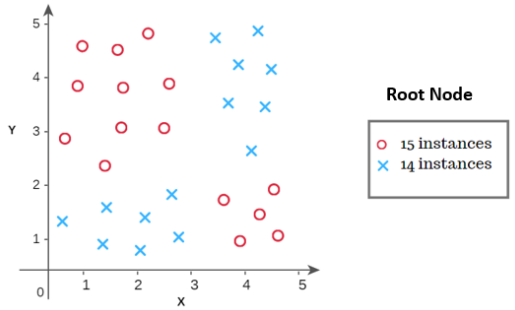
- A function representing a boundary which either classifies different data or to groups similar data
- Can be linear or non-linear
- Linear : line, plane or hyper plane
- Non –linear : polynomial, surface

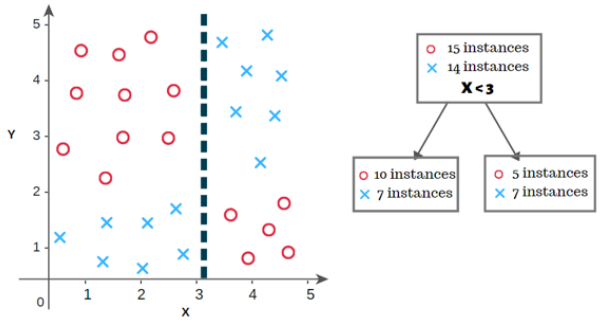


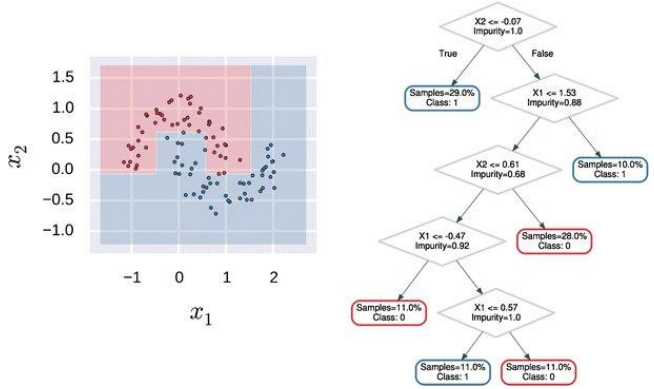
infinite number of hyperplanes that separate two linearly separable classes.

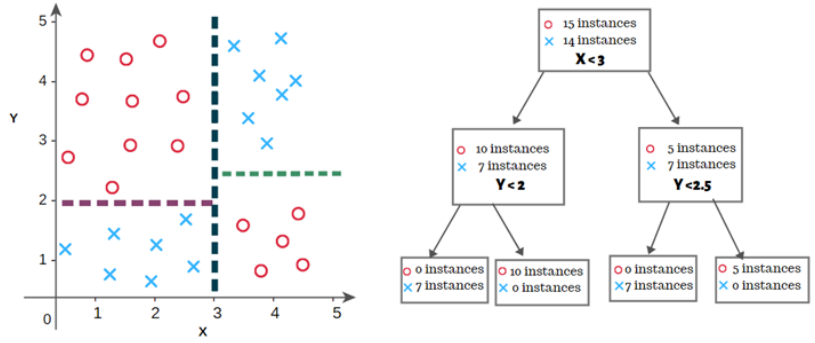


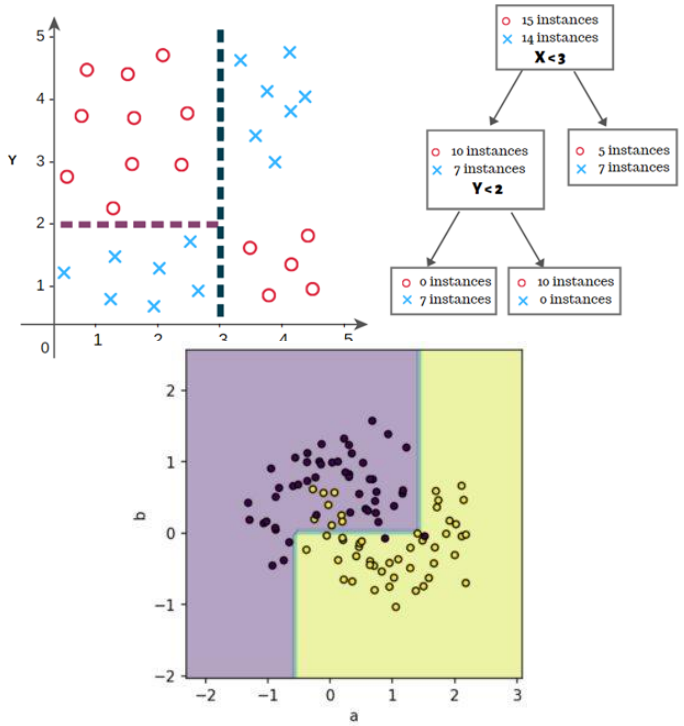
Decision Boundary in Decision Tree





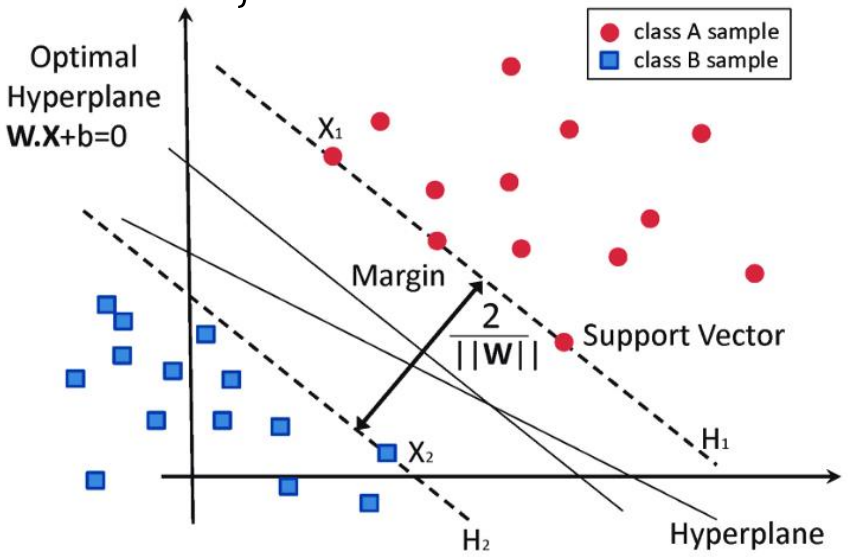




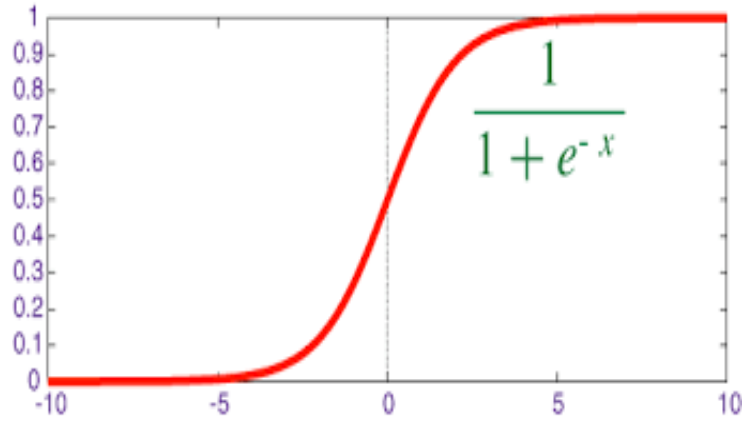


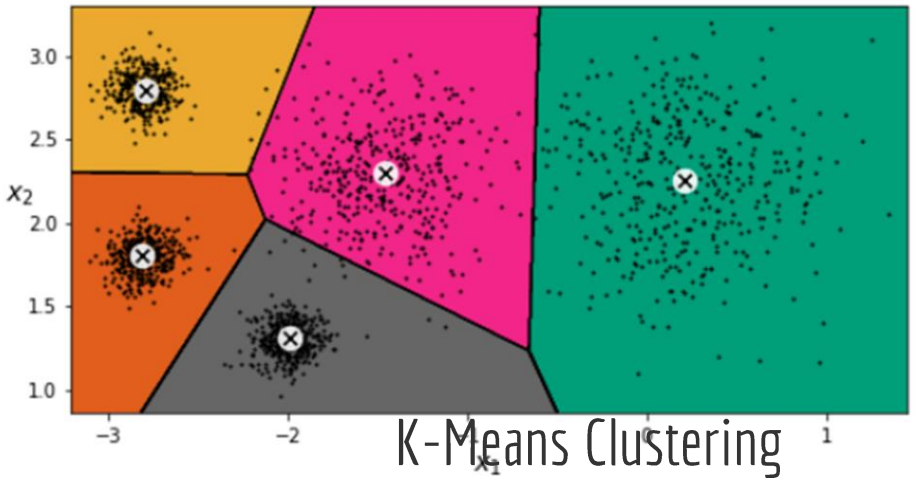
draw a line called “decision boundary” that separates the instances of different classes into different regions called “decision regions”.

Decision Boundary in SVM



Decision Boundary in SVM is linear with sta



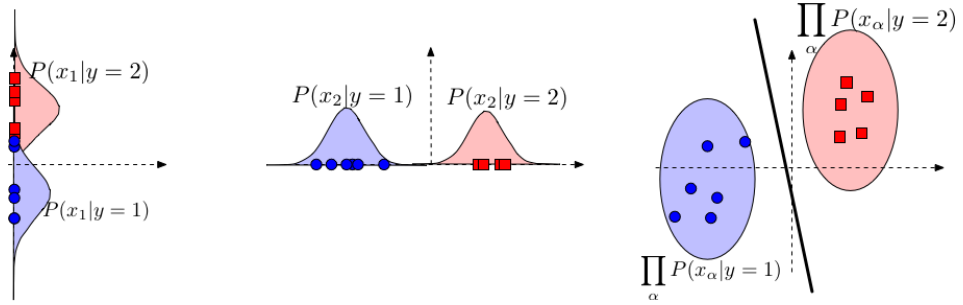


Logistic Regression – Classification
Decision Boundary is line

K-Means Clustering
Decision Boundary is Voronoi Cells

- Different classification and clustering algorithms have different decision boundaries

Naïve Bayes Classifier : Decision boundary

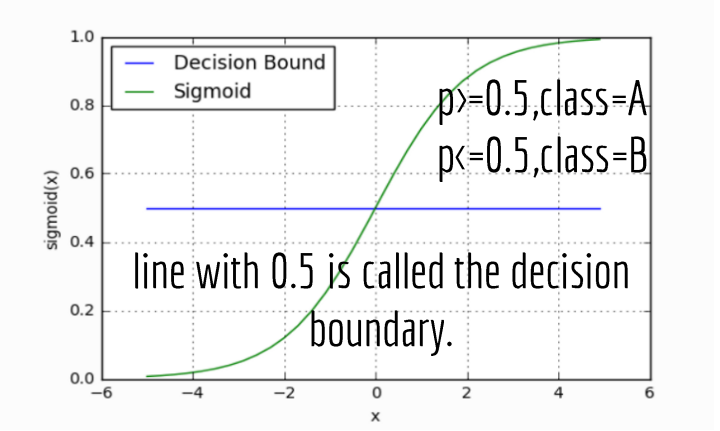


- Naive Bayes is linear classifier using Bayes
- strong independence condition among features.
- Given a data set with n features represented $F_1, F_2, F_3, \ldots F_n$
- Naive Bayes states the probability of output:
- Y=class from features F_1 F_2..... F_n
- This requires that the features F_i are conditionally independent.
- From Bayes Theorem:

$$P(Y|F_1,F_2,F_3,\ldots,F_n) = P(Y|F_1)P(Y|F_2)P(Y|F_3)\ldots P(Y|F_n) = \Pi_{i=1}^{n} P(Y|F_i)$$

- requires that the features F_i are conditionally independent.
- From Bayes Theorem: $P(Y|F_i) = \frac{P(F_i|Y)P(Y)}{P(F_i)}$

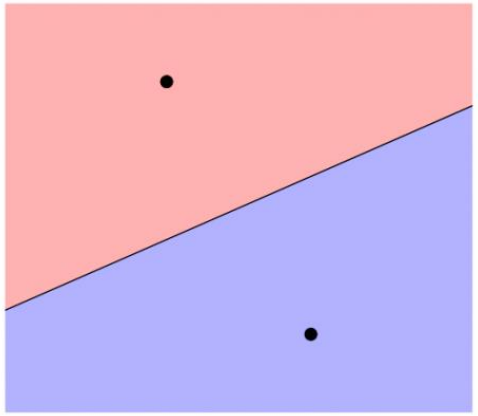
Logistic Regression : Decision boundary



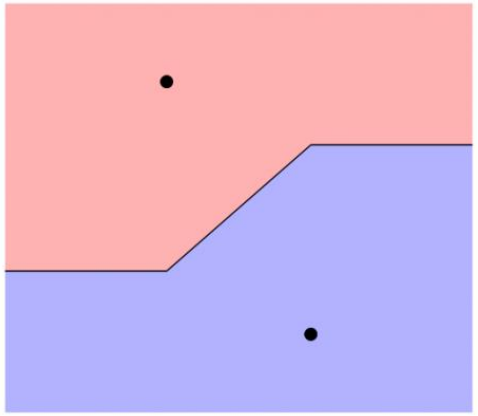
$$P(Y=1|F) = \frac{1}{1+exp(\beta_0 + \Sigma_{i=1}^{n}\beta_i F_i \rightarrow)}$$

$$P(Y=0|F) = \frac{exp(\beta_0 + \Sigma_{i=1}^{n}\beta_i F_i \rightarrow)}{1+exp(\beta_0 + \Sigma_{i=1}^{n}\beta_i F_i \rightarrow)}$$

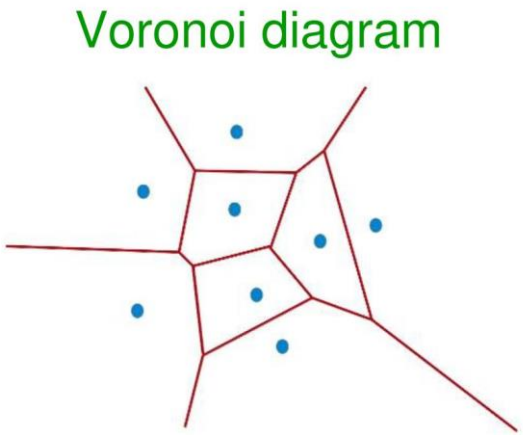
How the distance metric effects the k-means clustering and similarity to Voronoi tessellation?



Left : Euclidian



Right : Manhattan



Voronoi tessellation : partition of a plane into regions close to each of a given set of objects.

- The dividing line in Voronoi diagram is a perpendicular bisector between points!!!

K – Means : result of the algorithm is a list of centroids and assignment of every point to a centroid.

Voronoi Tessellation : The result of the algorithm is a list of polygons - the partition.

Classification:
Can be considered as “learning good decision boundaries”
Decision Boundary : a line/plane/hyper plane/curve/surface/hyper surface that separates the input data belonging to different classes in the data set.

In case of k-NN

- Decision Boundaries change with K
- Decision Boundaries change with the distance function

Clustering – Theory

Frequently Asked Questions on Clustering

- What is clustering?
- Is clustering a supervised or an unsupervised technique?
- Can you describe the K-means algorithm?
- What is its complexity? How would you reduce the number of computations?
- What are some termination criteria?
- What is the difference between hard and soft assignments?
- What are some weaknesses of k-means?
- The algorithm has some assumptions about the data distribution. What are they?
- How do you find the optimal number of clusters?
- How come two different executions of the algorithm produce different results? How does one address this problem?
- What kind of preprocessing would you apply to the data before running the algorithm?
- Is the algorithm sensitive to outliers? If yes, how could you mitigate the problem?
- What are the applications of k-means clustering?

K-Means Clustering – Theory

K-Means Algorithm

Data : $X = \{ X_1, X_2, ... X_n \}; X_i \in \mathbb{R}^d$

Cluster Centers : $C = \{C_1, C_2, ... C_K \}; C_j \in \mathbb{R}^d$

Cluster members $S = \{S_1, S_2, ... S_K \}; S_r \subseteq X$

Initialization : $C^0 \in X; |C^k|$

Assignment : $S^t = \{ x \in X; || x - C_i || < || x - C_j ||; 1 <= j <= K \}$

Update : $C_j^{t+1} = 1/(| \{S_j^t|) \Sigma x; x \in S_j^t|; 1 <= I <= K \}$

Convergence : $||C_j^{t+1} - C_j^t || <= \varepsilon; \varepsilon \in \mathbb{R}$

K-Means:Given

K = 3

Data points

Similarity metric Manhattan distance

	X	Y
A_1	2	10
A_2	2	5
A_3	8	4
A_4	5	8
A_5	7	5
A_6	6	4
A_7	1	2
A_8	4	9

Solution

Initial Guess of cluster center

C1 (2,10)

C2 (5,8)

C3(1,2)

Iteration 1

	X	Y	C1-d	C2-d	C3-d	Cluster
A_1	2	10	0	5	9	C1
A_2	2	5	5	6	4	C3
A_3	8	4	12	7	9	C2
A_4	5	8	5	0	10	C2
A_5	7	5	10	5	9	C2
A_6	6	4	10	5	7	C2
A_7	1	2	9	10	0	C3
A_8	4	9	3	2	10	C2

Cluster C1:{ A1}, C2 :{A3,A4,A5,A6}; C3 :{A2,A3}

Update Cluster Center

:C1 = A1 = (2,10)

C2 : (A3+A4+A5+A6+A8)/5 =(

(2+5+7+6+4)/5,(4+8+5+4+9)/5) = (6,6)

C3 : (A2+A7)/2 = ((2+1)/2,((5+2)/2) = (1.5,3.5)

Iteration 2 C1 = (2,10),C2 : (6,6),C3 : (1.5,3.5)

	X	Y	C1-d	C2-d	C3-d	Cluster
A_1	2	10	0	8	7	C1
A_2	2	5	5	5	2	C3
A_3	8	4	12	4	7	C2
A_4	5	8	5	3	8	C2
A_5	7	5	10	2	7	C2
A_6	6	4	10	2	5	C2
A_7	1	2	9	9	2	C3
A_8	4	9	3	5	8	C1

Cluster C1:{ A1,A8}
C2 :{A3,A4,A5,A6};
C3 :{A2,A7}

Update Cluster Center
:C1 = (A1+A8)/2 = (3,9.5)
C2 : (A3+A4+A5+A6)/4 =(26/4,21/4)
C3 : (A2+A7)/2 = (3/2,7/2)

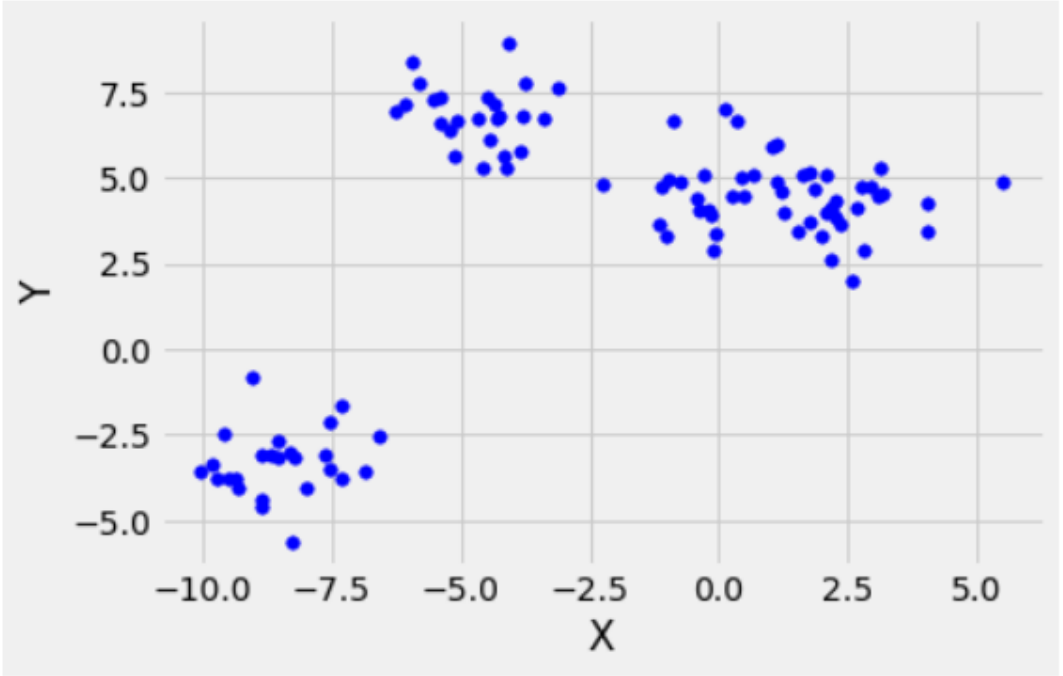
Q. How to choose K ?

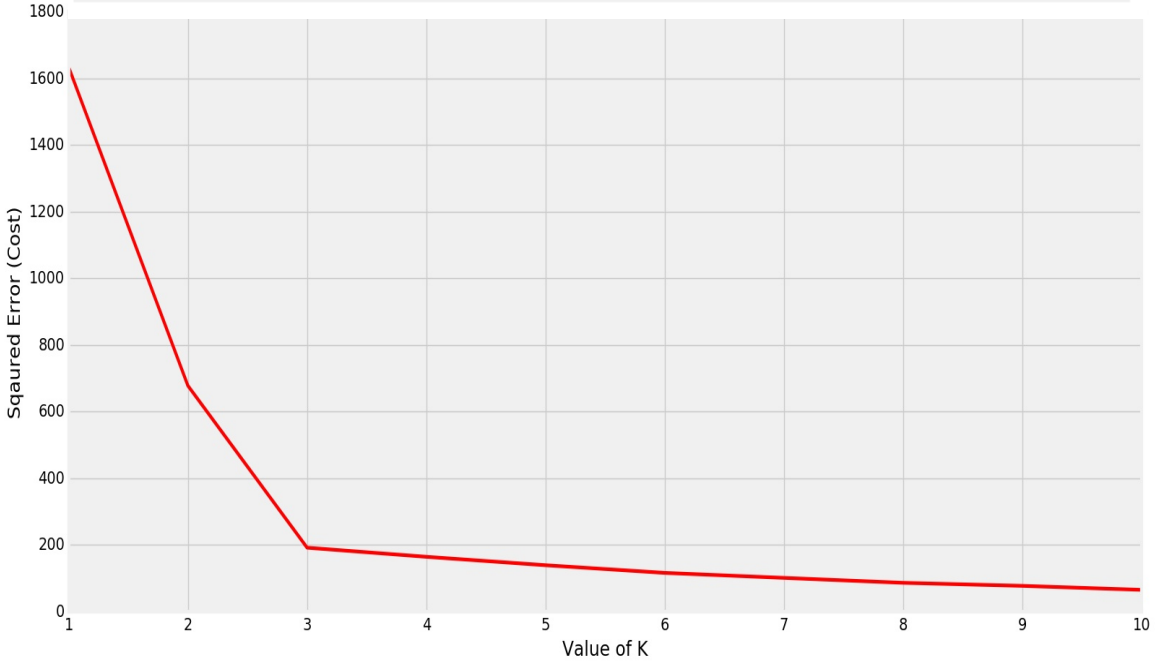
A.

- One method is ELBOW technique
- As the value of K increases, fewer elements in a cluster are found.
- the point where there distortion declines the most is the elbow point.

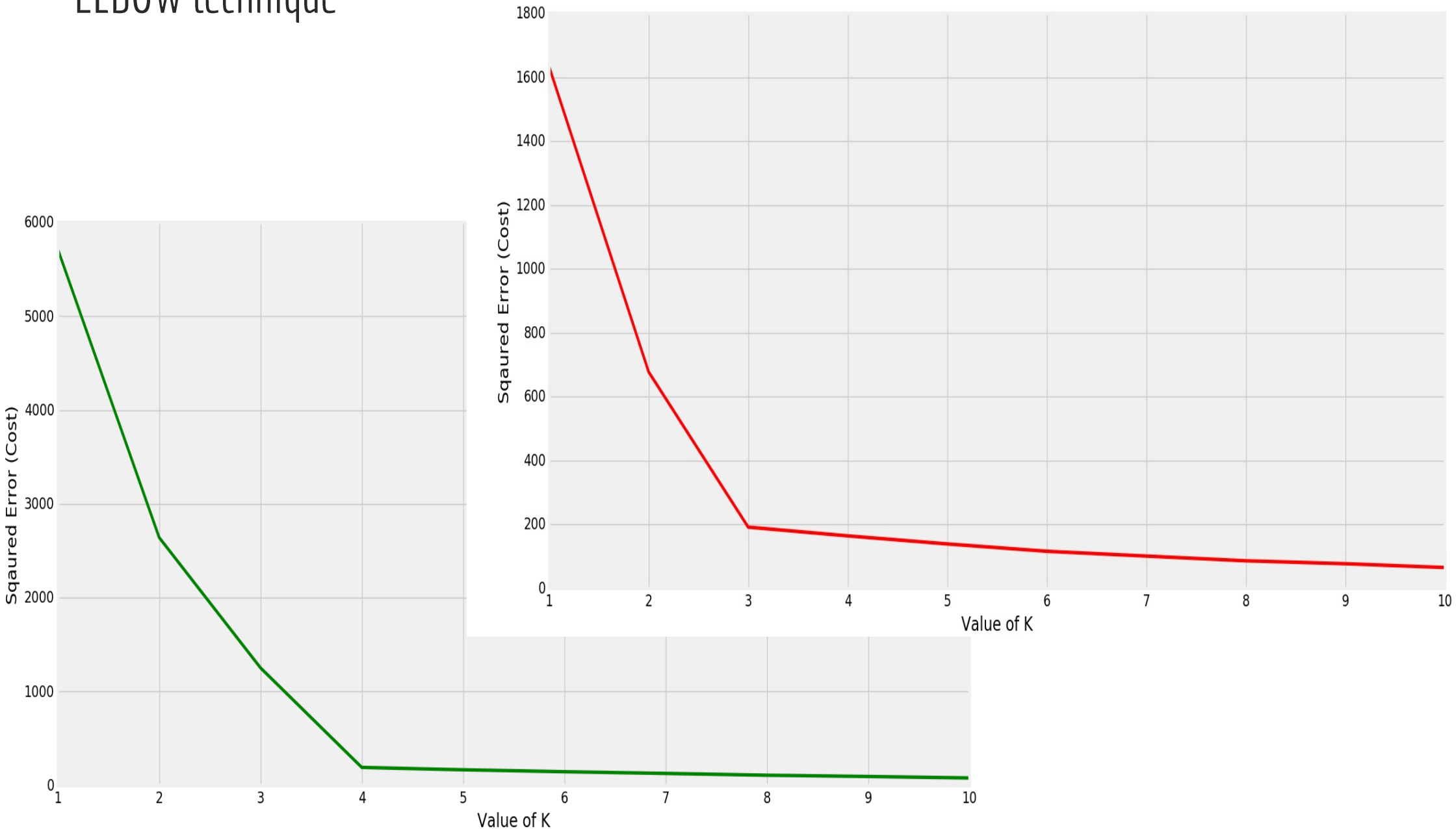
ELBOW technique

- Runs k-means clustering on the dataset for a range of values for k (say from 1-10)
- for each value of k computes an average score for all clusters.
- the distortion score is computed as the sum of square distances from each point to its assigned center.
- the line chart looks like an arm
- the “elbow” (the point of inflection on the curve) is the best value of k.
- The “arm” can be either up or down
- We look for a strong inflection point,
- Value of K at inflection point indicates that the underlying model fits best at that point.





ELBOW technique



•May end up with suboptimal solutions and a local minima - due to the random initialization of centroids
• important to run the algorithm several times to avoid this.
•specify k, which can be subjective .
•does not perform particularly well when clusters are different sizes, densities or non-spherical shapes.
•gives more weight to bigger clusters than smaller ones.
• data points in smaller clusters may be left away from the centroid in order to focus more on the larger cluster.
•leads to poorly assigned smaller clusters due to unbalanced data.
•assigns each instance into a non-overlapping cluster, there is no measure of uncertainty for points that lie near boundary lines

K-Medoid Clustering – Theory

The steps followed by the K-Medoids algorithm for clustering are :

1.Randomly choose ‘k’ points from the input data (‘k’ is the number of clusters to be formed).

2.Each data point gets assigned to the cluster to which its nearest medoid belongs.

3.For each data point of i^{th} cluster, its distance from all other data points is computed and added.

4.The point of i^{th} cluster for which the computed sum of distances from other points is minimal is assigned as the medoid for that cluster.

4.Steps (2) and (3) are repeated until convergence is reached i.e. the medoids stop moving.

- The complexity of the K-Medoids algorithm : $O(N^2K)$
- N : the number of data points
- K : number of clusters

Q. What is medoid?

- In statistics for 1D data, Median is the middle point when the data is arranged in ascending order
- For odd number of data points, middle point is easy.
- For even number data there can be two data points if a constraint is put that the median is point in the data set
- If such constraint is not imposed, average of the middle points can be taken
- It is not easy or even not possible to sort the data for higher dimensional data.
- Medoid is an extension of finding a representative data point similar to median for multi dimensional data

Q. Give an algorithm to arrive at medoid for n-D data?

Let : $S = \{ x_1, x_2, ... x_n \}$; $x_i \in \mathbb{R}^d$

$\bar{x} = (1/n) \Sigma X_i$; $1 <= i <= n$

Let $X_i \in S$ such that

$d(x_i, \bar{x}) <= d(x, \bar{x})$; $\forall x \in S$

Medoid of $S = x_i$

X_i is closet to $\bar{x}$

Let : $S = \{ x_1, x_2, ... x_n \}$; $x_i \in \mathbb{R}^d$

$a_i = \max\{d(x,x_i); x \in S\}$; $i=1,2..n$

$a_{i0} = \min (a_i)$

Medoid of $S = x_{i0}$

Medoid $Mediod(x_1, x_2..x_n) \in argmin(\sum_1^n d(x, x_i))$

$x \in (x_1, x_2..x_n)$

- Initialize k clusters in the given data space S
- Randomly choose k x_m data points (medoids) from dataset
- Assign the chosen medoids to K clusters
- $\forall x, x \in S$ & $x \neq x_j$ compute the $sim(x,x_j)$; $1<=j<=m$
- Assign x to that cluster whose $sim(x,x_j)$ is minimum among all prev. medoids x_j
- C^t = Update the total cost i.e., of choosing the K medoids at iteration t
  - sum of all the non-medoid objects distance from its cluster medoid.
- Randomly select new-non-medoid point.
- swap the prev. medoid point with new medoid point
- Update cost and assign it to C^{t+1}.
- If $C^{t+1}<C^t$ swap cluster membership
- Repeat until no change in cluster membership or cost reaches an inflection point

Clustering : K Medoid

Given : k = 2 ; Similarity Metric = $||L||_2$

Order of Complexity : $O(k(n-k)^2)$.

Iteration : 1; choose medoids as (1,4) & (10,4)

Convergence :

CurrCost-PrevCost < 0; Clustering is Medoids of Prev.Cost

	X	Y	D1	D2	Closest
A_1	1	4	**0**	9	(1,4)
A_2	5	1	5	5.83	(1,4)
A_3	5	2	4.47	5.39	(1,4)
A_4	5	4	4	5	(1,4)
A_5	10	4	9	**0**	(10,4)
A_6	25	4	24	**15**	(10,4)
A_7	25	6	24.08	**15.13**	(10,4)
A_8	25	7	24.19	**15.3**	(10,4)
A_9	25	8	24.33	**15.52**	(10,4)
A_{10}	29	7	28.16	**19.24**	(10,4)

cost is the sum of distance of each non-medoid point from the medoid of the cluster

Iteration 1 : the clusters formed : {(1,4), (5,1), (5,2), (5,4)} and{(10,4), (25,4), (25,6), (25,7), (25,8), (29,7)}.

Total Cost :Cost[(1,4),(5,1)]+cost[(1,40),(5,2)]+ cost[(1,4),(5,4)] + Cost[(10,4),(25,4)]+cost[(10,4),(25,6)]+cost[(10,4),(25,7)]+cost[(10,4),(25,8)] + cost[(10,4),(25,8)]+cost[(10,4),(25,7)]

=|1-5|+|1-4|+|1-5|+|4-2|+|1-5|+|4-4|+|10-25|+|4-4|+|10-25|+|6-4|+|10-25|+|7-4|+|10-25|+|4-8|+|29-10|+|7-4|

=4+3+4+2+4+15++15+2+15+3+15+4+19+3 = 108

Iteration : 2; choose medoids as (5,4) & (25,7)

	X	Y	D1(5,4)	D2(25,7)	Closest
A_1	1	4	**4**	24.19	(5,4)
A_2	5	1	3	20.8	(5,4)
A_3	5	2	2	20.62	(5,4)
A_4	5	4	0	20.22	(5,4)
A_5	10	4	5	**15.3**	(5,4)
A_6	25	4	20	**3**	(25,7)
A_7	25	6	20.10	**1**	(25,7)
A_8	25	7	20.22	**0**	(25,7)
A_9	25	8	20.40	**1**	(25,7)
A_{10}	29	7	24.19	**4**	(25,7)

Iteration 2 : clusters formed : {(1,4), (5,1), (5,2), (5,4),(10,4)} and {((25,4), (25,6), (25,7), (25,8), (29,7)}.

Total Cost = cost((5, 4), (1,4)) + cost((5,4), (5, 1)) + cost((5, 4), (5, 2)) + cost((25,7), (25,4))+ cost((25, 7), (25, 6))+ cost((25, 7), (25, 8))+ cost((25,7), (29, 7))
=4+3+2+3+1+1+1+4 = 23

Swapping cost= Curr.cost- Prev.Cost
cost = 23 – 108 = -85

Iteration : 3; choose medoids as (10,4) & (29,7)

	X	Y	D1(10,4)	D2(29,7)	Closest Object
A_1	1	4	**9**	28.16	(10,4)
A_2	5	1	5.4	24.73	(10,4)
A_3	5	2	5.3	24.51	(10,4)
A_4	5	4	5	24.18	(10,4)
A_5	10	4	0	**19.23**	(10,4)
A_6	25	4	15	**5**	(29,7)
A_7	25	6	15.13	**4.12**	(29,7)
A_8	25	7	15.29	**4**	(29,7)
A_9	25	8	15.52	**4.12**	(29,7)
A_{10}	29	7	19.23	**0**	(29,7)

Iteration 3 : clusters formed : {(1,4), (5,1), (5,2), (5,4),(10,4))} and {((25,4), (25,6), (25,7), (25,8), (29,7)}.

Total Cost = cost((10,4), (1,4)) +cost((10,4), (5,1)) + cost((10,4), (5,2)) + cost((10,4), (5,4)) + cost((10,4), (10,4))+ cost((29,7), (25, 4))+ cost((29,7), (25, 6))+ cost((29,7), (25,8))+cost((29,7), (29,7)) = 9+8+4+9+5+3+1=35

Swapping cost= Curr.cost- Prev.Cost = 23 – 35 = -12

Hierarchical clustering

Single linkage criterion

- Distance between two clusters is the smallest pairwise distance between the data points each belonging to different clusters

Complete linkage criterion

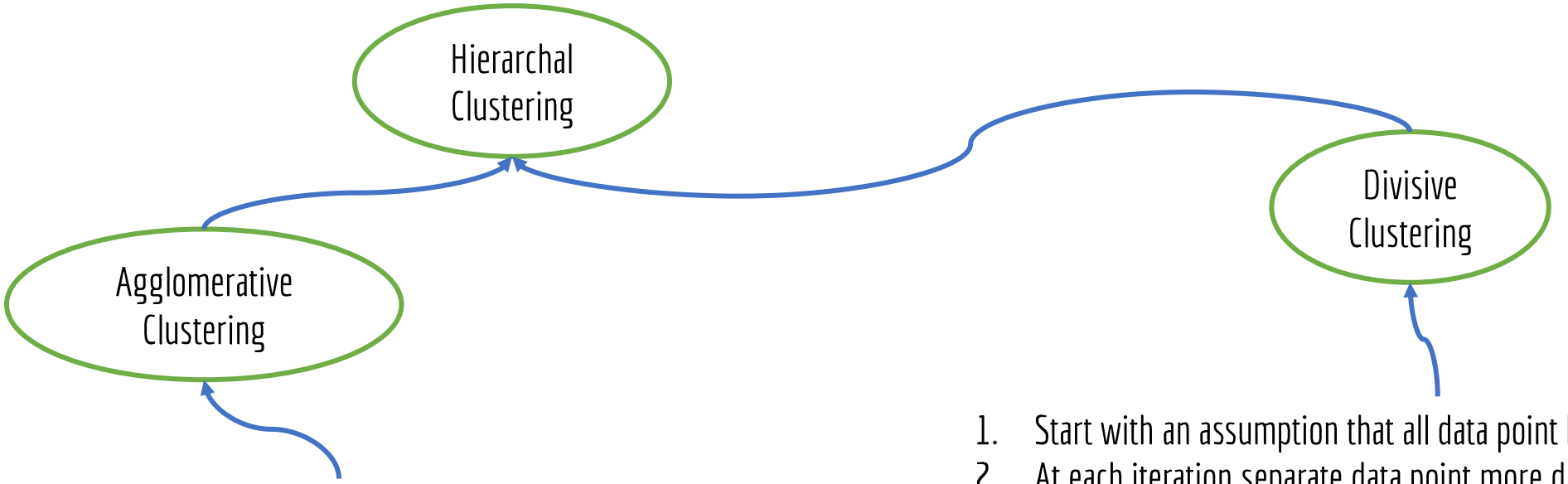
- Distance between two clusters is the largest pairwise distance between two data points each belonging to different clusters

Mean Linkage Clustering

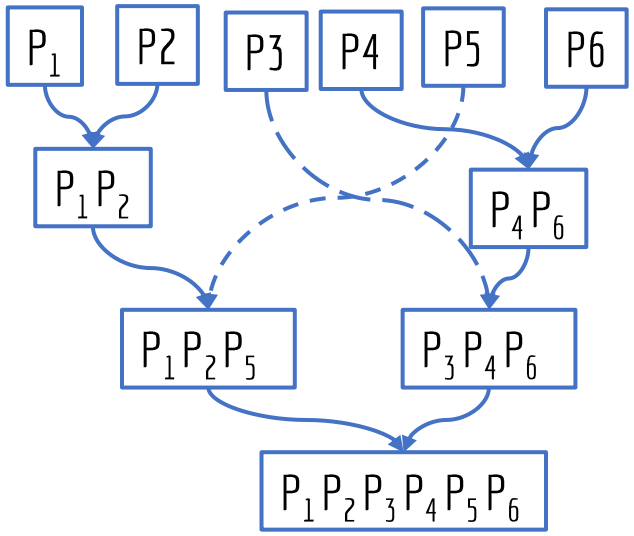
- Distance between two clusters is the average of all pairwise distances each data point belonging to different clusters

Centroid linkage criterion

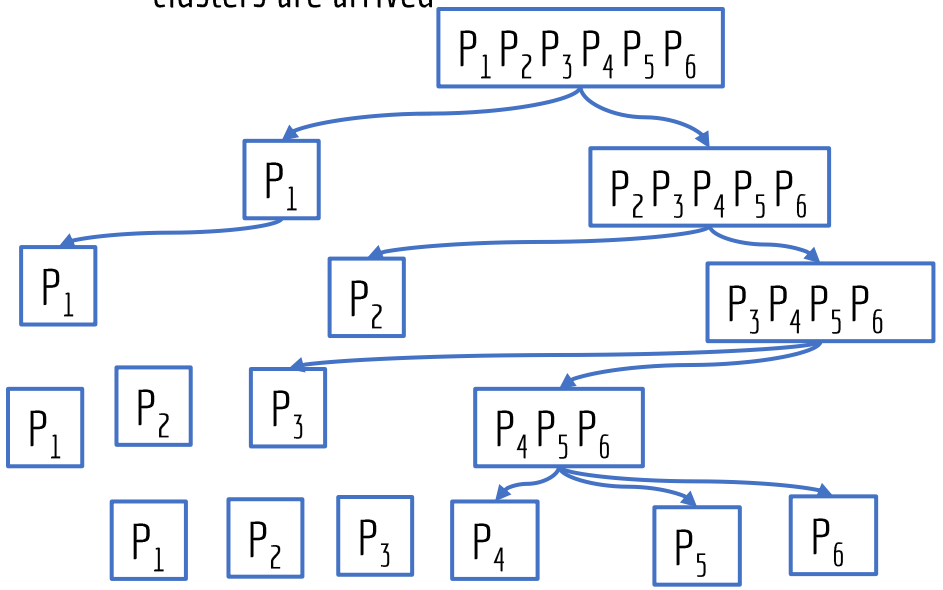
- Distance between two clusters is the distance between centroids



1. Start with an assumption that each data point is cluster
2. At each iteration similar clusters are merged
3. Stop till either all data points are clustered into a single cluster or K desired cluster are arrived



1. Start with an assumption that all data point belong to one cluster
2. At each iteration separate data point more dissimilar to rest of data points in the cluster
3. Stop till either all data points are separately clustered or K desired clusters are arrived



Single Linkage Example : merge in each step the two clusters, whose two closest members have the smallest distance.

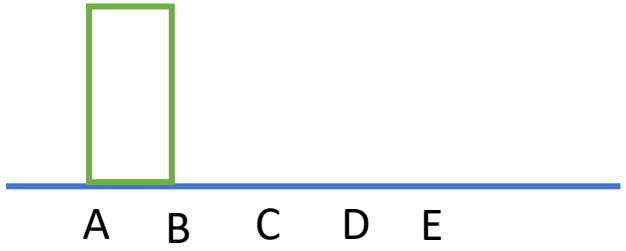
	A
A	7
B	10
C	20
D	28
E	35

Generate Distance matrix ->

	A	B	C	D	E
A	0				
B	3	0			
C	13	10	0		
D	21	18	18	0	
E	28	25	15	7	0

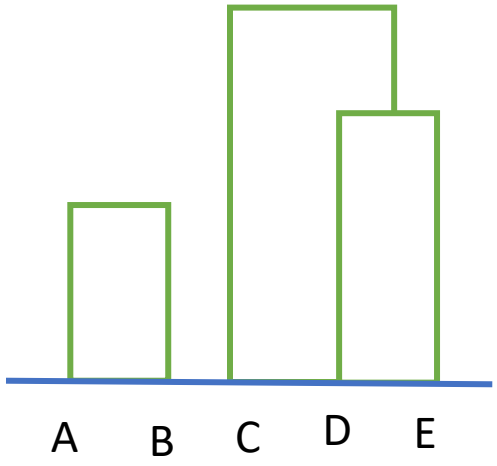
Cluster with min distance ->

	(A,B)	C	D	E
(A,B)	0			
C	10	0		
D	18	8	0	
E	25	15	7	0



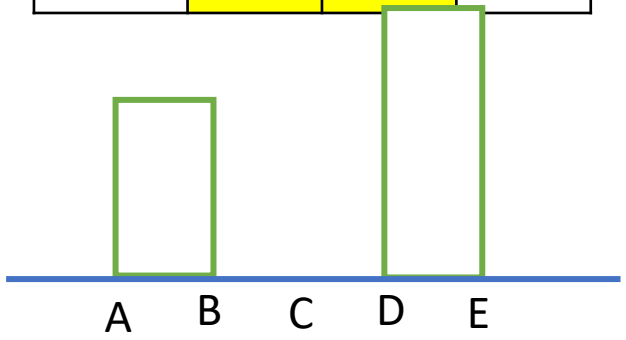
Cluster with min distance ->

	(A,B)	(C,D,E)
(A,B)	0	10
(C,D,E)	10	0

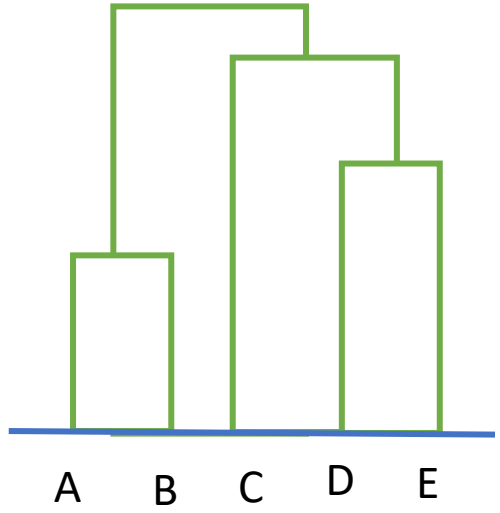


Cluster with min distance ->

	(A,B)		(D,E)
(A,B)	0		
C	10	0	
(D,E)	18	8	0



Final ->



Complete Linkage : Merge cluster on max distance

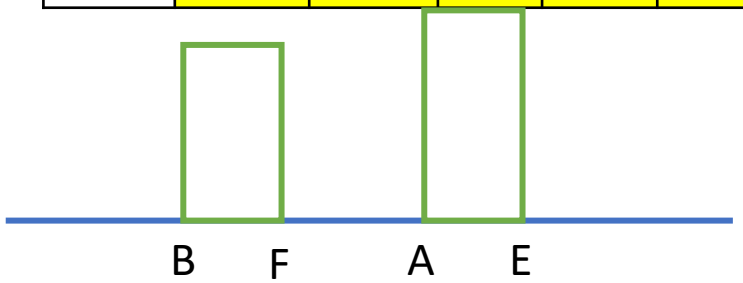
	A	B	C	D	E	F	G
A	0	5					
B	5	0					
C	4.3	7.1	0				
D	10	8.3	1	0			
E	2.5	6.7	4.3	10	0		
F	6.2	2	6.7	8	7.8	0	
G	3.7	7.8	3.3	8.6	3.8	7.5	0

1. Start wit pair of points closest to each other -› B & F are close cluster (B,F)

2. Compute Dissimilarity between (B,F) and rest of points
3. min again is between A and E; hence group them (A,E)

	A	(B,F)	C	D	E	G
A	0	6.2				
(B,F)	6.2	0				
C	4.3	7.1	0			
D	10	8.3	10	0		
E	2.5	7.8	4.3	10	0	
G	3.7	7.8	3.3	8.6	3.8	0

1. Dissimilarity metric between already grouped points and other points have to computed
2. Based on the choice of determining the similarity/dissimilarity between cluster and a data points results in different hierarchal clustering techniques
3. complete linkage : dissimilarity is obtained as the maximum distance between each point within the group and the rest of points



2. Compute Dissimilarity between (B,F) and (A,E) -› Max (A-›(B,F) and E(B,F))-›Max(6.2,7.8) -› 7.8
3. Between (A,E) and rest of point sis minimum criterion, ex. (A-›C & E-›C) Min((4.3,4.3) = 4.3

	(A,E)	(B,F)	C	D	G
(A,E)	0				
(B,F)	7.8	0			
C	4.3	7.1	0		
D	10	8.3	10	0	
G	3.8	7.8	3.3	8.6	0

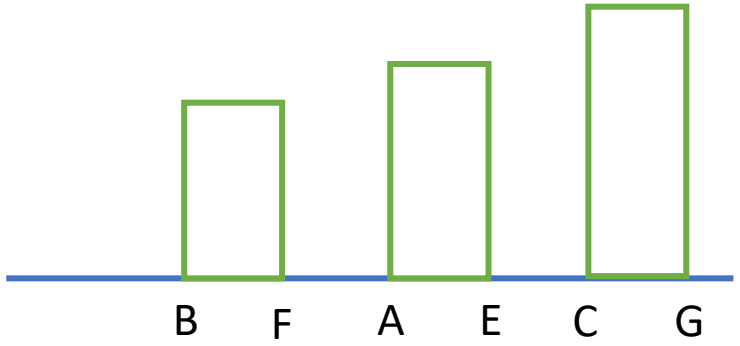
	(A,E)	(B,F)	C	D	G
(A,E)	0				
(B,F)	7.8	0			
C	4.3	7.1	0		
D	10	8.3	10	0	
G	3.8	7.8	3.3	8.6	0

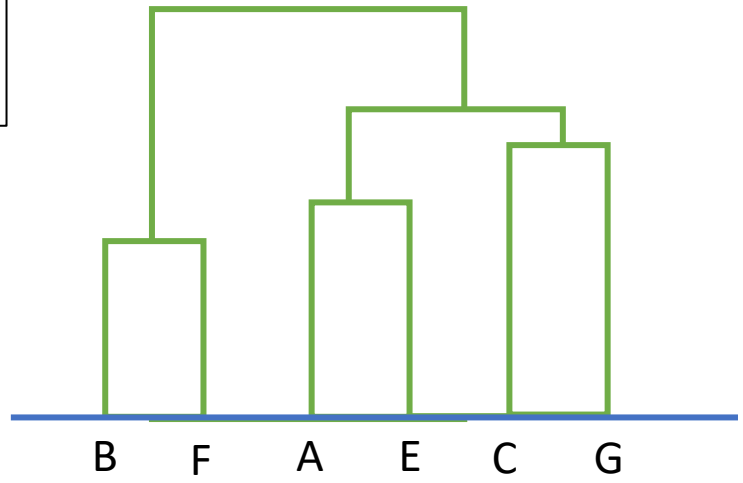
Using the principle of complete linkage join the groups at a node, groups with min dissimilarity.

Join Min distance groups treating them as points.

	(A,E,C,G)	D
(A,E,C,G,B,F)	0	10
D	10	0

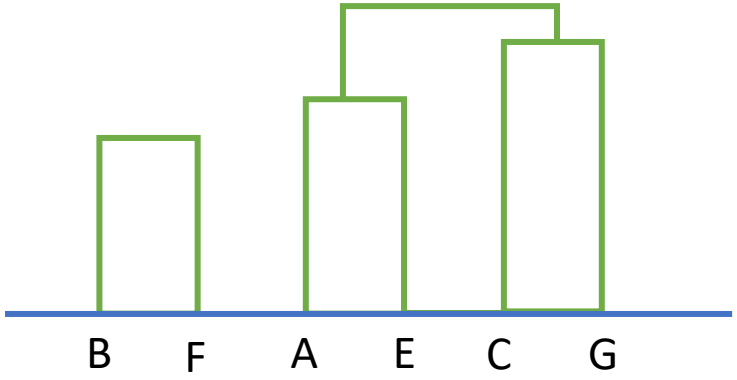
	(A,E)	(B,F)	(C,G)	D
(A,E)	0	7.8		
(B,F)	7.8	0		
(C,G)	4.3	7.8	0	
D	10	8.3	10	0





Proceed further by considering the groups (A,E),(B,F), (C,G) and D
Min is between (A,E) and (C,G); hence group them

	(A,E,C,G)	(B,F)	D
(A,E,C,G)	0	7.8	10
(B,F)	7.8	0	8.3
D	10	8.3	0

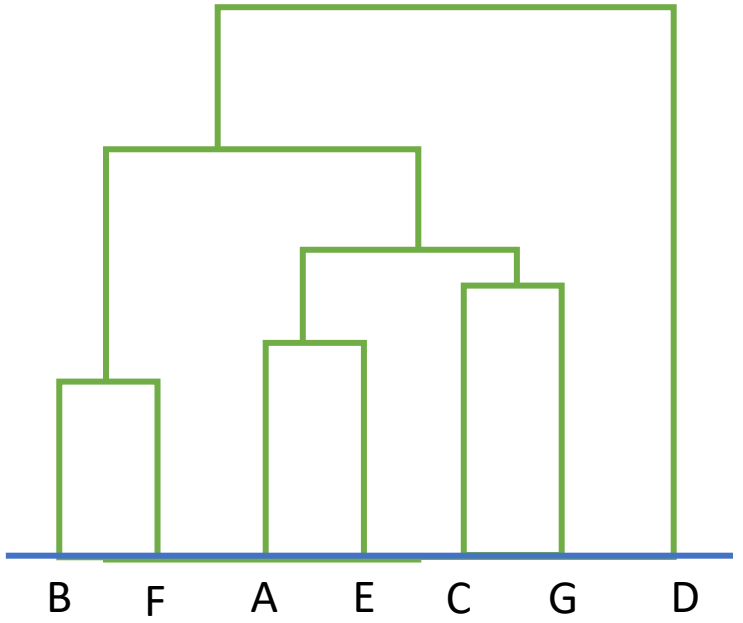


Join Min distance groups treating them as points.

Join the last data point

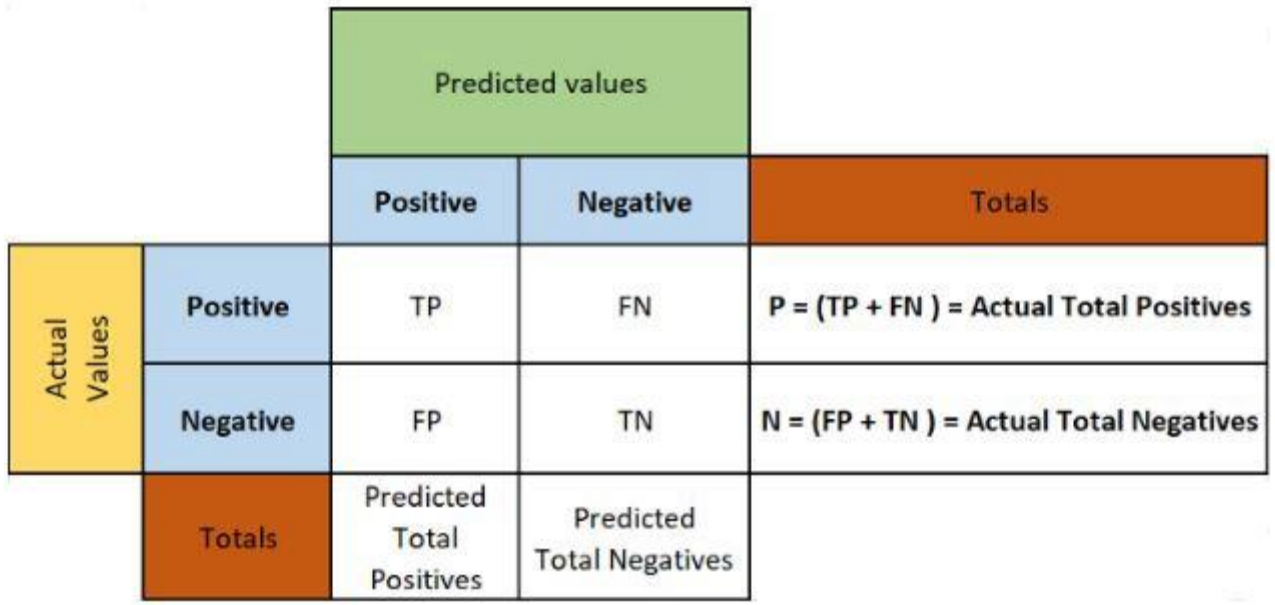
	(A,E,C,G)	D
(A,E,C,G,B,F)	0	10
D	10	0

	(A,E,C,G,B,F,D)
(A,E,C,G,B,F,G)	0



Metrics for Similarity in Hierarchical clustering

- MIN
- Max
- Average
- Distance between centroid
- Ward


Predicted values
Positive
Negative
Totals
Actual Values
Positive
TP
FN
P = (TP + FN) = Actual Total Positives
Negative
FP
TN
N = (FP + TN) = Actual Total Negatives
Totals
Predicted Total Positives
Predicted Total Negatives

Data Preparation

Scaling

Values of different features can differ by orders of magnitude

the larger values dominate the smaller values in computation

Scaling overcomes the problem of different ranges for features

Scaling features to a range is a common choice of range between zero and one.

Strategies in scaling

Standardization:

- Subtract the mean of a feature and divide by the standard deviation.
- the feature values are normally distributed, we'll get a Gaussian, centered around zero with a variance of one.

Label encoding

replace each label with an integer

Problem : ML conclude that there's an order.

Asia and North America in the preceding case differ by 4 No real significance!!

Label	Encoded Label
Africa	1
Asia	2
Europe	3
South America	4
North America	5
Other	6

One hot encoding

use dummy variables to encode categorical features.

dummy variables have binary values (0,1) or (T,F)
need as many dummy variables as there
are unique labels minus one.

	is_africa	is_asia	is_europe	is_sam	is_nam
Africa	**1**	0	0	0	0
Asia	0	**1**	0	0	0
Europe	0	0	**1**	0	0
South America	0	0	0	**1**	0
North America	0	0	0	0	**1**
Other	0	0	0	0	0

Input Feature Standardization:

Method 1 :

- Subtract the mean of a feature and divide by the standard deviation.
- the feature values are normally distributed, we'll get a Gaussian, centered around zero with a variance of one.

Method 2:

Scaling based on mapping the original range (0.1-0.9) :

$0.8*[(x-x_{min})/(*(x_{max}-x_{min})] + 0.1$ -› maps the original range to the scaled range :

Method 3:

When data is not normally distributed,

remove the median and divide by the interquartile range.

interquartile range is a range between the first and third quartile (or 25th and 75th percentile).

Method 4:
transform numerical features into a normal distribution.
Example : Box-Cox Transform

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^{\lambda}-1}{\lambda} & if \lambda \neq 0, \\ 1n(y_i) & if \lambda = 0, \end{cases}$$

Method 5:
Binning :separate feature values into several bins, binarize the values, so that a true value is obtained if the precipitation value isn't zero and a false value otherwise
binning process inevitably leads to loss of information

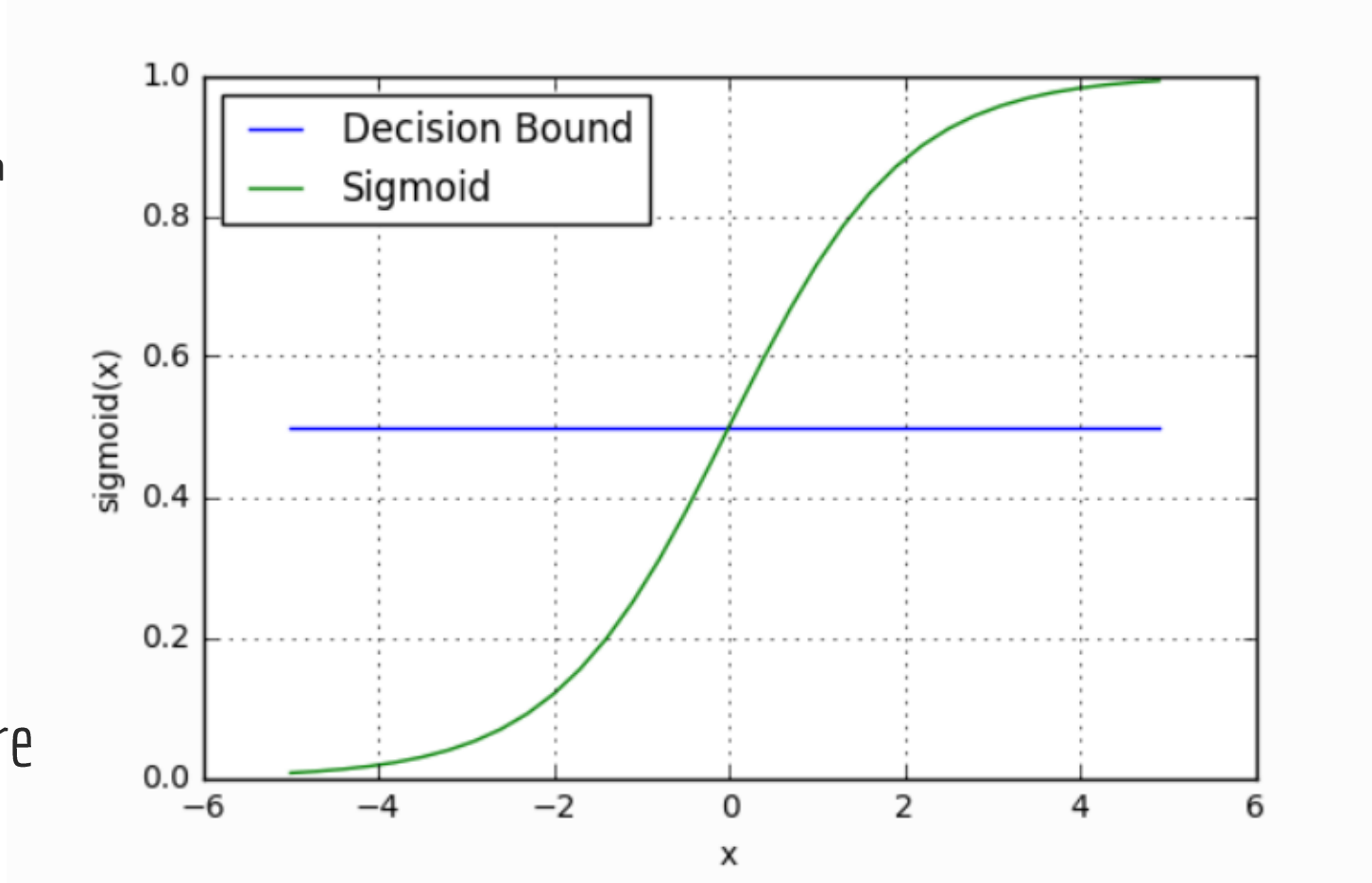
Voting and averaging : The predicted values

logistic regression : split the datapoints to accurately predict a class to which given observation belongs to using the information present in the features.

Model
S(z)=1/(1+e^-z)
•S(z) = Output between 0 and 1 (probability estima
•z = Input to the function (z= mx + b)

Classification Rule:
p>=0.5,class=A
p<=0.5,class=B

line with y=0.5 is called the decision boundary.



Decision Boundaries span through the entire feature space of the training samples

Method	Ouput variable	Pros	Cons
Linear regression	Continuous	•Easy to implement and interpret •Provides prediction along the range if real numbers	•Strong statistical assumptions •Homoscedasticity of data, residual independence, etc
Logistic regression	Continuous in range [0,1]	•Low variance •Provides probability of outcomes •Works well with diagonal decision boundaries	•High bias
Decision trees	Continuous or discrete	•Easy to interpret visually •Can easily handle categorical features •Works well with boundaries parallel to feature axis	•Prone to overfitting
k-means	Discrete	•Works well with large amount of data •Easy to implement and interpret	•Poor performance for non-hyper-spherical clusters •Results depend on selection of K