Basic Terminology in Classification Algorithms

- **Classifier:** An algorithm that maps the input data to a specific category.
- **Classification model:** A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature:** A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification:** Classification task with two possible outcomes. **Eg: Gender classification (Male / Female)**
- **Multi-class classification:** Classification with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. **Eg: An animal can be a cat or dog but not both at the same time.**
- **Multi-label classification:** Classification task where each sample is mapped to a set of target labels (more than one class). **Eg: A news article can be about sports, a person, and location at the same time.**

Applications of Classification Algorithms

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drugs classification
- Facial key points detection

Naïve Bayes Classification

- Based on Bayes theorem
- Assumption
  - Presence of one evidence is independent of other other evidence /feature (naïve)
  - equal contribution to the outcome.
- Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
- **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
- Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.

Why is it called Naïve Bayes?

- **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
- **Bayes**: It is called Bayes because it depends on the principle of Bayes' Theorem.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where,

- **P(A|B) is Posterior probability**: Probability of hypothesis A on the observed event B.
- **P(B|A) is Likelihood probability**: Probability of the evidence given that the probability of a hypothesis is true.
- **P(A) is Prior Probability**: Probability of hypothesis before observing the evidence.
- **P(B) is Marginal Probability**: Probability of Evidence.

Working of Naïve Bayes' Classifier:

- Convert the given dataset into frequency tables.
- Generate Likelihood table by finding the probabilities of given features.
- Now, use Bayes theorem to calculate the posterior probability.

LIKELIHOOD
The probability of "B" being True, given "A" is True

PRIOR
The probability "A" being True. This is the knowledge.

$$P(A|B) = \frac{P(B|A).P(A)}{P(B)}$$

POSTERIOR
The probability of "A" being True, given "B" is True

MARGINALIZATION
The probability "B" being True.

- **Problem**: If the weather is sunny, then the Player should play or not?
- **Solution**: To solve this, first consider the below dataset:

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY GOLF
0	Sunny	Hot	High	Weak	No
1	Sunny	Hot	High	Strong	No
2	Overcast	Hot	High	Weak	Yes
3	Rain	Mild	High	Weak	Yes
4	Rain	Cool	Normal	Weak	Yes
5	Rain	Cool	Normal	Strong	No
6	Overcast	Cool	Normal	Strong	Yes
7	Sunny	Mild	High	Weak	No
8	Sunny	Cool	Normal	Weak	Yes
9	Rain	Mild	Normal	Weak	Yes
10	Sunny	Mild	Normal	Strong	Yes
11	Overcast	Mild	High	Strong	Yes
	Overcast	Hot	Normal	Weak	Yes
	Rain	Mild	High	Strong	No

Outlook

	Yes	No	P(?/yes)	P(?/No)	P()
Sunny	**2**	**3**	**P(Sunny/yes)=2/9**	**P(Sunny/No)=3/5**	**P(Sunny)=5/14**
Overcast	**4**	**0**	**P(Overcast/Yes)=4/9**	**P(Overcast/No)=0**	**P(Overcast)=4/14**
Rainy	**3**	**2**	**P(Rainy/yes)=3/9**	**P(Rainy/No)=2/5**	**P(Rainy)=5/14**
	9	**5**			
	P(Yes) =9/14	**P(No) =5/14**			

Similarily for Temperature

	Yes	**No**	**P(?/yes)**	**P(?/no)**
Hot	**2**	**2**	**2/9**	**2/5**
medium	**4**	**2**	**4/9**	**2/5**
Cool	**3**	**1**	**3/9**	**1/5**
	9	**5**		
	9/14	**5/14**		

Humidity

	Yes	No	P(?/yes)	P(?/no)	P(s)
HIgh	3	4	3/9	4/5	7/14
Normal	6	1	6/9	1/5	7/14
	9	5			
	9/14	5/14			

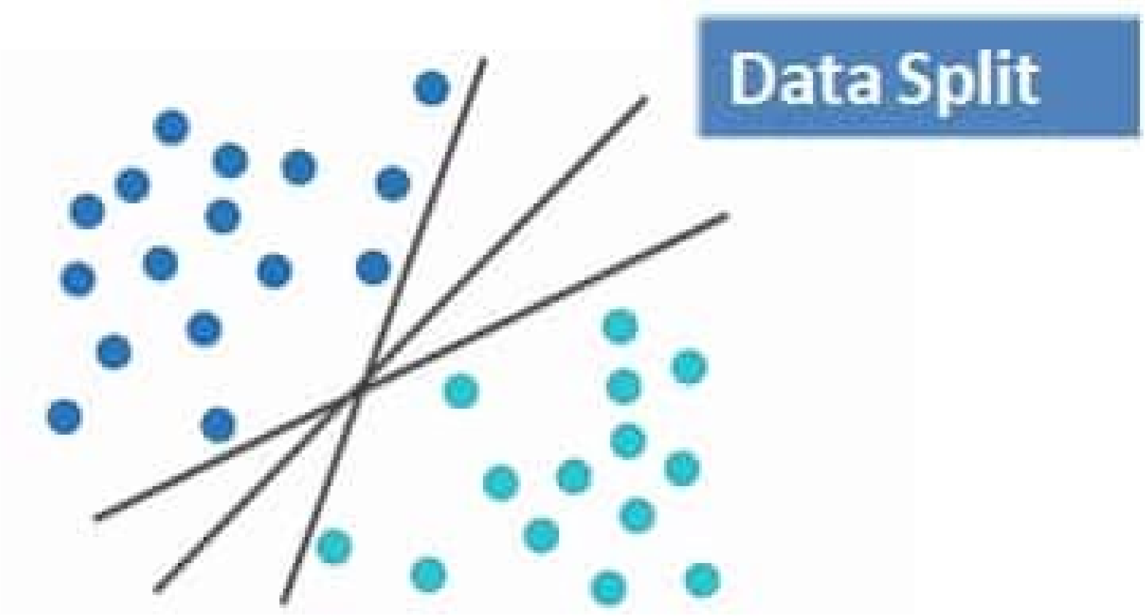
Windy

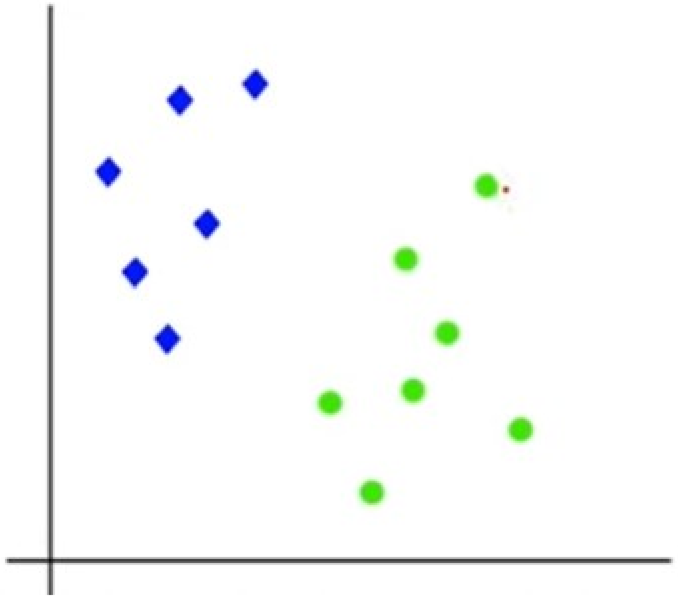
	Yes	No	P(?/yes)	P(?/no)	P()
Weak	6	2	6/9	2/5	8/14
Strong	3	3	3/9	3/5	6/14
	9	5			
	9/14	5/14			

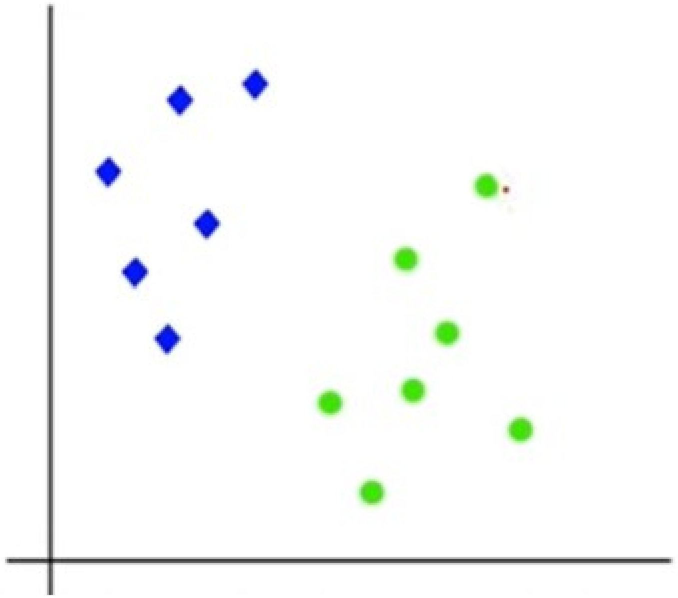
Play	P(yes/no)
Yes	9/14
NO	5/14

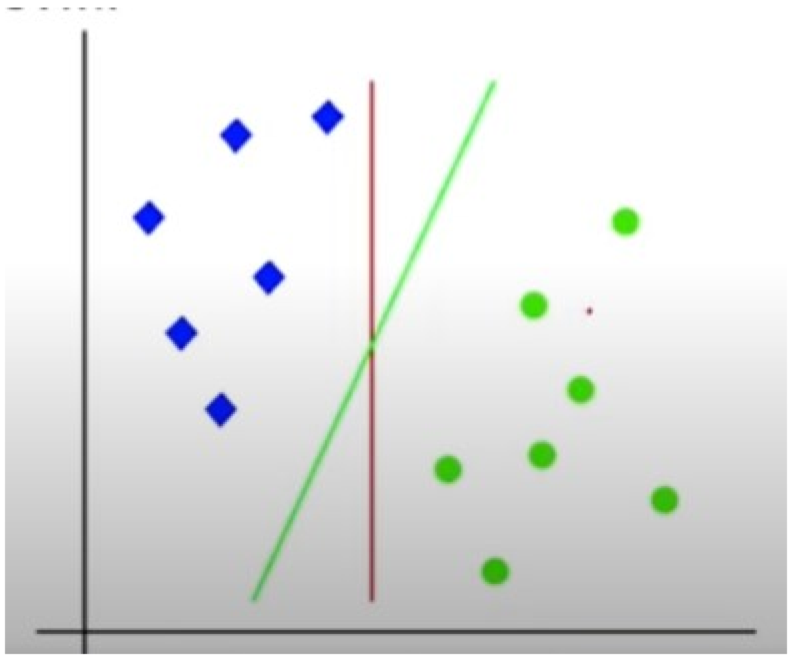
SVM

A **Support Vector Machine (SVM)** is a supervised machine learning algorithm which can be used for both classification and regression problems.





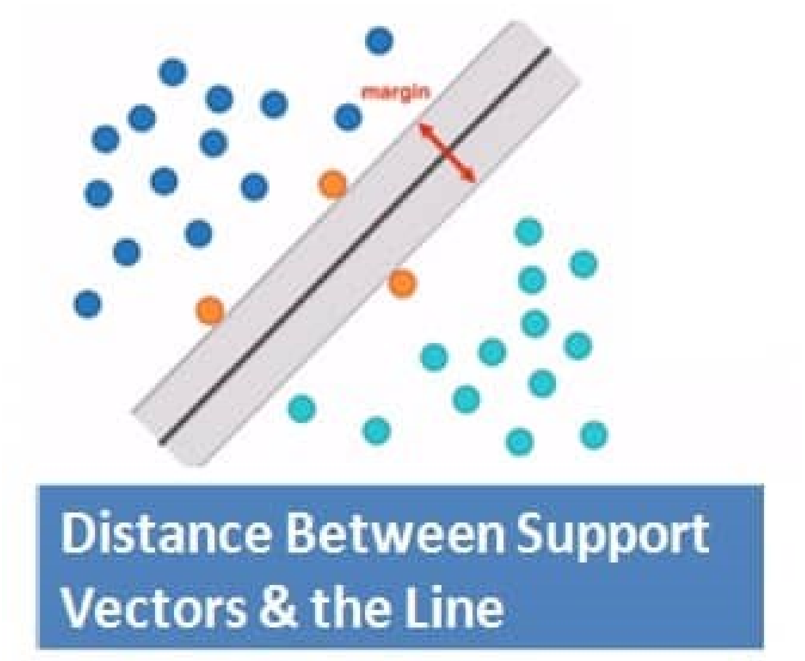


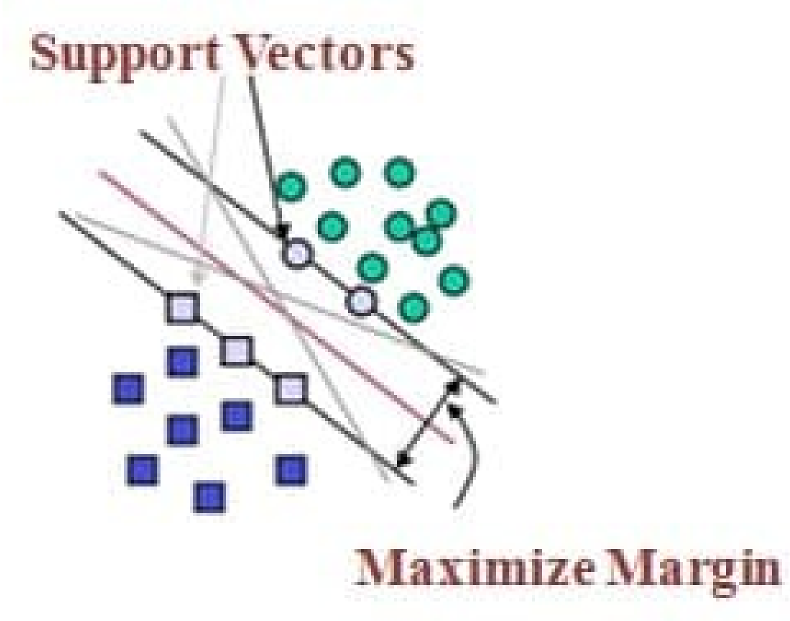


Support Vectors

SVMs maximize the margin around the separating hyperplane.

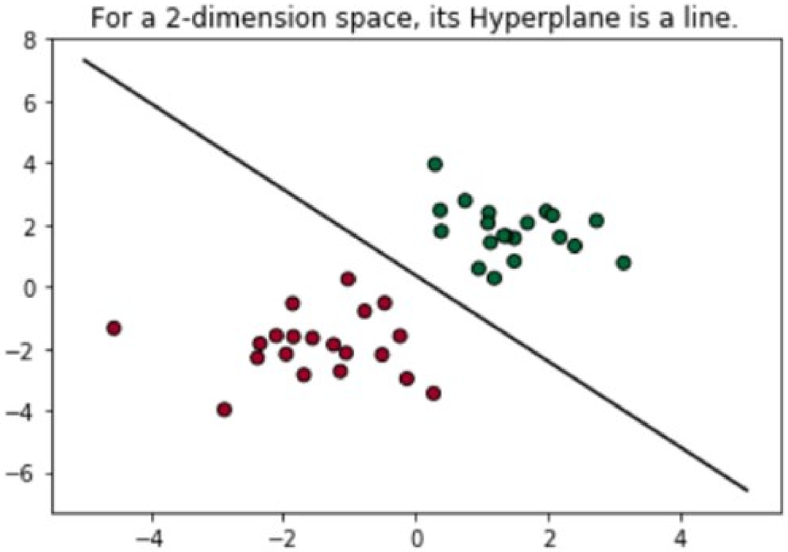
Margins are the (perpendicular) distances between the line and those dots closest to the line.

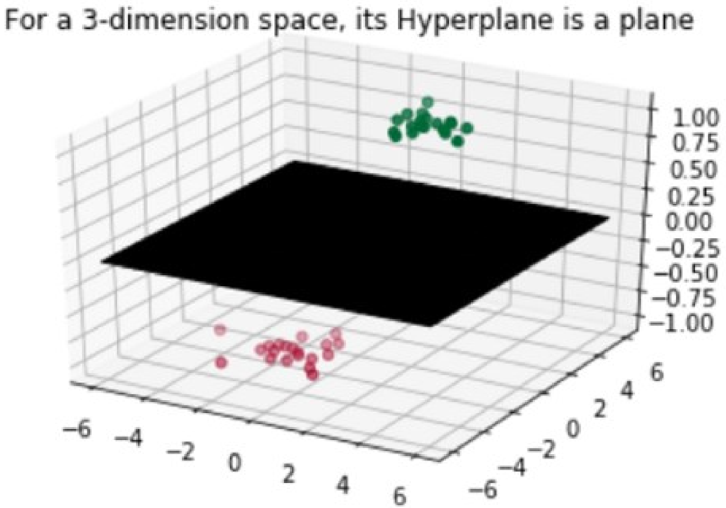




So what is a **Hyperplane**?

- Hyperplane is an (n minus 1)-dimensional subspace for an n-dimensional space
- For a 2-dimension space, its hyperplane will be 1-dimension, which is just a line.
- For a 3-dimension space, its hyperplane will be 2-dimension, which is a plane that slice the cube.



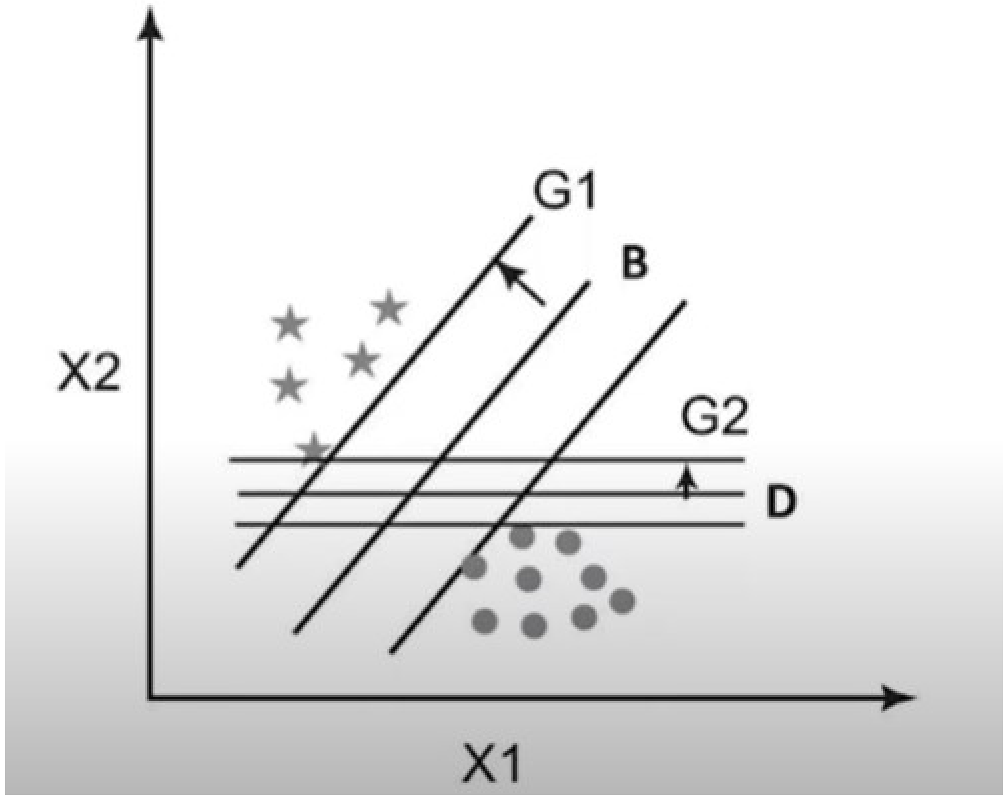


- Any Hyperplane can be written mathematically as below:

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 + \ldots + \beta_n * x_n = 0$$

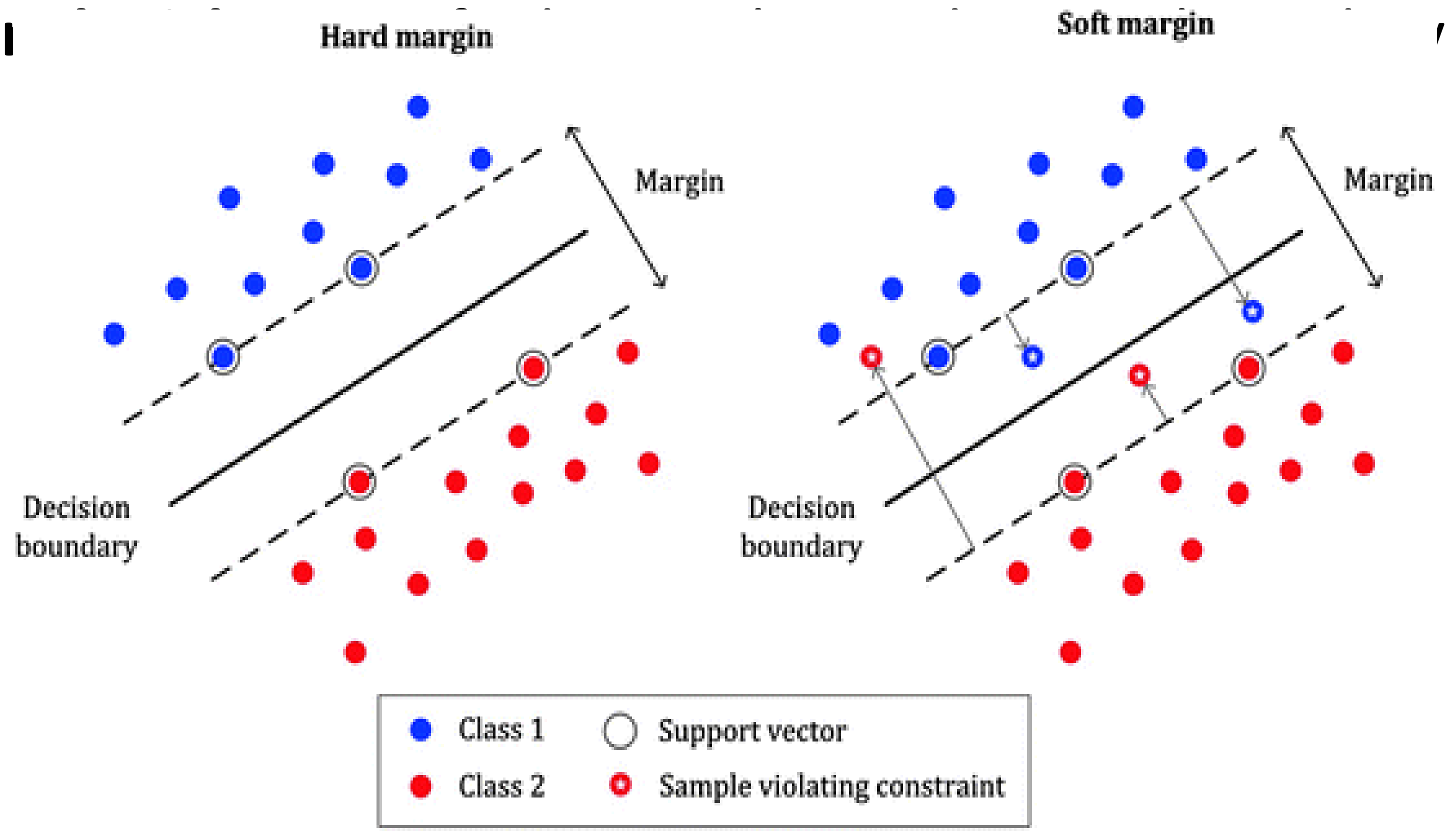
- For a 2-dimensional space, the Hyperplane, which is the line.

$$\beta_0 + \beta_1 * x_1 + \beta_2 * x_2 = 0$$


G1
B
X2
G2
D
X1

Soft Margin Vs. Hard Margin

- **Soft Margin:** try to find a line to separate, but tolerate one or few misclassified dots (e.g. the dots circled in red)
- **Ker**



Comparison with other classifiers

- SVM can be used to separate linear as well as non-linear space (kernel trick)
- Due to maximum margin on both sides it is less likely to result in over-fitting
- Can work with even smaller datasets
- Complexity is linear

- ***Degree of tolerance***
  How much tolerance(soft) we want to give when finding the decision boundary is an important hyper-parameter for the SVM (both linear and nonlinear solutions).
- It is represented as the penalty term — ‘C’.
- The bigger the C, the more penalty SVM gets when it makes misclassification..

we define linear classifier function or Separating classifier hyper-plane as

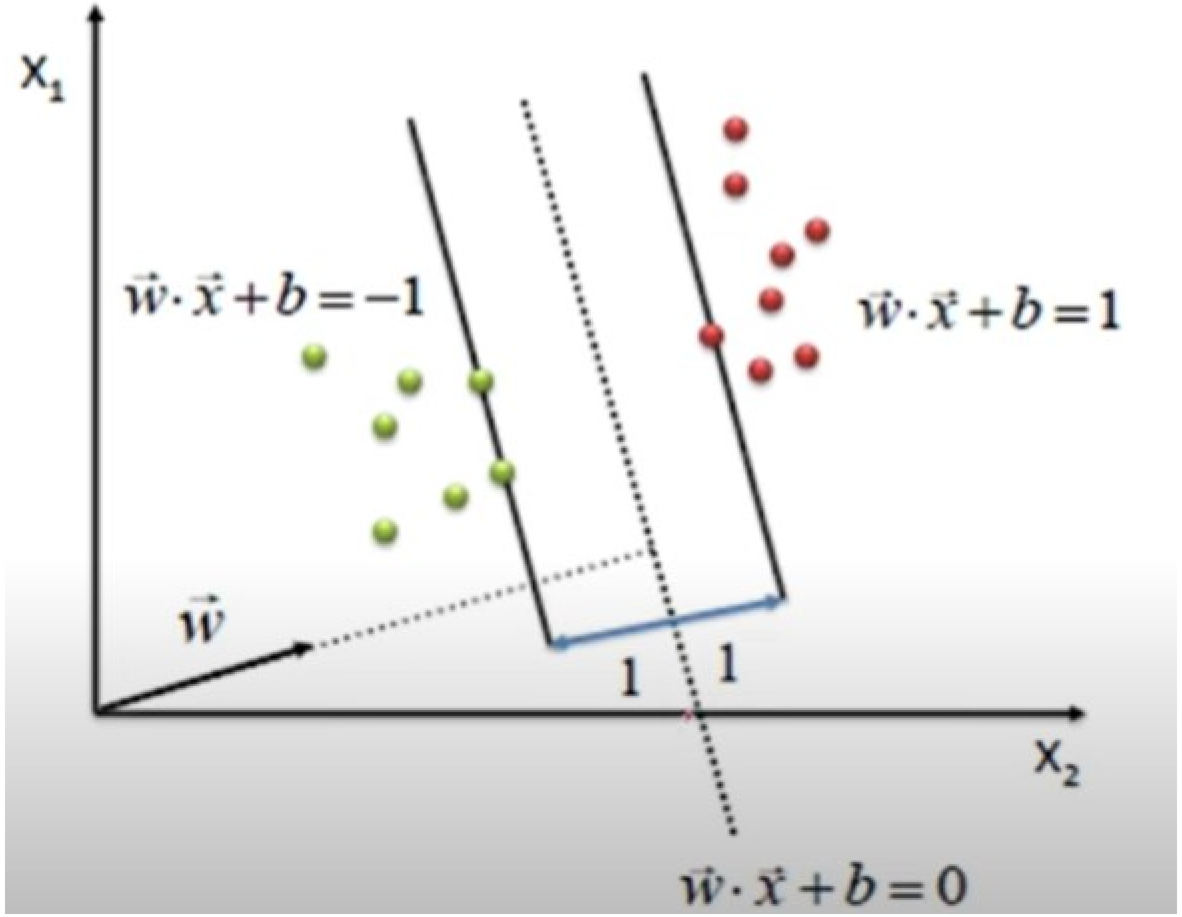
W.T* X + B OR W dot X + B

After training we get optimal values of W & B Then

If for any xi

W.T* X + B>0 then xi lies on +ve side (or class c1)

W.T* X + B<0 then xi lies on -ve side (or class c2)



X_1
$\vec{w}\cdot\vec{x}+b=-1$
$\vec{w}\cdot\vec{x}+b=1$
$\vec{w}$
1
1
X_2
$\vec{w}\cdot\vec{x}+b=0$

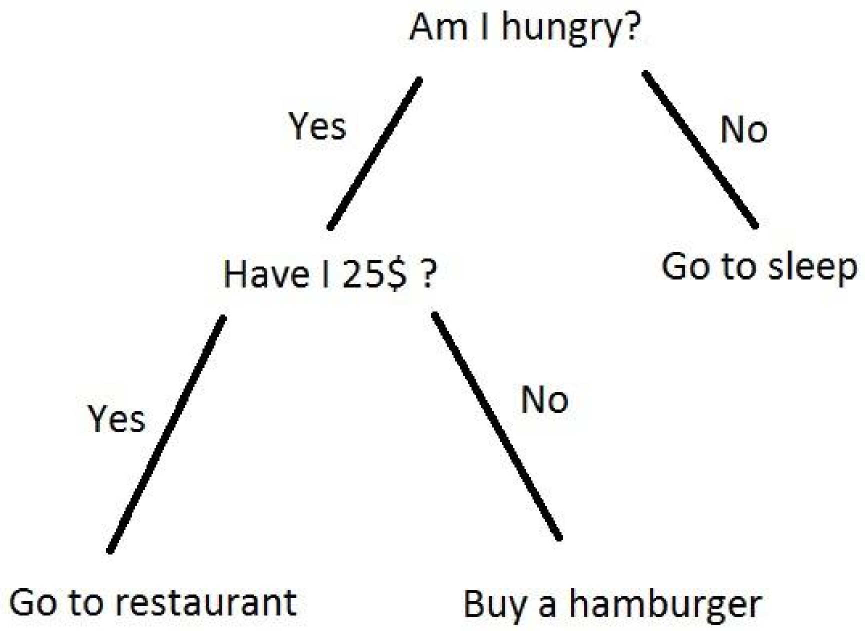
Decision Tree

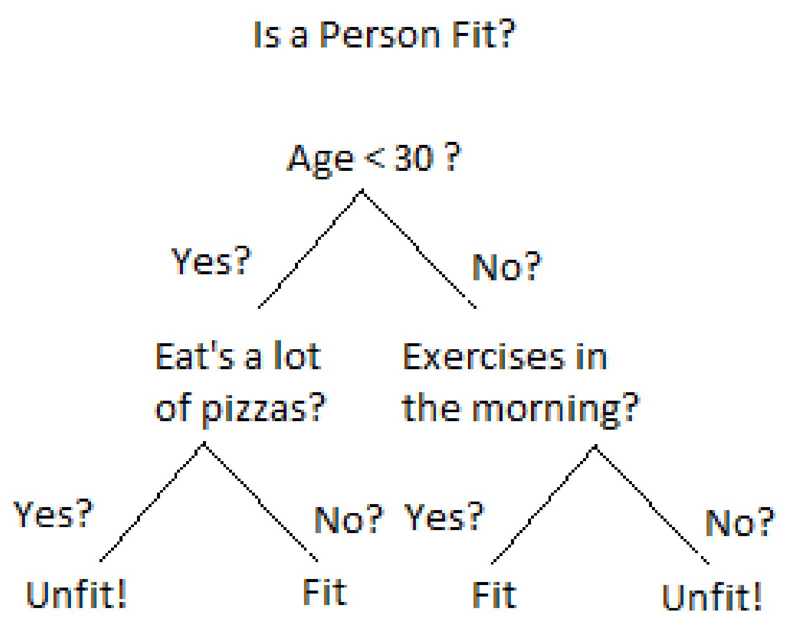
- Decision trees are used for both classification and regression problems
- Decision tree is the most powerful and popular tool for classification and prediction.
- A Decision tree is a flowchart like tree structure, where each internal node denotes a test on an attribute, each branch represents an outcome of the test, and each leaf node (terminal node) holds a class label.
- simple to understand the data and make some good interpretations.

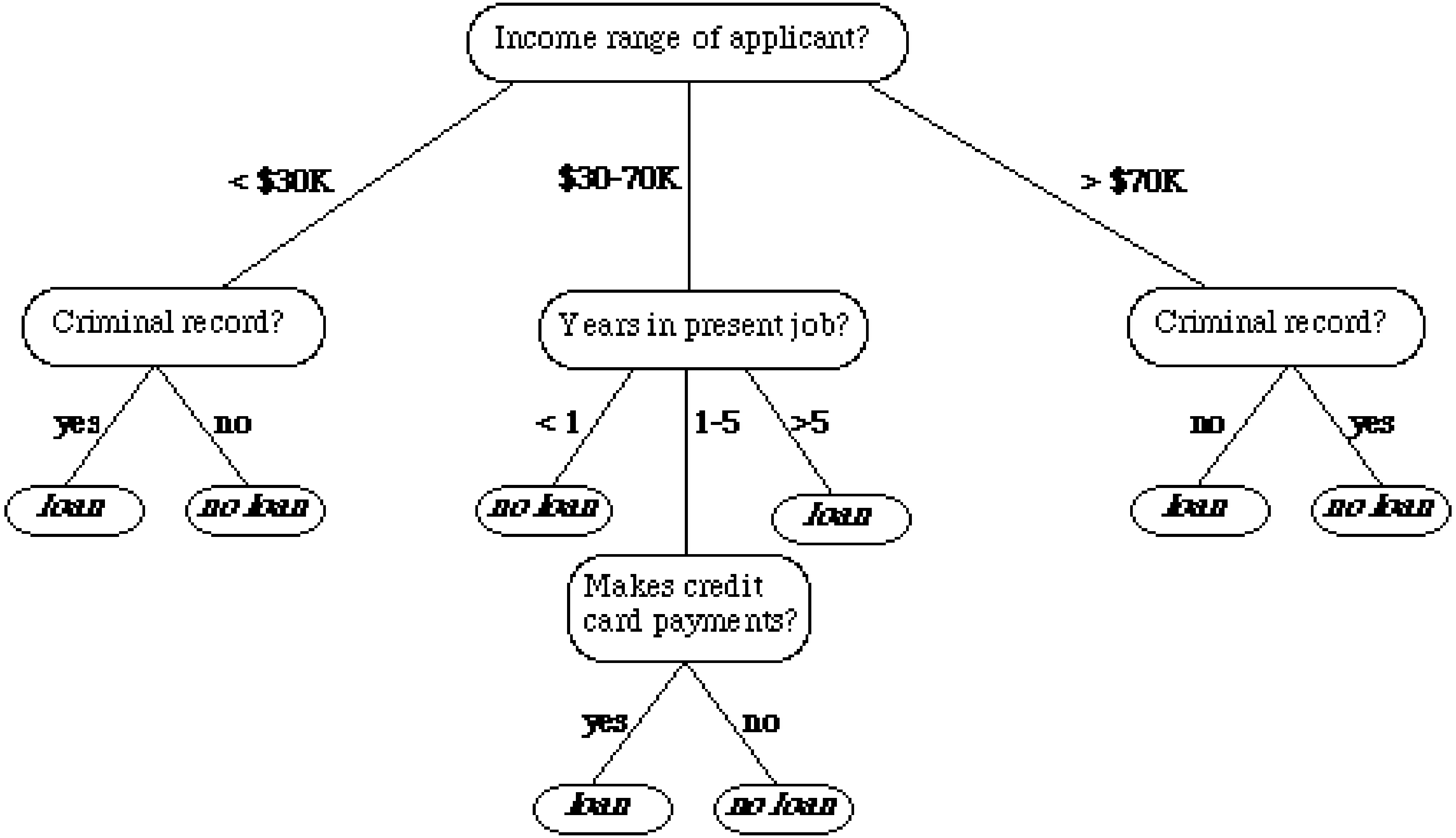
A decision tree is a tree where each node represents a feature(attribute), each link(branch) represents a decision(rule) and each leaf represents an outcome(categorical or continues value).

Tree can be built by using many algorithm but Popular are

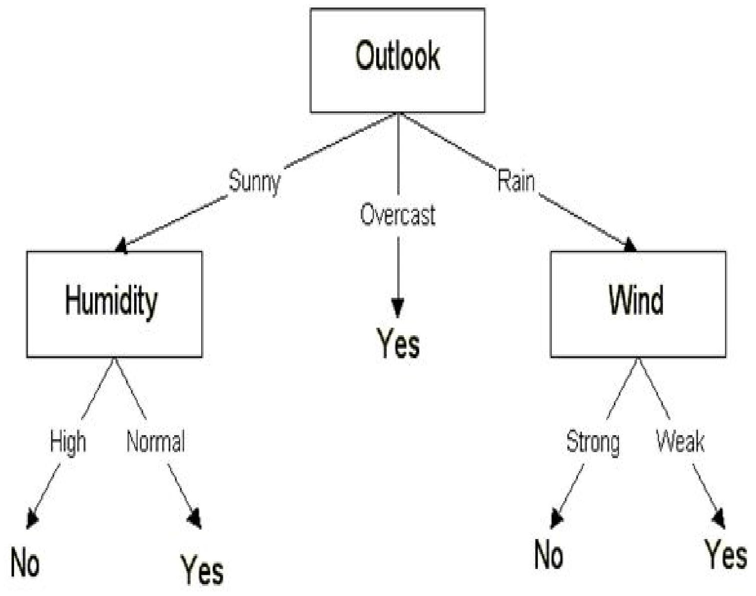
1. ID3 (Iterative Dichotomiser 3) → uses ***Entropy function*** and ***Information gain*** as metrics.
2. CART (Classification and Regression Trees) → uses ***Gini Index(Classification)*** as metric.


Am I hungry?
Yes
No
Have I 25$?
Go to sleep
Yes
No
Go to restaurant
Buy a hamburger


Is a Person Fit?
Age < 30 ?
Yes?
No?
Eat's a lot of pizzas?
Exercises in the morning?
Yes?
No?
Yes?
No?
Unfit!
Fit
Fit
Unfit!


Income range of applicant?
< $30K
$30-70K
> $70K
Criminal record?
yes
no
loan
no loan
Years in present job?
< 1
1-5
>5
no loan
loan
Makes credit card payments?
yes
no
loan
no loan
Criminal record?
no
yes
loan
no loan

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no



Entropy

- ***Entropy*** *is the measures of* ***impurity***, ***disorder*** *or* ***uncertainty*** *in a bunch of examples.*
- *Entropy controls how a Decision Tree decides to* ***split*** *the data. It actually effects how a* ***Decision Tree*** *draws its boundaries.*

$$Entropy = -\sum p(X) \log p(X)$$

here p(x) is a <u>fraction</u> of examples in a given <u>class</u>

Information Gain

- ***Information gain (IG)*** *measures how much "information" a feature gives us about the class.*
- **Information gain** is the main key that is used by **Decision Tree Algorithms** to construct a Decision Tree.
- **Decision Trees** algorithm will always tries to maximize **Information gain**.
- An **attribute** with highest **Information gain** will tested/split first.

$$\text{Information gain} = \text{entropy (parent)} - [\text{weightes average}] * \text{entropy (children)}$$

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

$$H(S) = \sum_{c \in C} -p(c) \log_2 p(c)$$

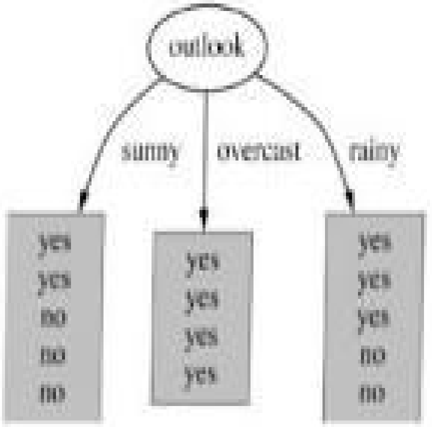
C ={yes,no}

Out of 14 instances, 9 are classified as yes, and 5 as no

pyes = −(9/14)*log2(9/14) = 0.41

pno = −(5/14)*log2(5/14) = 0.53

H (S) = pyes + pno = 0.94



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

E (Outlook=overcast) = $-1\log(1) - 0\log(0) = 0$

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

H(S,Outlook)

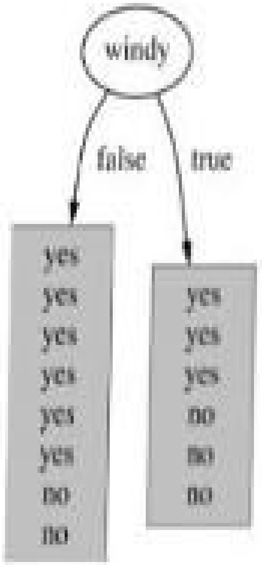
Average Entropy information for Outlook

I (Outlook) = $\frac{5}{14} * 0.971 + \frac{4}{14} * 0 + \frac{5}{14} * 0.971 = 0.693$

$$\sum_{t \in T} p(t)H(t)$$

Gain (Outlook) = E(S) – I (outlook) = 0.94-.693 = 0.247

$$IG(A,S) = H(S) - \sum_{t \in T} p(t)H(t)$$



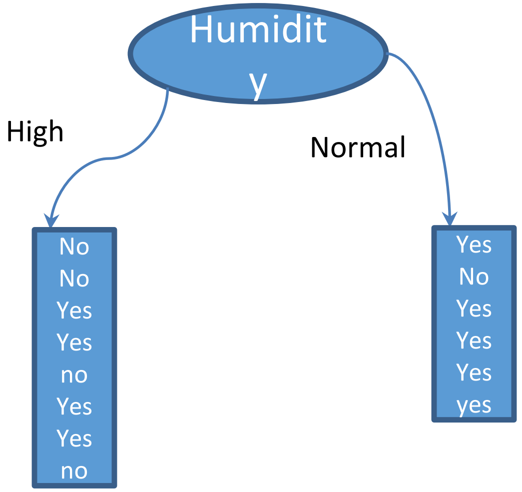
E (Windy=false) = $-\frac{6}{8}\log\left(\frac{6}{8}\right)-\frac{2}{8}\log\left(\frac{2}{8}\right) = 0.811$

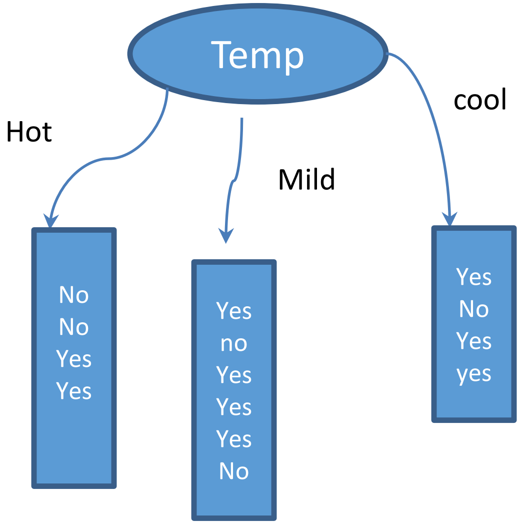
E (Windy=true) = $-\frac{3}{6}\log\left(\frac{3}{6}\right)-\frac{3}{6}\log\left(\frac{3}{6}\right) = 1$

Average entropy information for Windy

I (Windy) = $\frac{8}{14} * 0.811 + \frac{6}{14} * 1 = 0.892$

Gain (Windy) = E(S) – *I* (Windy) = 0.94-0.892=0.048





E(Humidity,high)=-4/8*log(4/8)-4/8log(4/8)=x

E(humidity,normal)=-5/6*log5/6-1/6*log(1/6)=y

Average entropy of humidity (z)

Z=8/14*x+6/14*y

Gain(humidity)=.94-z=p

E(temp, High)=-2/4*log(2/4)-2/4*log(2/4)=s

E(temp, mild)=-4/6*log(4/6)-2/6*log(2/6)=t

E(temp, cool)=-3/4*log(3/4)-1/4*log(1/4)=u

Average entropy of Temp

Q=4/14*s+6/14*t+4/14*u

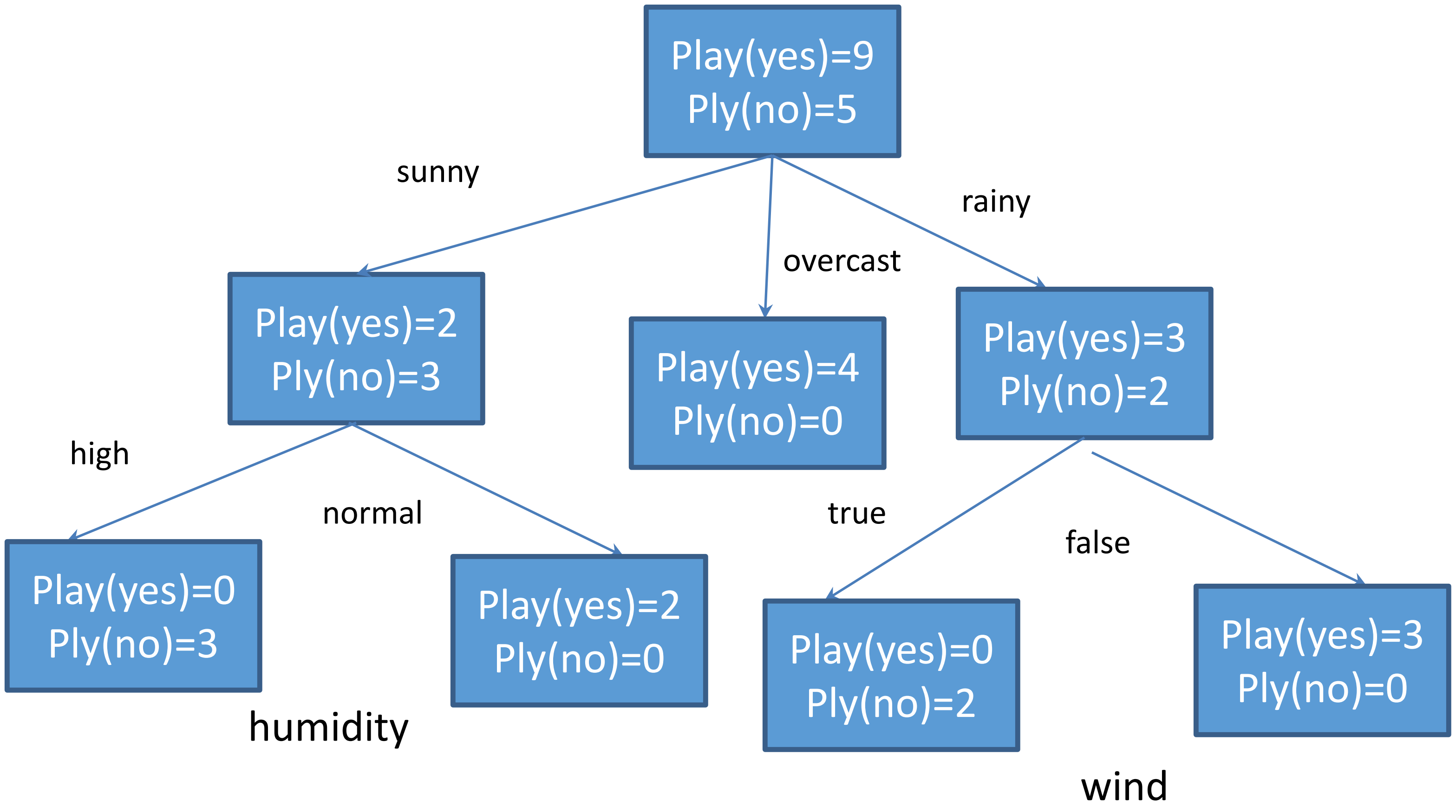
Gain(temp)=.94-Q

$IG(S, Outlook) = 0.246$

$IG(S, Temperature) = 0.029$

$IG(S, Humidity) = 0.151$

$IG(S, Wind) = 0.048$ (Previous example)



Play(yes)=9
Ply(no)=5
sunny
overcast
rainy
Play(yes)=2
Ply(no)=3
Play(yes)=4
Ply(no)=0
Play(yes)=3
Ply(no)=2
high
normal
true
false
Play(yes)=0
Ply(no)=3
Play(yes)=2
Ply(no)=0
Play(yes)=0
Ply(no)=2
Play(yes)=3
Ply(no)=0
humidity
wind

CART

- Classification and Regression Tree(CART)
- Gini index is a metric for classification tasks in CART.
- It stores sum of squared probabilities of each class.

  Gini = $1 - \Sigma (Pi)^2$ for i=1 to number of classes

outlook	temp.	humidity	windy	play
sunny	hot	high	false	no
sunny	hot	high	true	no
overcast	hot	high	false	yes
rainy	mild	high	false	yes
rainy	cool	normal	false	yes
rainy	cool	normal	true	no
overcast	cool	normal	true	yes
sunny	mild	high	false	no
sunny	cool	normal	false	yes
rainy	mild	normal	false	yes
sunny	mild	normal	true	yes
overcast	mild	high	true	yes
overcast	hot	normal	false	yes
rainy	mild	high	true	no

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$

Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$

Gini(Outlook=Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$

Then, we will calculate weighted sum of gini indexes for outlook feature.

Gini(Outlook) = $(5/14) \times 0.48 + (4/14) \times 0 + (5/14) \times 0.48 = 0.171 + 0 + 0.171 = 0.342$

Outlook	Yes	No	Number of instances
Sunny	2	3	5
Overcast	4	0	4
Rain	3	2	5

Gini(Outlook=Sunny) = $1 – (2/5)^2 – (3/5)^2 = 1 – 0.16 – 0.36 = 0.48$
Gini(Outlook=Overcast) = $1 – (4/4)^2 – (0/4)^2 = 0$
Gini(Outlook=Rain) = $1 – (3/5)^2 – (2/5)^2 = 1 – 0.36 – 0.16 = 0.48$
Then, we will calculate weighted sum of gini indexes for outlook feature.
Gini(Outlook) = (5/14) x 0.48 + (4/14) x 0 + (5/14) x 0.48 = 0.171 + 0 + 0.171 = 0.342

Temperature	Yes	No	Number of instances
Hot	2	2	4
Cool	3	1	4
Mild	4	2	6

Temperature

Gini(Temp=Hot) = $1 – (2/4)^2 – (2/4)^2 = 0.5$
Gini(Temp=Cool) = $1 – (3/4)^2 – (1/4)^2 = 1 – 0.5625 – 0.0625 = 0.375$
Gini(Temp=Mild) = $1 – (4/6)^2 – (2/6)^2 = 1 – 0.444 – 0.111 = 0.445$
We'll calculate weighted sum of gini index for temperature feature
Gini(Temp) = (4/14) x 0.5 + (4/14) x 0.375 + (6/14) x 0.445 = 0.142 + 0.107 + 0.190 = 0.439

Humidity	Yes	No	Number of instances
High	3	4	7
Normal	6	1	7

Humidity

Humidity is a binary class feature. It can be high or normal.

Gini(Humidity=High) = $1 – (3/7)^2 – (4/7)^2$
$= 1 – 0.183 – 0.326$
$= 0.489$

Gini(Humidity=Normal) = $1 – (6/7)^2 – (1/7)^2$
$= 1 – 0.734 – 0.02$
$= 0.244$

Weighted sum for humidity feature will be calculated next

Gini(Humidity) = (7/14) x 0.489 + (7/14) x 0.244 = 0.367

Wind	Yes	No	Number of instances
Weak	6	2	8
Strong	3	3	6

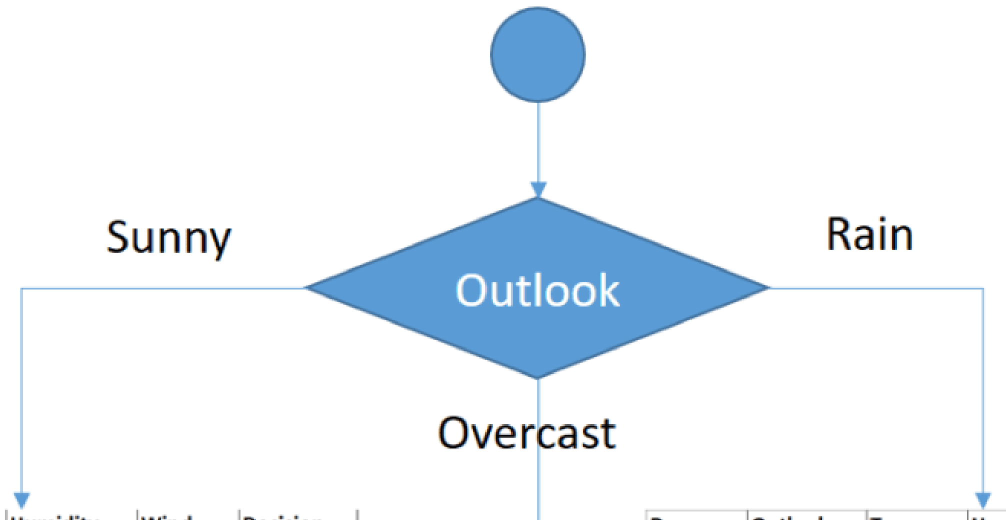
Wind

Gini(Wind=Weak) = $1 – (6/8)^2 – (2/8)^2$
$= 1 – 0.5625 – 0.062 = 0.375$

Gini(Wind=Strong) = $1 – (3/6)^2 – (3/6)^2$
$= 1 – 0.25 – 0.25 = 0.5$

Gini(Wind) = (8/14) x 0.375 + (6/14) x 0.5 = 0.428

Feature	Gini index
Outlook	0.342
Temperature	0.439
Humidity	0.367
Wind	0.428



Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
10	Rain	Mild	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

Day	Outlook	Temp.	Humidity	Wind	Decision
3	Overcast	Hot	High	Weak	Yes
7	Overcast	Cool	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes

Temperature	Yes	No	Number of instances
Hot	0	2	2
Cool	1	0	1
Mild	1	1	2

Gini of temperature for sunny outlook

Gini(Outlook=Sunny and Temp.=Hot) = 1 – $(0/2)^2$ – $(2/2)^2$ = 0
Gini(Outlook=Sunny and Temp.=Cool) = 1 – $(1/1)^2$ – $(0/1)^2$ = 0
Gini(Outlook=Sunny and Temp.=Mild) = 1 – $(1/2)^2$ – $(1/2)^2$ = 1 – 0.25 – 0.25 = 0.5
Gini(Outlook=Sunny and Temp.) = (2/5)x0 + (1/5)x0 + (2/5)x0.5 = 0.2

Wind	Yes	No	Number of instances
Weak	1	2	3
Strong	1	1	2

Gini of wind for sunny outlook

Gini(Outlook=Sunny and Wind=Weak) = $1 – (1/3)^2 – (2/3)^2 = 0.266$

Gini(Outlook=Sunny and Wind=Strong) = $1- (1/2)^2 – (1/2)^2 = 0.2$

Gini(Outlook=Sunny and Wind) = $(3/5)x0.266 + (2/5)x0.2 = 0.466$

Humidity	Yes	No	Number of instances
High	0	3	3
Normal	2	0	2

Gini of humidity for sunny outlook

Gini(Outlook=Sunny and Humidity=High) = $1 – (0/3)^2 – (3/3)^2 = 0$

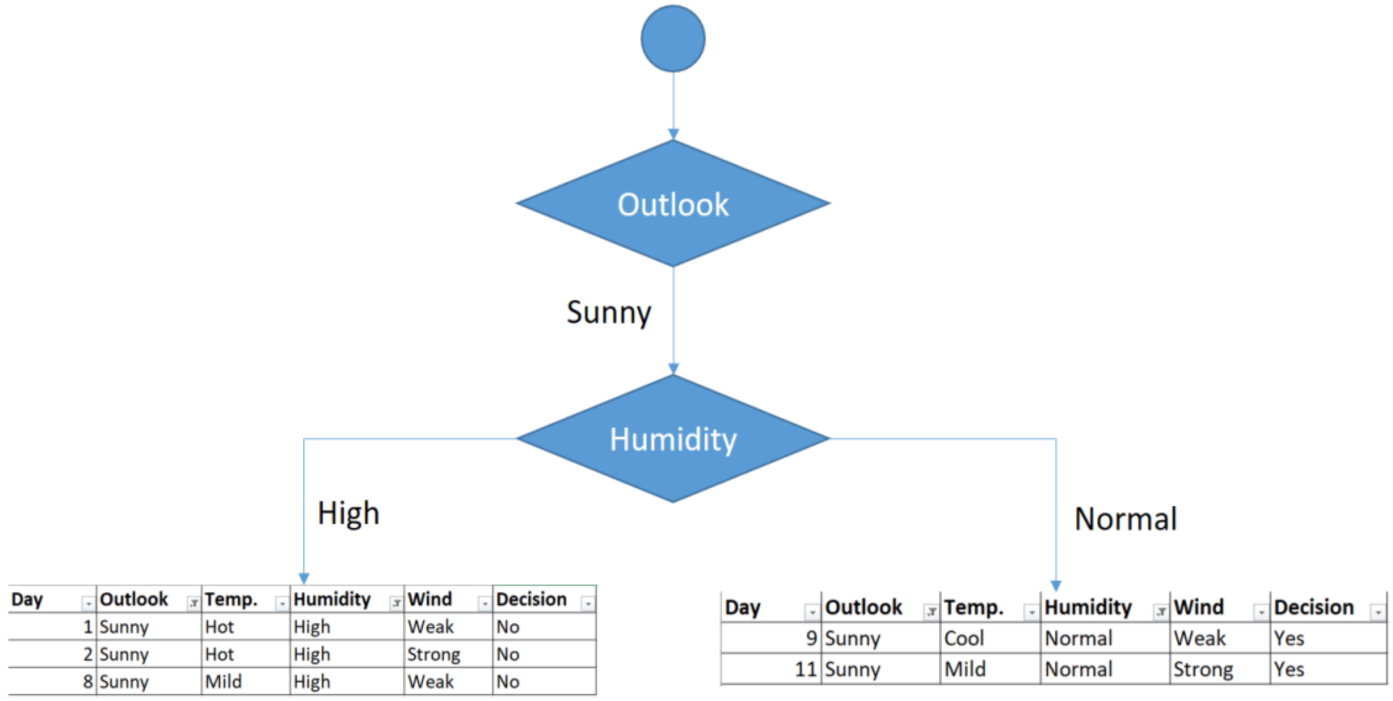
Gini(Outlook=Sunny and Humidity=Normal) = $1 – (2/2)^2 – (0/2)^2 = 0$

Gini(Outlook=Sunny and Humidity) = $(3/5)x0 + (2/5)x0 = 0$

Decision for sunny outlook

We’ve calculated gini index scores for feature when outlook is sunny. The winner is humidity because it has the lowest value. We’ll put humidity check at the extension of sunny outlook.

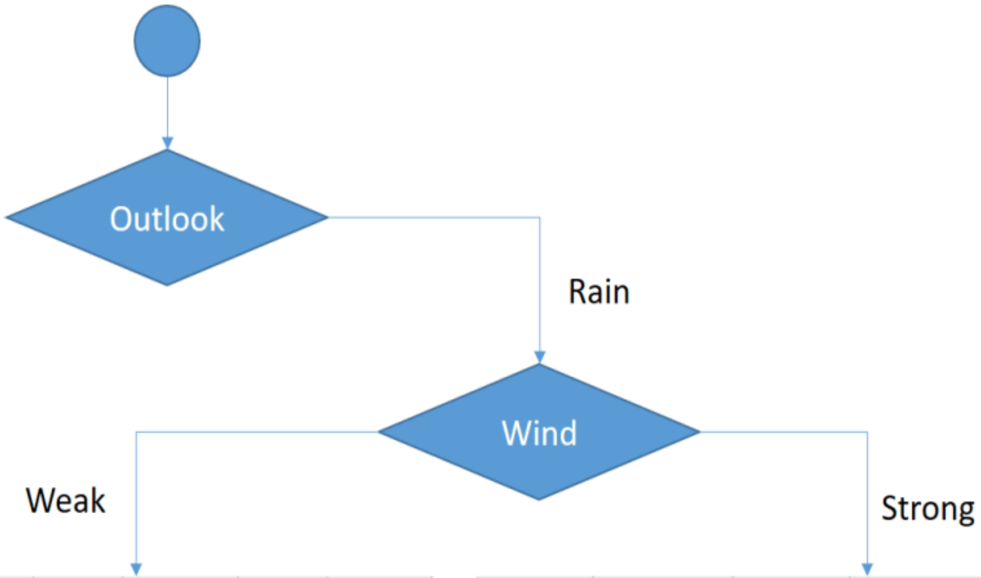
Feature	Gini index
Temperature	0.2
Humidity	0
Wind	0.466



Decision for rain outlook

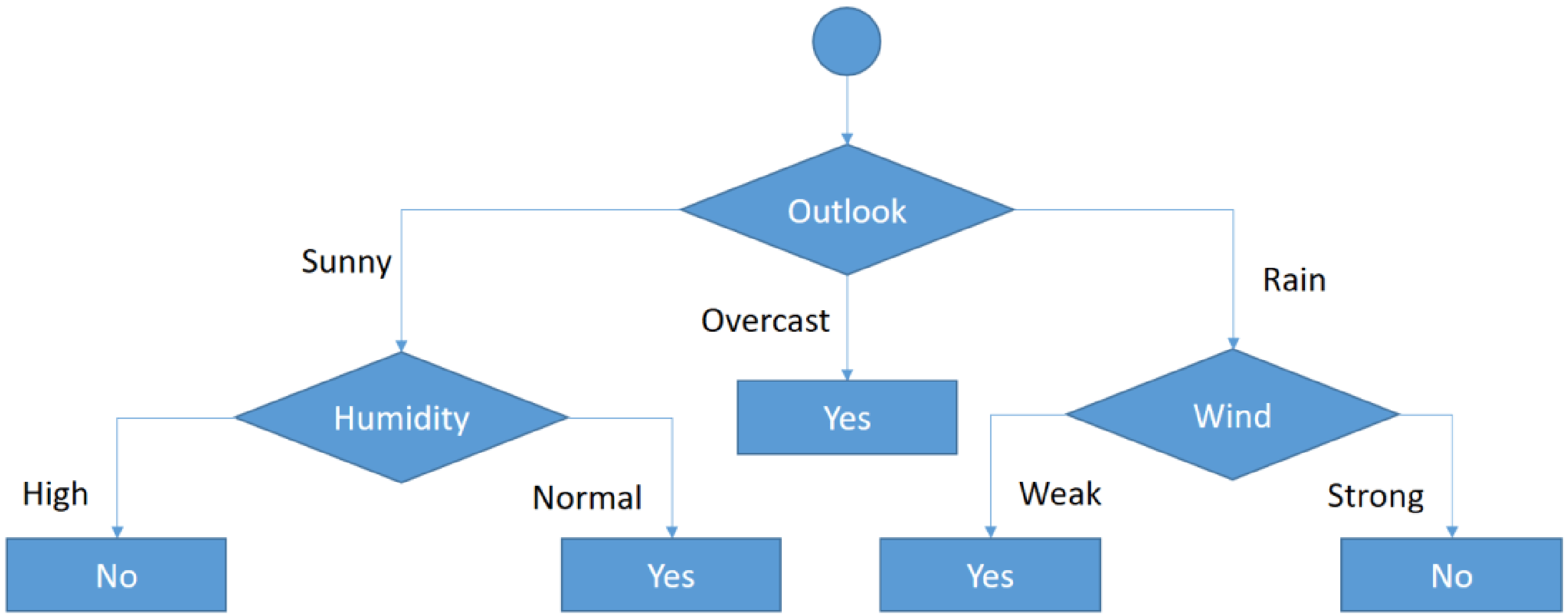
The winner is wind feature for rain outlook because it has the minimum gini index score in features.
Put the wind feature for rain outlook branch and monitor the new sub data sets.

Temperature	0.466
Humidity	0.466
Wind	0



Day	Outlook	Temp.	Humidity	Wind	Decision
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes

Day	Outlook	Temp.	Humidity	Wind	Decision
6	Rain	Cool	Normal	Strong	No
14	Rain	Mild	High	Strong	No


Outlook
Sunny
Overcast
Rain
Humidity
Yes
Wind
High
Normal
Weak
Strong
No
Yes
Yes
No