

# Classification

# Logistic Regression

# Logistic Regression

- Logistic regression is kind of like linear regression but is used when the dependent variable is not a number, but something else (like a Yes/No response)
- Its called Regression but performs classification as based on the regression it classifies the dependent variable into either of the classes

# Logistic Regression

- Logistic regression is used for prediction of output which is binary
- For example, if a credit card company is going to build a model to decide whether to issue a credit card to a customer or not, it will model for whether the customer is going to “Default” or “Not Default” on this credit card.

# Logistic Regression

# K-Nearest Neighbors (K-NN)

# K-Nearest Neighbors (K-NN)

- it is used to identify the data points that are separated into several classes to predict the classification of a new sample point
- K-NN is a **non-parametric**, lazy learning algorithm
- It classifies new cases based on a similarity measure (e.g. distance functions)

# K-Nearest Neighbors (K-NN)

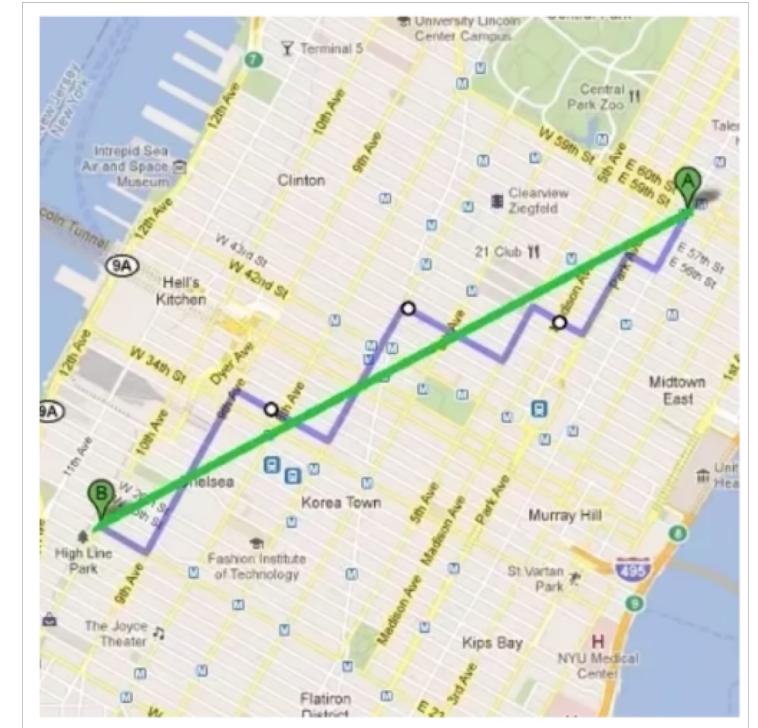
- K is the number of neighbors to the consider
- Scaling is important
- K should be odd in order to avoid the ties
- Voting can be weighted by the distance to each neighbor
- Does not scale to large data well

# K-Nearest Neighbors (K-NN)

- K-Nearest Neighbor does not learn
- It is lazy and just memorizes the data
- Works well with a small number of input variables but struggles when the number of inputs is very large

# K-Nearest Neighbors (K-NN)

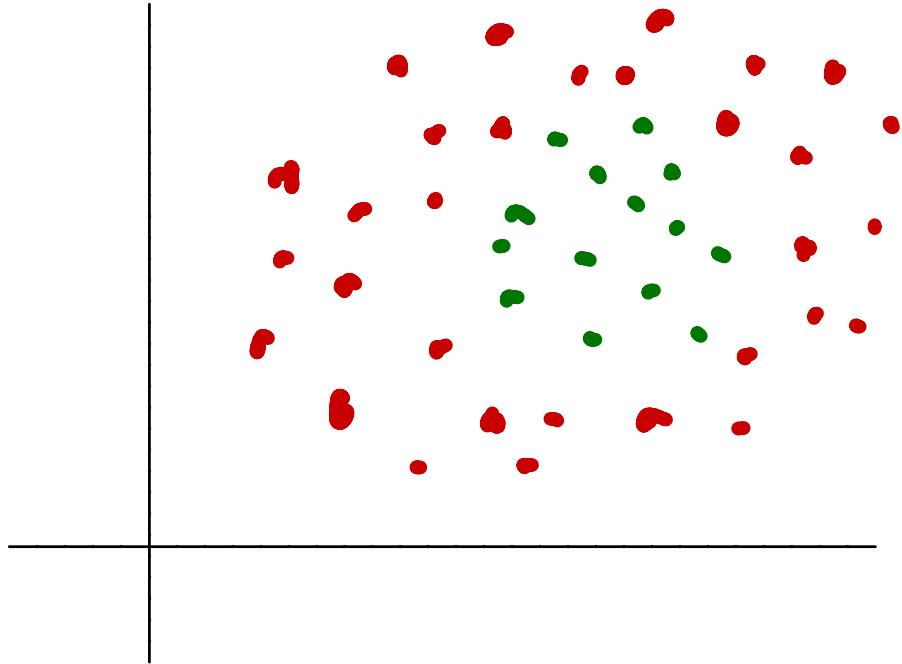
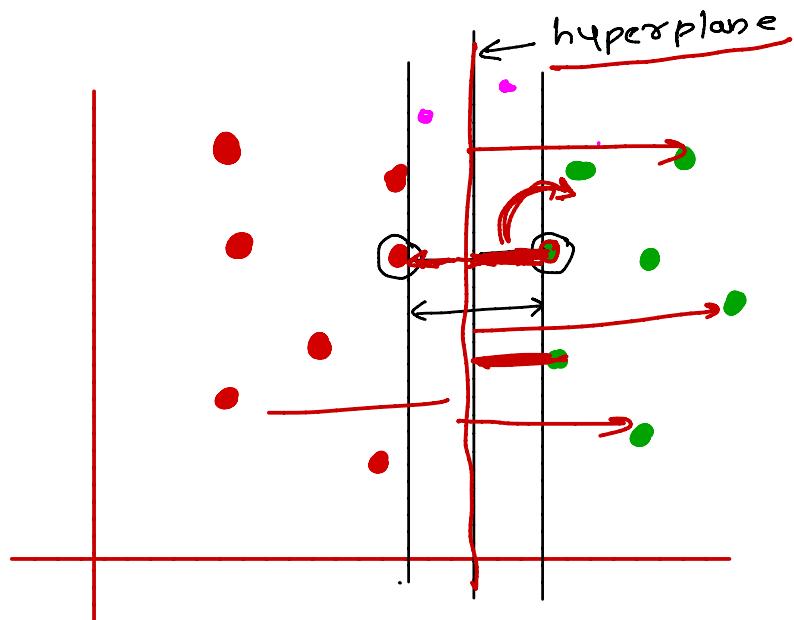
- Distance Algorithms
  - Euclidean distance
  - Manhattan distance

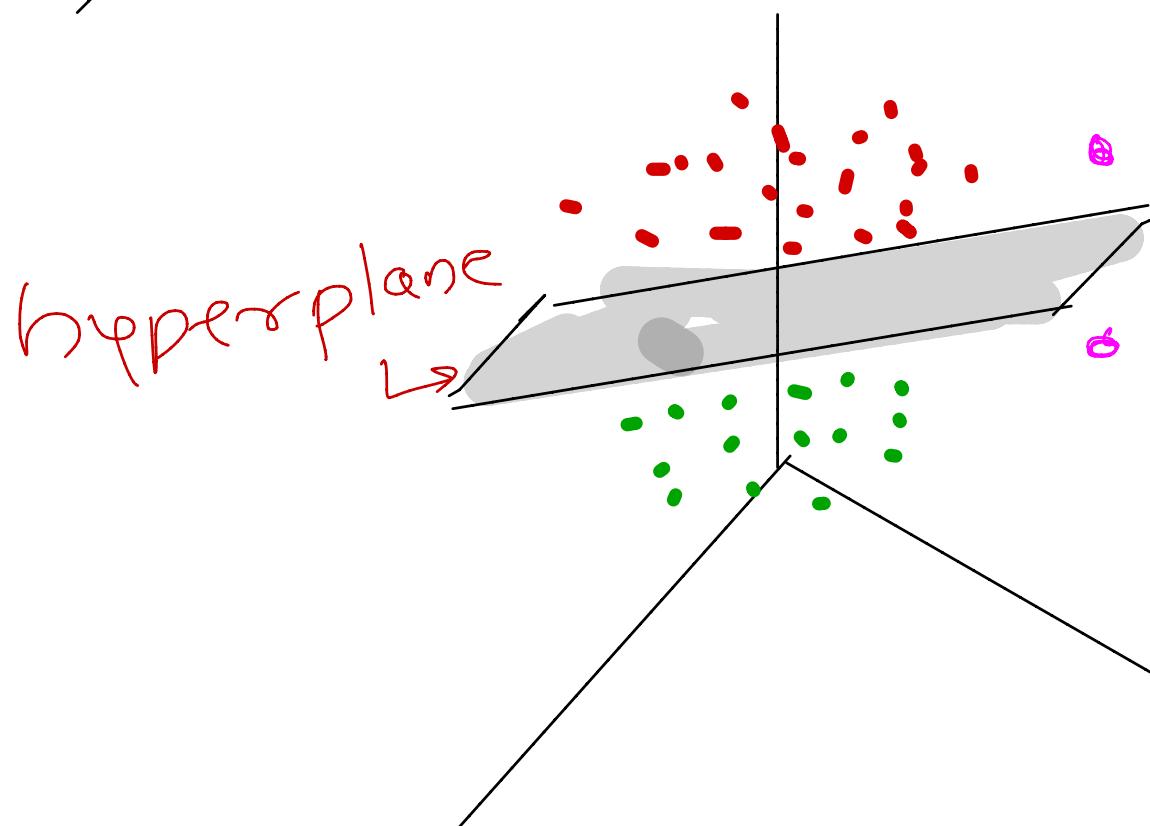
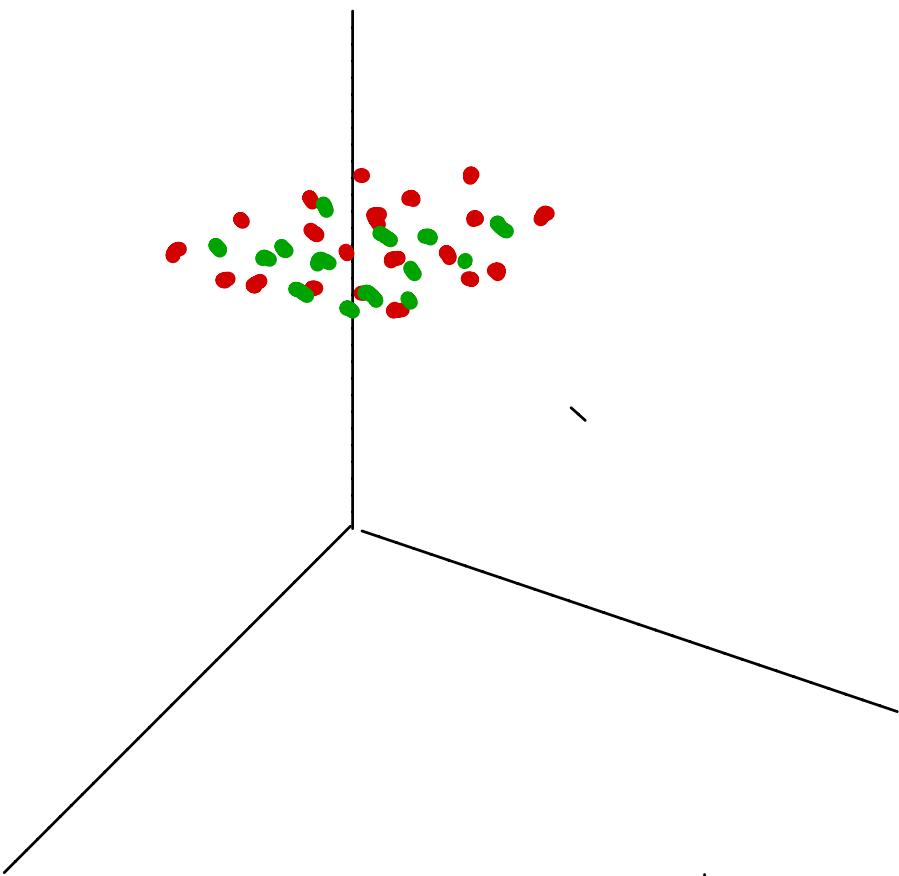


# Support Vector Machine

# Support Vector Machine

- used for both regression and Classification
- It is based on the concept of decision planes that define decision boundaries /
- A decision plane(hyperplane) is one that separates between a set of objects having different class memberships
- It performs classification by finding the hyperplane that maximizes the margin between the two classes with the help of support vectors.



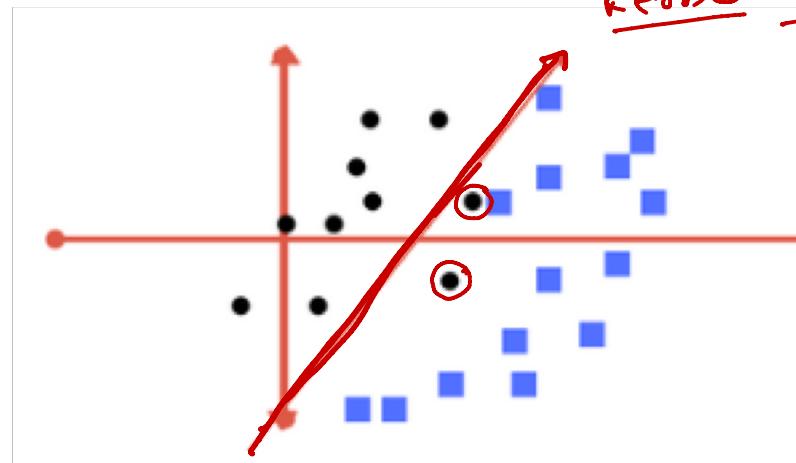


# Support Vector Machine - Terminologies

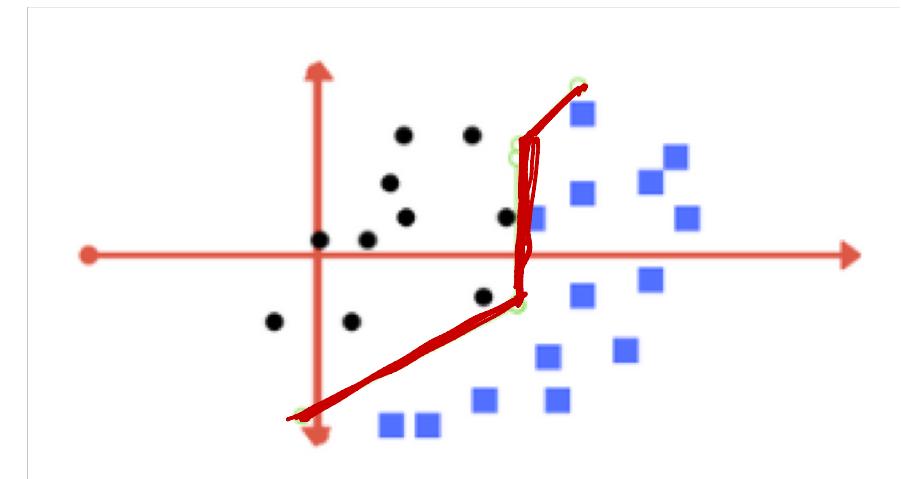
- **Regularization**
  - The Regularization parameter tells the SVM optimization how much you want to avoid misclassifying each training example
  - For large values of  $C$ , the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly
  - Conversely, a very small value of  $C$  will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points

# Support Vector Machine - Terminologies

- Regularization



Low Regularization value



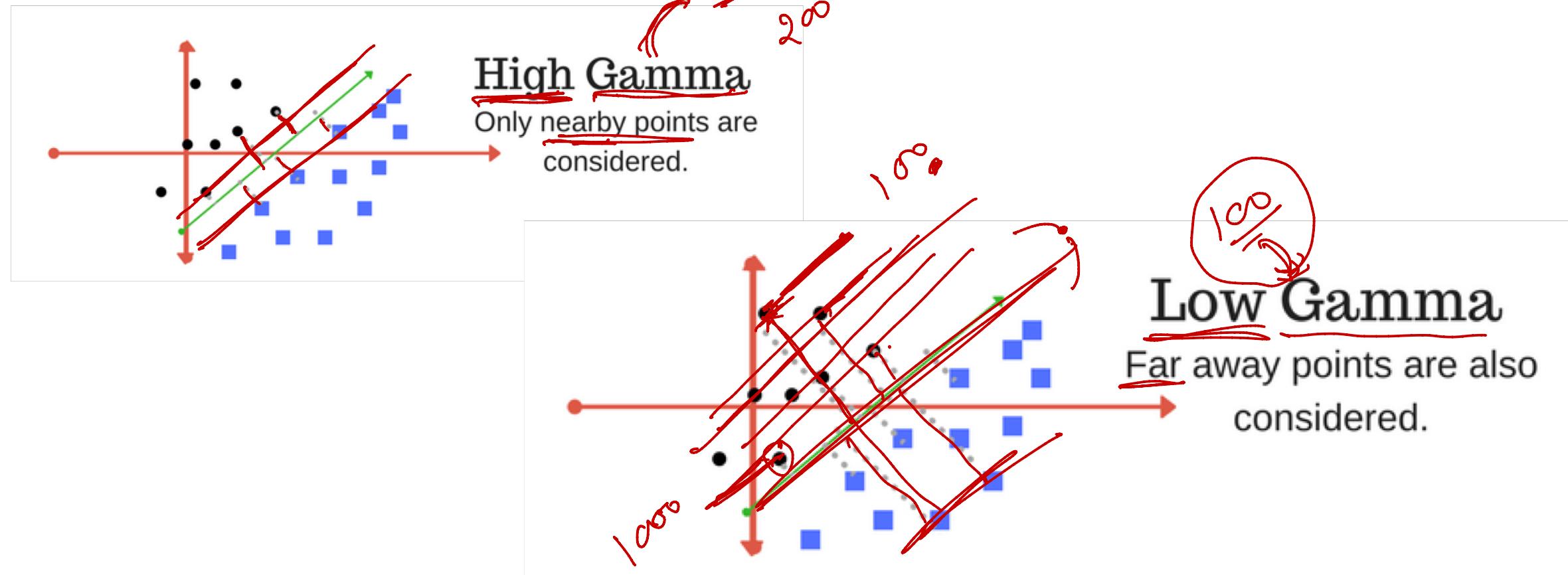
High Regularization value

# Support Vector Machine - Terminologies

- Gamma
  - The gamma parameter defines how far the influence of a single training example reaches
  - low values meaning 'far'
    - points far away from plausible separation line are considered in calculation for the separation line
  - high values meaning 'close'
    - the points close to plausible line are considered in calculation

# Support Vector Machine - Terminologies

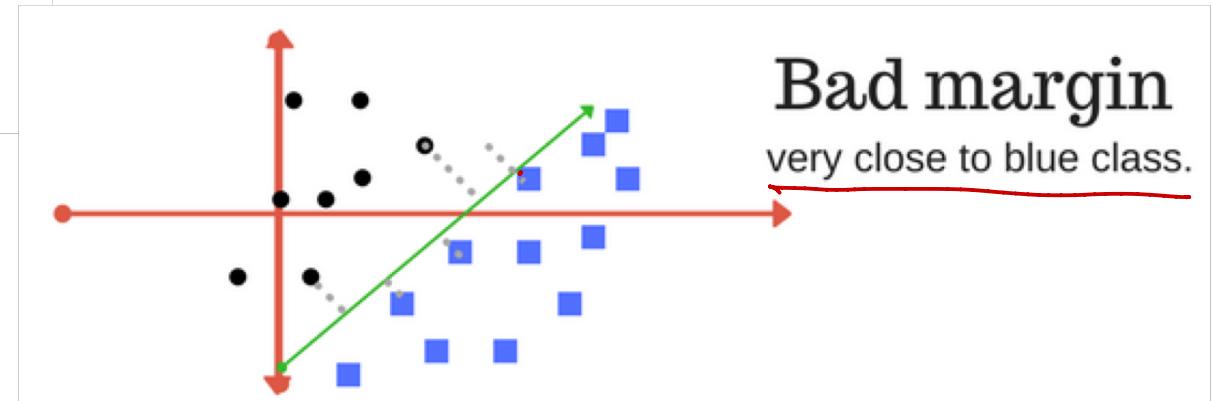
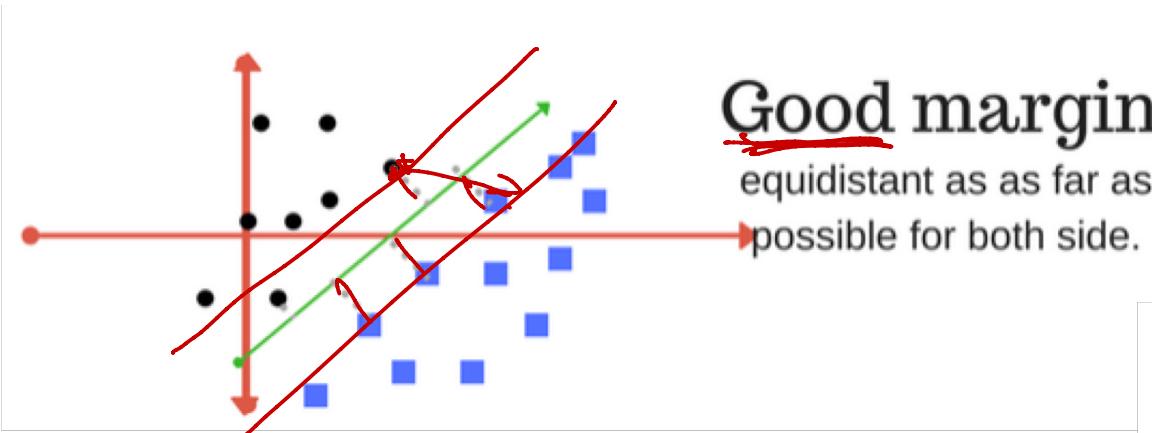
- Gamma



# Support Vector Machine - Terminologies

- Margin

- A margin is a separation of line to the closest class points
- A good margin is one where this separation is larger for both the classes



# Support Vector Machine - Terminologies

- Kernel

- The learning of the hyperplane in linear SVM is done by transforming the problem using some linear algebra

- The function which does the transformation is called as kernel

- • For **linear kernel**

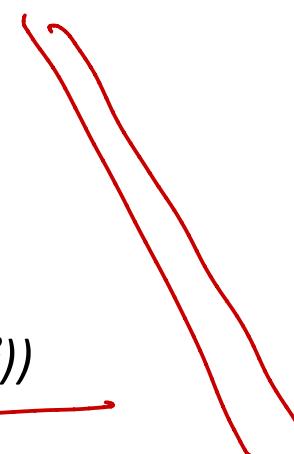
- $f(x) = B(0) + \sum(a_i * (x, x_i))$

- For **polynomial kernel**

- $K(x, x_i) = 1 + \sum(x * x_i)^d$

- For **exponential kernel**

- $K(x, x_i) = \exp(-\gamma * \sum((x - x_i)^2))$



# SVM - Advantages

- High Dimensionality
  - SVM is an effective tool in high-dimensional spaces, which is particularly applicable to document classification and sentiment analysis where the dimensionality can be extremely large.

# SVM - Advantages

- **Memory Efficiency**
  - Since only a subset of the training points are used in the actual decision process of assigning new members, just these points need to be stored in memory (and calculated upon) when making decisions.

# SVM - Advantages

- **Versatility**
  - Class separation is often highly non-linear. The ability to apply new kernels allows substantial flexibility for the decision boundaries, leading to greater classification performance.

# SVM - Disadvantages

- **Kernel Parameters Selection**

- SVMs are very sensitive to the choice of the kernel parameters
- In situations where the number of features for each object exceeds the number of training data samples, SVMs can perform poorly

# Decision Tree



# Decision Tree

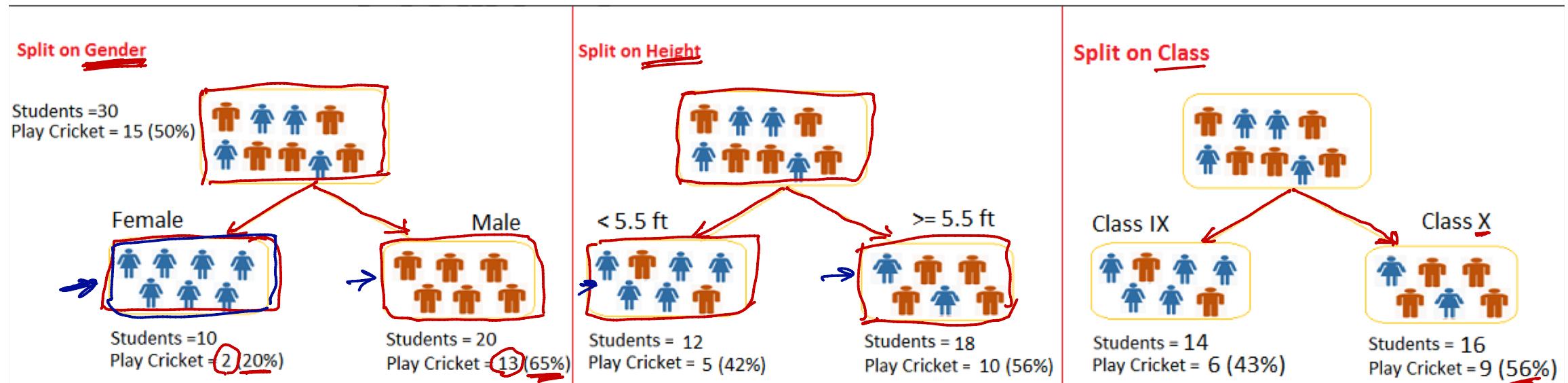
- used mostly in classification
- we split the population or sample into two or more homogeneous sets (or sub-populations) based on most significant splitter / differentiator in input variables

# Decision Tree

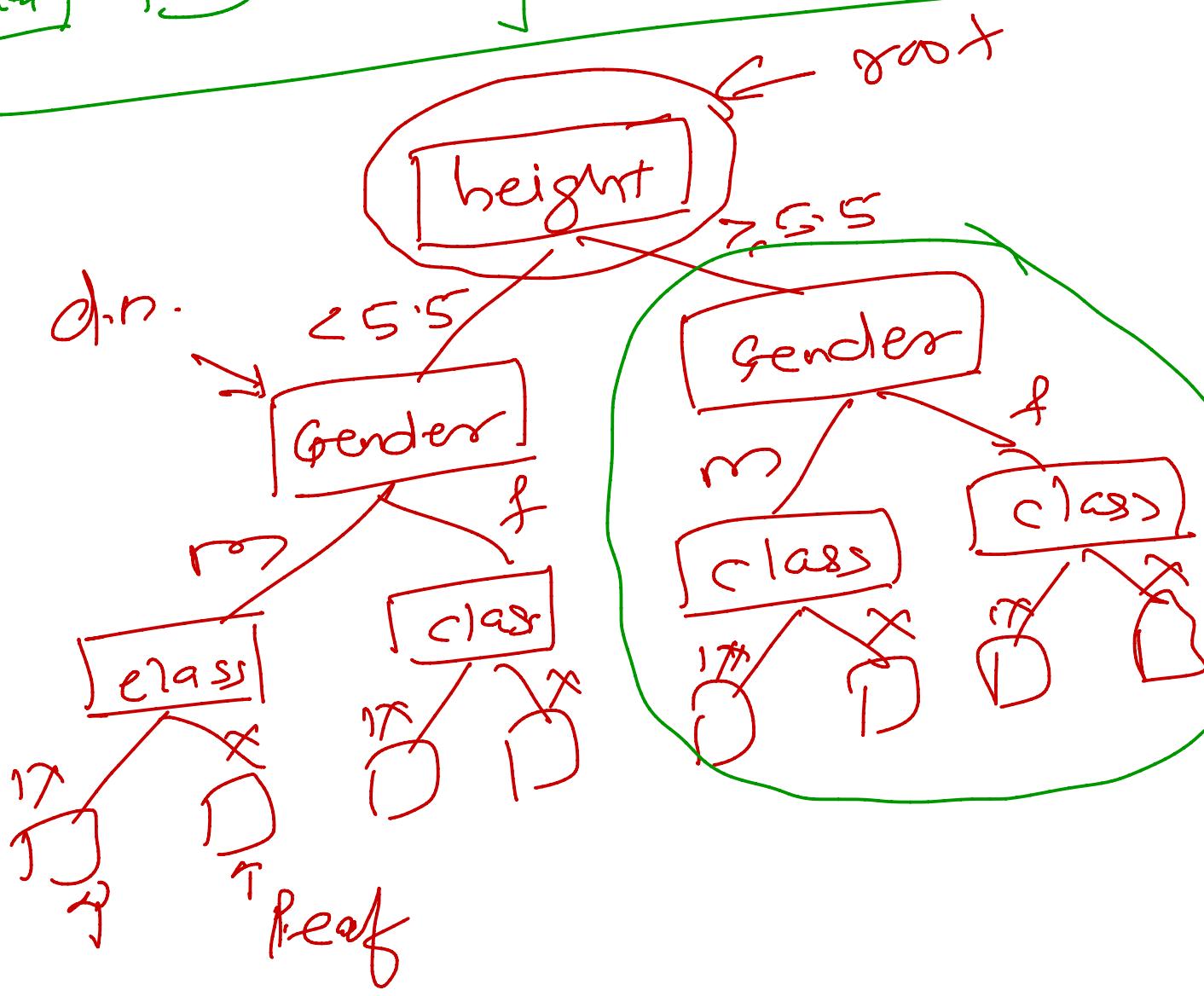
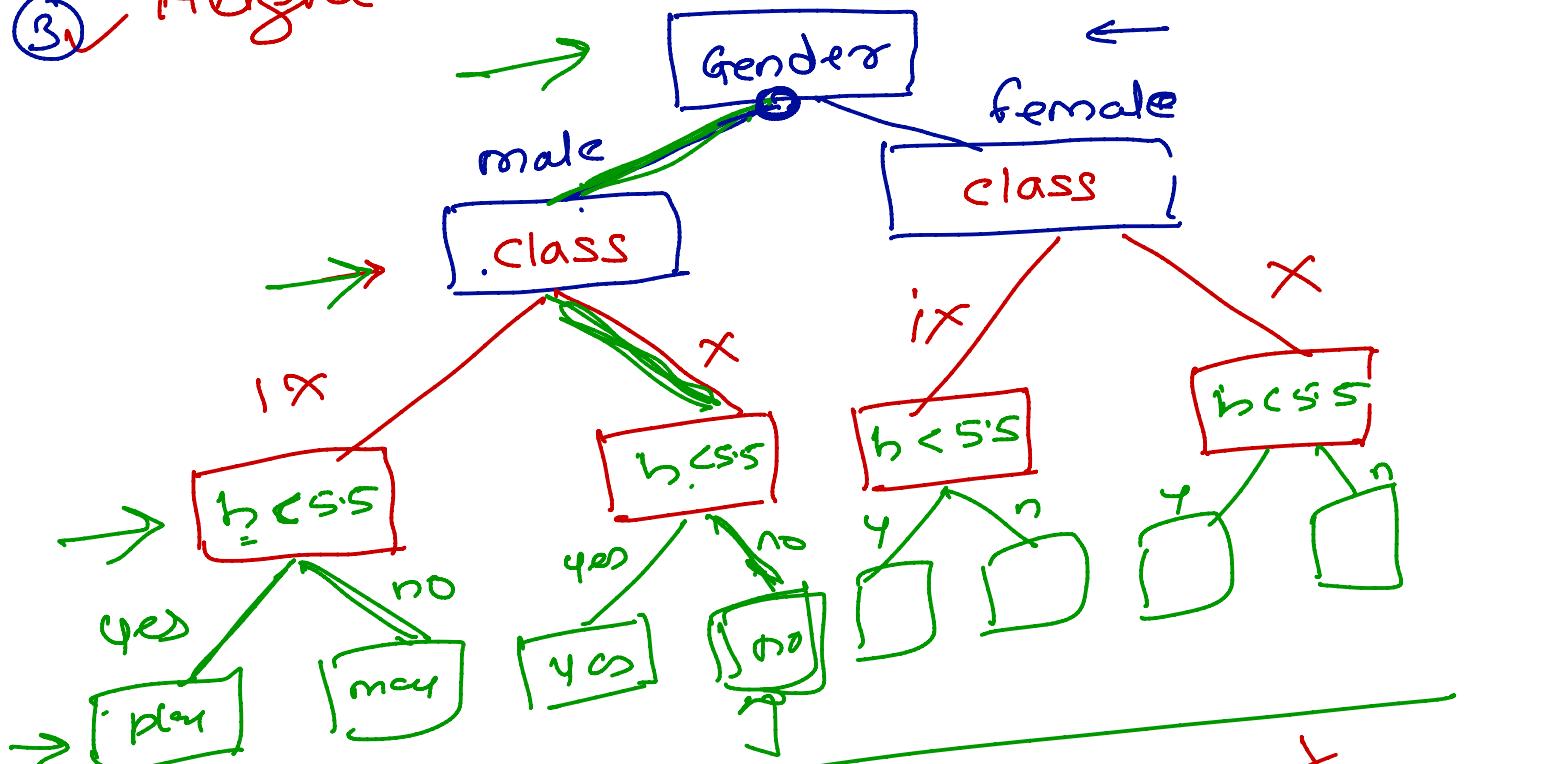
- Let's say we have a sample of 30 students with three variables Gender (male/female), Class(IX/X) and Height (5 to 6 ft). 15 out of these 30 play cricket in leisure time. 50% ↓
- Consider following data
  - Male players: 10 and female players: 20
    - 20% of female students and 65% of male students play cricket
  - Students < 5.5ft: 12 and  $\geq 5.5ft: 18$ 
    - 42% of cricket players are below 5.5ft and 56% are above or equal 5.5ft
  - Class IX: 14, X: 16
    - 43% of class IX and 56% of class X play cricket

# Decision Tree

- variable Gender is able to identify best homogeneous sets compared to the other two variables



- ① Gender ✓
  - ② Class ✓
  - ③ Height ✓



# Decision Tree - Types

- **Categorical Variable Decision Tree**

- Decision Tree which has categorical target variable then it called as categorical variable decision tree
  - E.g. Student will play cricket or not: "Yes" or "NO"

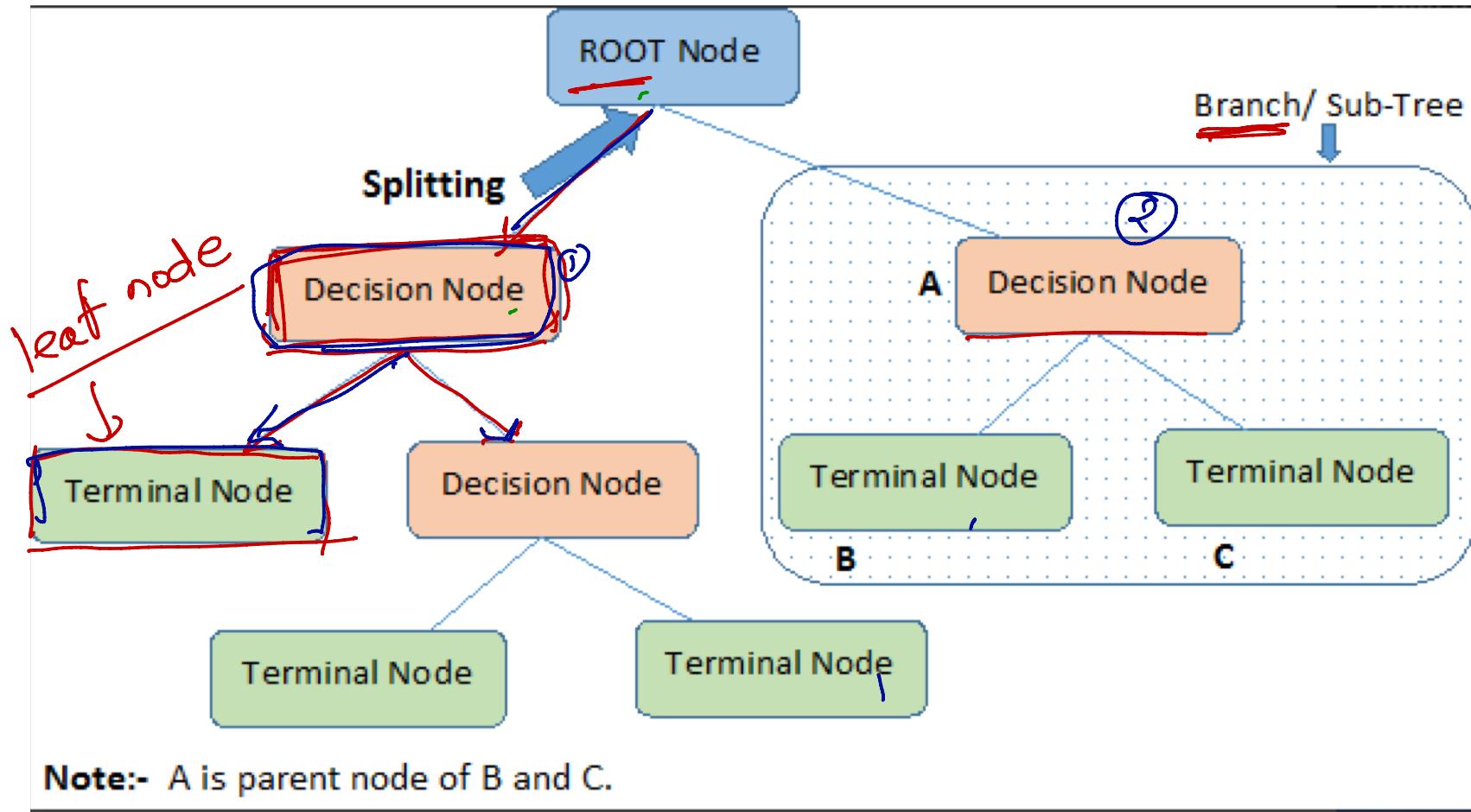
- **Continuous Variable Decision Tree**

- Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree
  - E.g. Age

# Decision Tree - Algorithms

- CART
  - Classification and Regression Trees ✓ →
- ID3
  - Iterative Dichotomiser 3 →
- CHAID
  - Chi-square Automatic Interaction Detector →

# Decision Tree - Terminologies



# Decision Tree - Terminologies

- **Root Node**
  - It represents entire population or sample and this further gets divided into two or more homogeneous sets
- **Splitting**
  - It is a process of dividing a node into two or more sub-nodes
- **Decision Node**
  - When a sub-node splits into further sub-nodes, then it is called decision node
- **Leaf/ Terminal Node**
  - Nodes do not split is called Leaf or Terminal node

# Decision Tree - Terminologies

- **Pruning**
  - When we remove sub-nodes of a decision node, this process is called pruning
  - It is opposite process of splitting
- **Branch / Sub-Tree**
  - A sub section of entire tree is called branch or sub-tree
- **Parent and Child Node**
  - A node, which is divided into sub-nodes is called parent node of sub-nodes where as sub-nodes are the child of parent node

# Decision Tree - Terminologies

- Gini Index

- if we select two items from a population at random then they must be of same class and probability for this is 1 if population is pure
- It works with categorical target variable “Success” or “Failure”
- It performs only Binary splits
- Higher the value of Gini higher the homogeneity
- CART (Classification and Regression Tree) uses Gini method to create binary splits

# Decision Tree - Terminologies

- **Chi-Square**
  - It is an algorithm to find out the statistical significance between the differences between sub-nodes and parent node
  - It works with categorical target variable “Success” or “Failure”
  - Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node
- Chi-square =  $\sqrt{\frac{(actual - expected)^2}{Expected}}$
- It generates tree called CHAID (Chi-square Automatic Interaction Detector)

# Decision Tree - Terminologies

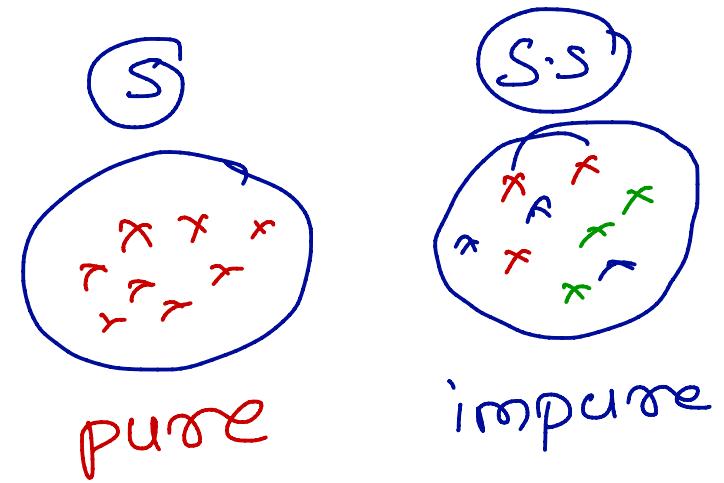
- **Entropy** /

- Metric to measure the impurity of the data

$$\underline{\text{entropy}(s) = -P(\text{yes}) * \log_2 P(\text{yes}) - P(\text{no}) * \log_2 P(\text{no})}$$

Where

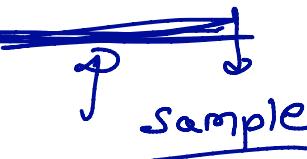
s is total sample space



# Decision Tree - Terminologies

- **Information Gain**
  - Measure of the reduction in entropy
  - Decides which attribute should be selected as decision node
- Information Gain =  $\text{Entropy}(S) - [(\text{Weighted Avg}) * \text{Entropy}(\text{feature})]$

$$\text{Entropy}(S) - [(\text{Weighted Avg}) * \text{Entropy}(\text{feature})]$$

 A handwritten note consisting of a small 'p' followed by a downward arrow and the word 'sample' written below it.

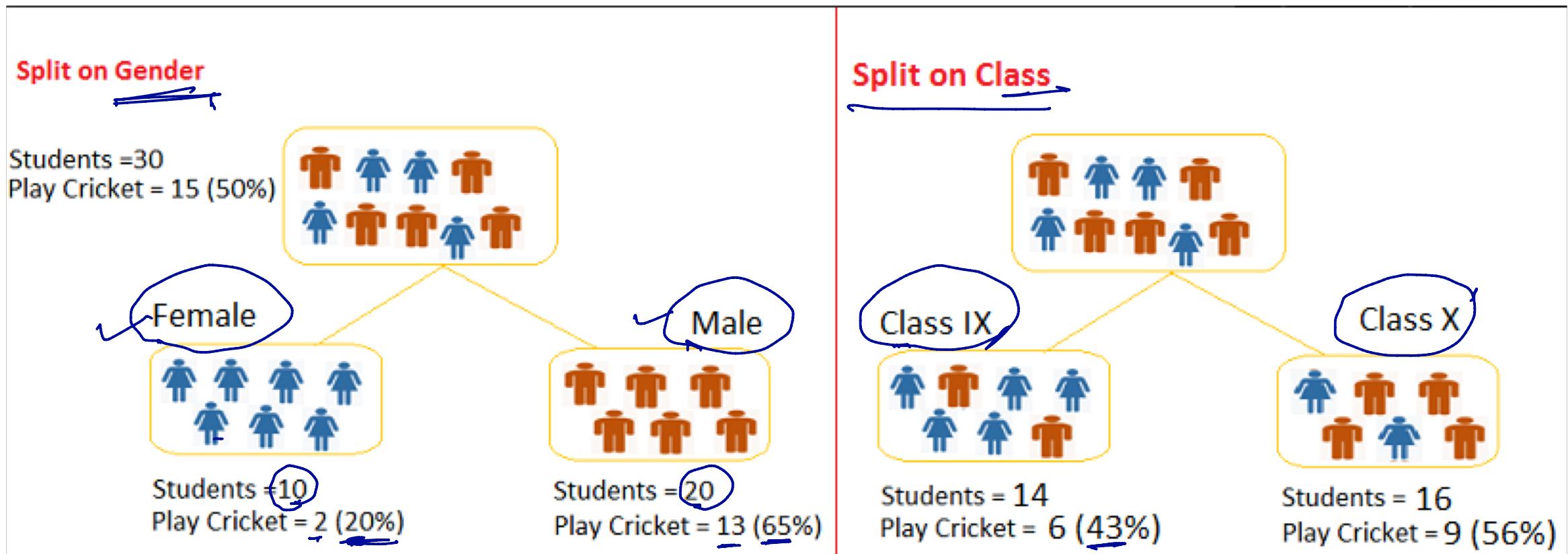
# Gini Index

# Decision Tree – Gini Index

- (use the previous example ☺) we want to segregate the students based on target variable ( playing cricket or not )
- we can split the population using two input variables Gender and Class
- we want to identify which split is producing more homogeneous sub-nodes using Gini index
- Calculate Gini for sub-nodes, using formula
  - sum of square of probability for success and failure ( $p^2+q^2$ )
  - Calculate Gini for split using weighted Gini score of each node of that split

$$\begin{aligned} p &= P(\text{success}) \\ q &= (1-p) \end{aligned}$$

# Decision Tree – Gini Index



# Decision Tree – Gini Index

- **Split on Gender**

- Calculate, Gini for sub-node Female =  $(0.2)*(0.2)+(0.8)*(0.8)=0.68$

- Gini for sub-node Male =  $(0.65)*(0.65)+(0.35)*(0.35)=0.55$

- Calculate weighted Gini for Split Gender =  $(10/30)*0.68+(20/30)*0.55 = 0.59$

$$P(F) \times g(F) + P(M) \times g(M) = 0.59$$

- **Split on Class**

- Gini for sub-node Class IX =  $(0.43)*(0.43)+(0.57)*(0.57)=0.51$

- Gini for sub-node Class X =  $(0.56)*(0.56)+(0.44)*(0.44)=0.51$

- Calculate weighted Gini for Split Class =  $(14/30)*0.51+(16/30)*0.51 = 0.51$

$$P(IX) \times g(IX) + P(X) \times g(X) = 0.51$$

# Chi-Square

# Chi-Square

- Split of Gender

$$\frac{(-3)^2}{5^2} =$$

Node	<u>actual Play Cricket</u>	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
Female	2	8	10	5	5	-3	3	1.34	1.34
Male	13	7	20	10	10	3	-3	0.95	0.95
<b>Total Chi-Square</b>								<b>4.58</b>	

# Chi-Square

- Split of Class

Node	Play Cricket	Not Play Cricket	Total	Expected Play Cricket	Expected Not Play Cricket	Deviation Play Cricket	Deviation Not Play Cricket	Chi-Square	
								Play Cricket	Not Play Cricket
IX	6	8	14	7	7	-1	1	0.38	0.38
X	9	7	16	8	8	1	-1	0.35	0.35
Total Chi-Square								1.46	

# Information Gain

# Information Gain

- We have four X values (outlook, temp, humidity and windy)
- use the attribute with the highest *information gain* to build the tree

	outlook	temp.	humidity	windy	play
1	sunny	hot	high	false	no
2	sunny	hot	high	true	no
3	overcast	hot	high	false	yes
4	rainy	mild	high	false	yes
5	rainy	cool	normal	false	yes
6	rainy	cool	normal	true	no
7	overcast	cool	normal	true	yes
8	sunny	mild	high	false	no
9	sunny	cool	normal	false	yes
10	rainy	mild	normal	false	yes
11	sunny	mild	normal	true	yes
12	overcast	mild	high	true	yes
13	overcast	hot	normal	false	yes
14	rainy	mild	high	true	no

# Information Gain

- Out of 14 instances we have 9 yes and 5 no.
- Calculate Entropy of sample [E(S)]

$$\begin{aligned} E(S) &= -P(\text{yes}) \log_2 P(\text{yes}) - P(\text{no}) \log_2 P(\text{no}) \\ &= -(9/14) * \log_2(9/14) - (5/14) * \log_2(5/14) \\ &= 0.41 + 0.53 \\ &= 0.94 \end{aligned}$$

# Information Gain

$E(\text{outlook} = \text{Sunny})$

$$= -\frac{2}{5} \log \left( \frac{2}{5} \right) - \frac{3}{5} \log \left( \frac{3}{5} \right) = 0.971$$

$E(\text{outlook} = \text{Overcast})$

$$= -\frac{1}{14} \log \left( \frac{1}{14} \right) - \frac{0}{14} \log \left( \frac{0}{14} \right) = 0$$

$E(\text{outlook} = \text{Rainy})$

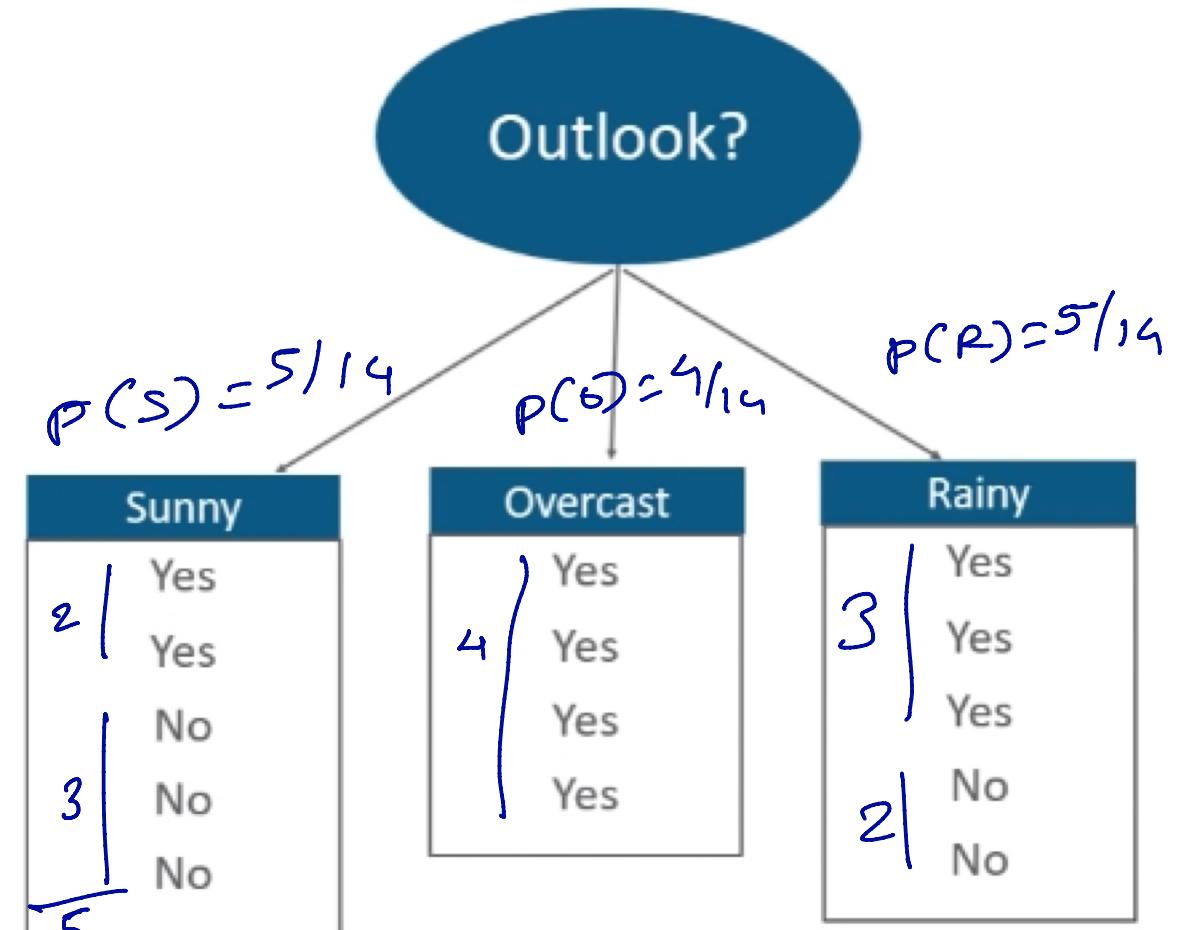
$$= -\frac{3}{5} \log \left( \frac{3}{5} \right) - \frac{2}{5} \log \left( \frac{2}{5} \right) = 0.971$$

Information from Outlook  $I(O)$

$$= \frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.971 = 0.693$$

Information Gain from Outlook

$$= E(S) - I(\text{Outlook}) = 0.94 - 0.693 = 0.247$$



# Information Gain

$$E(\text{windy} = \text{True})$$

$$= -1/2 \log (1/2) - 1/2 \log (1/2) = 1$$

$$E(\text{windy} = \text{False})$$

$$= -6/8 \log (6/8) - 2/8 \log (2/8) = 0.811$$

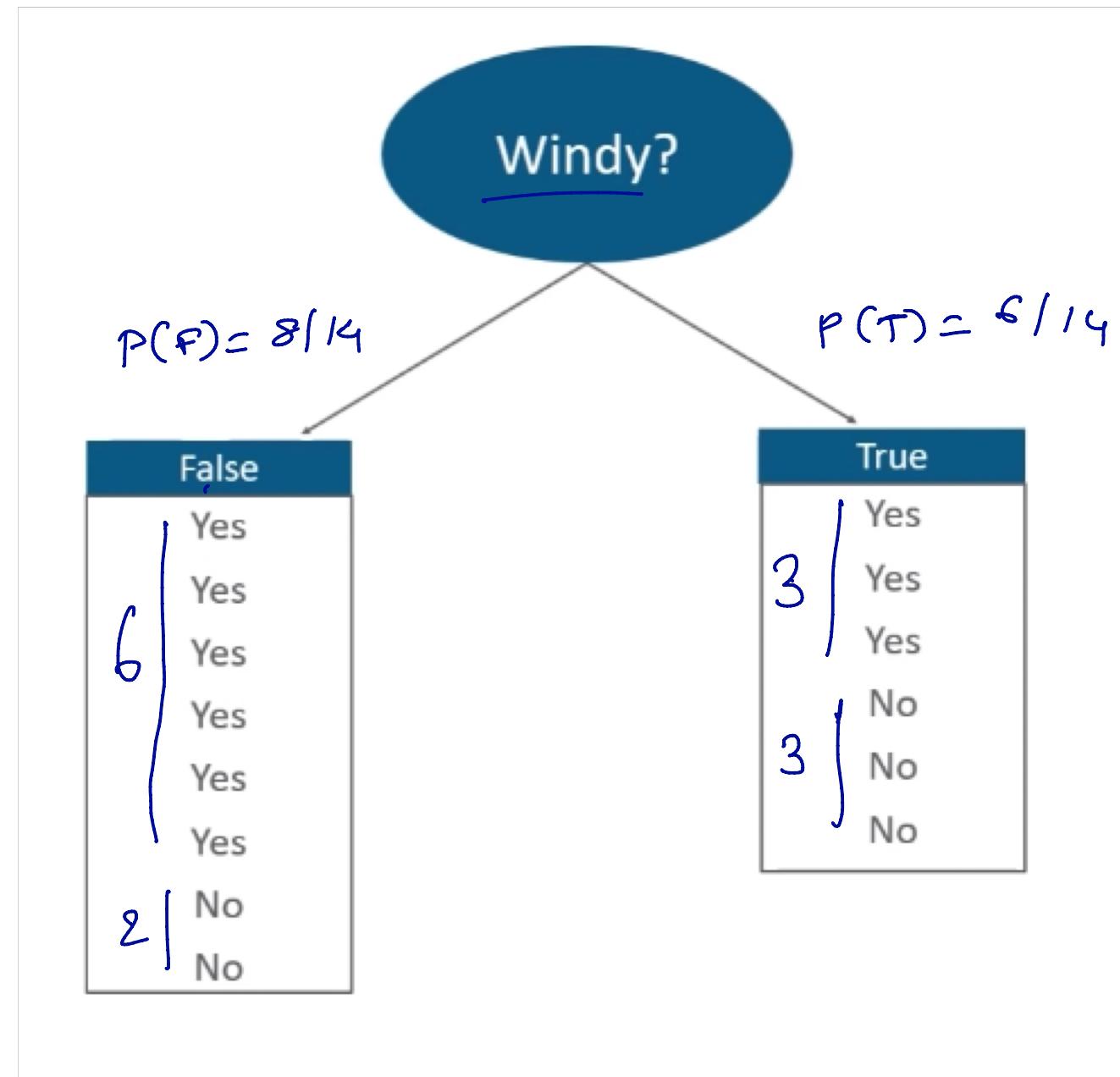
Information from Windy  $I(\text{Windy})$

$$= 8/14 * 0.811 + 6/14 * 1 = 0.892$$

Information Gain from Windy

$$= E(S) - I(\text{windy}) = 0.94 - 0.892 = 0.048$$

$I(\text{windy})$



# Information Gain

## Outlook:

Info

Gain: 0.940-0.693

0.693

0.247

## Temperature:

Info

Gain: 0.940-0.911

0.911

0.029

## Humidity:

Info

Gain: 0.940-0.788

0.788

0.152

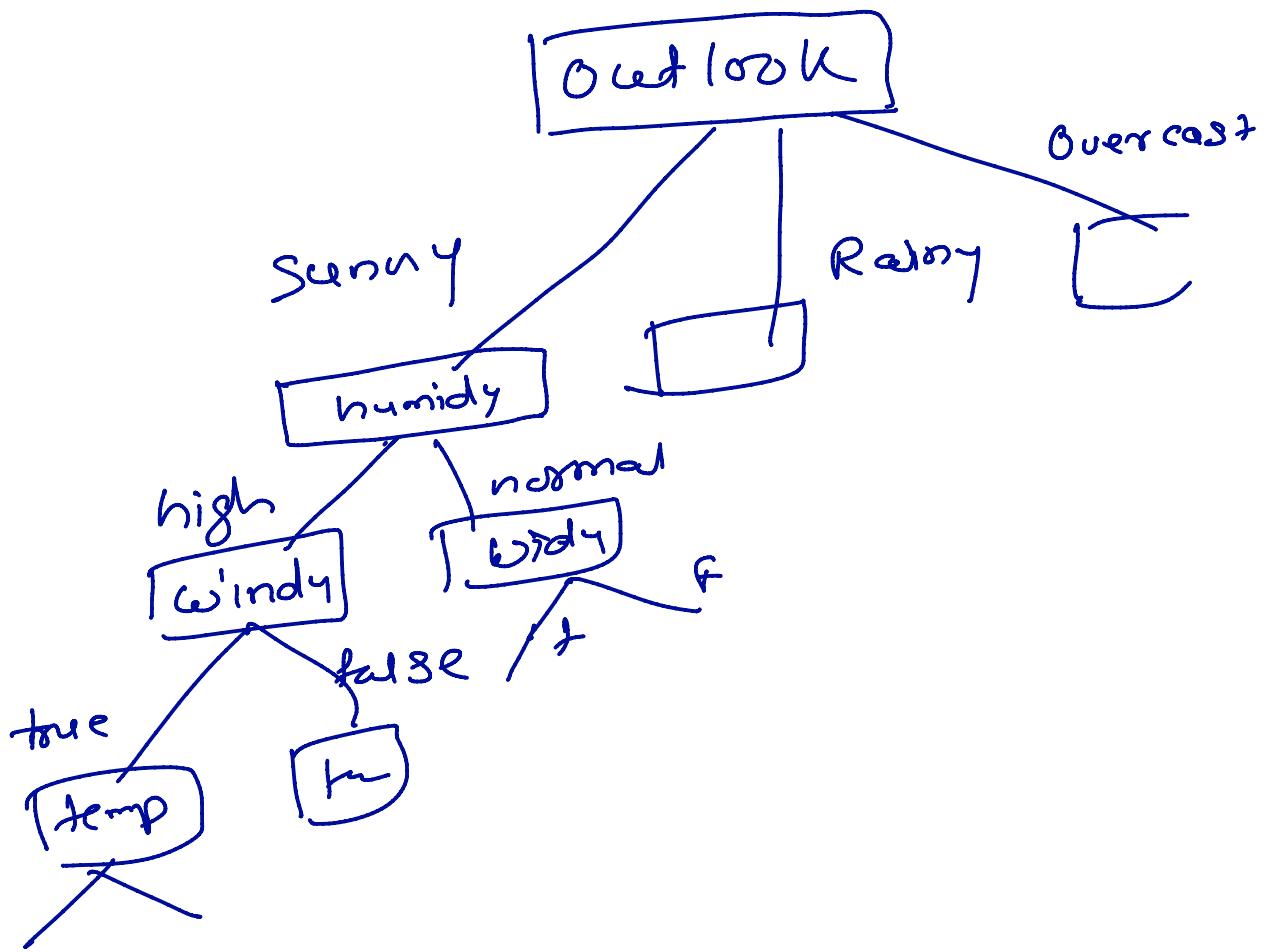
## Windy:

Info

Gain: 0.940-0.982

0.892

0.048



# Decision Tree - Advantages

- **Easy to Understand**

- Decision tree output is very easy to understand even for people from non-analytical background
- It does not require any statistical knowledge to read and interpret them
- Its graphical representation is very intuitive and users can easily relate their hypothesis

- **Less data cleaning required**

- It requires less data cleaning compared to some other modelling techniques
- It is not influenced by outliers and missing values to a fair degree

# Decision Tree - Advantages

- **Useful in Data exploration**
  - Decision tree is one of the fastest way to identify most significant variables and relation between two or more variables
- **Data type is not a constraint**
  - It can handle both numerical and categorical variables

# Decision Tree - Disadvantages

- **Over fitting**
  - This problem gets solved by setting constraints on model parameters and pruning
- **Not fit for continuous variables**
  - While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories

# Decision Tree - Disadvantages

- **Over fitting**
  - This problem gets solved by setting constraints on model parameters and pruning
- **Not fit for continuous variables**
  - While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories

# Decision Tree - Disadvantages

- **Over fitting**
  - This problem gets solved by setting constraints on model parameters and pruning
- **Not fit for continuous variables**
  - While working with continuous numerical variables, decision tree loses information when it categorizes variables in different categories