DECISION TREE

DECISION TREE

- ClAssification and Regression Tree (CART)

Recommendation System - 1

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Woman, works at an office.
What app do we recommend?

- ○  Pokémon Go
- ○  WhatsApp
- ○ 

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Woman, works at an office.
What app do we recommend?

- ○ Pokémon Go
- ● WhatsApp
- ○ Snapchat

Recommendation System - 2

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Man, works at a factory.
What app do we recommend?

- ○ Pokémon Go
- ○ WhatsApp
- ○ Snapchat

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Man, works at a factory. What app do we recommend?

- Pokémon Go
- WhatsApp
- Snapchat

Recommendation System - 3

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Girl, goes to high school.
What app do we recommend?

- ○ Pokémon Go
- ○ WhatsApp
- ○ Snapchat

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Girl, goes to high school.
What app do we recommend?

- Pokémon Go
- WhatsApp
- Snapchat

Way Machine approaches

Recommending Apps

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Quiz: Between **Gender** and **Occupation**, which one seems more decisive for predicting what app will the users download?

- ○ **Gender**
- ○ **Occupation**

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

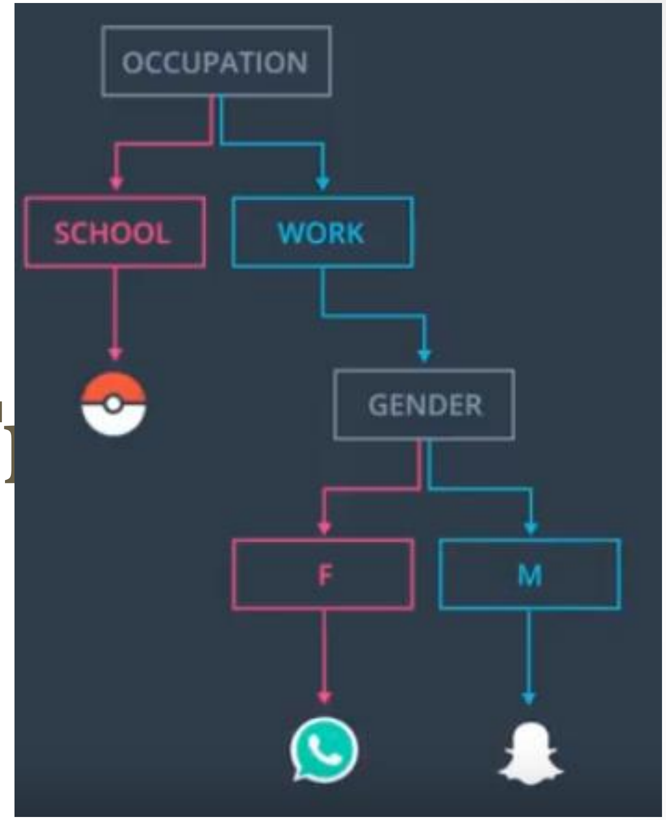
Quiz: Between **Gender** and **Occupation**, which one seems more decisive for predicting what app will the users download?

○ **Gender**

● **Occupation**

Gender	Occupation	App
F	Study	
F	Work	
M	Work	
F	Work	
M	Study	
M	Study	

Tr



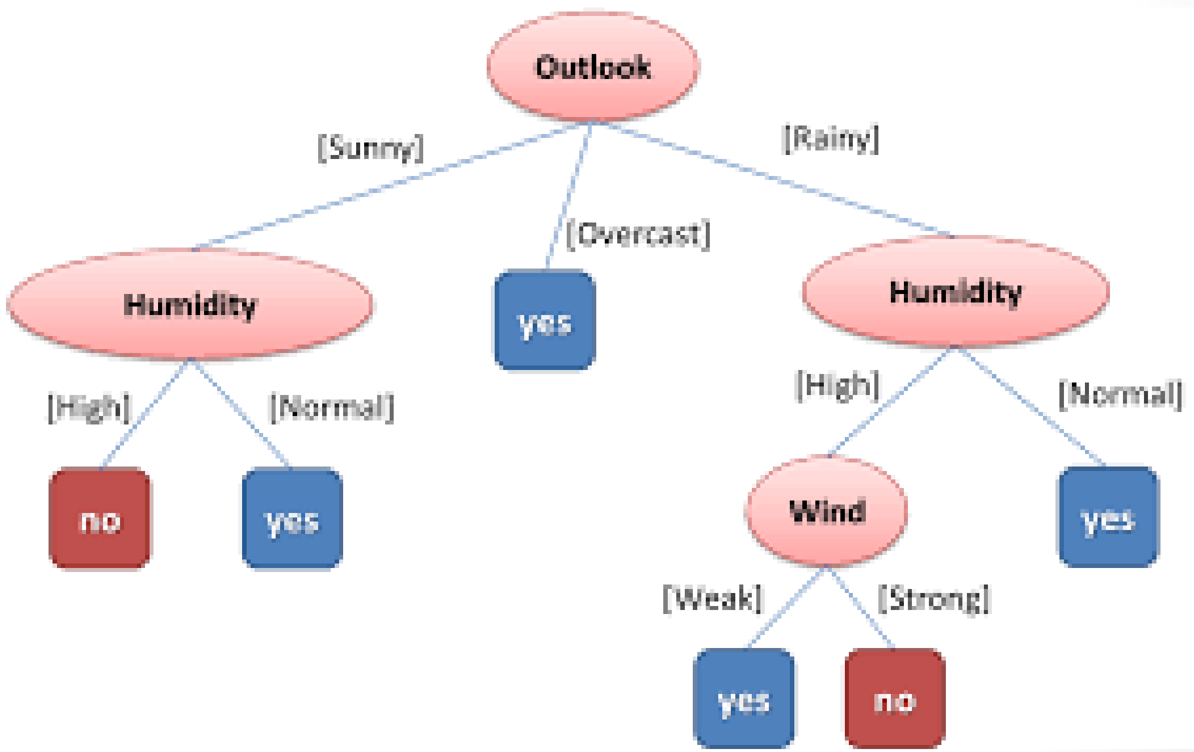
Supervised learning algorithm

Root Node

Decision node

Leaves

Structure of a Tree



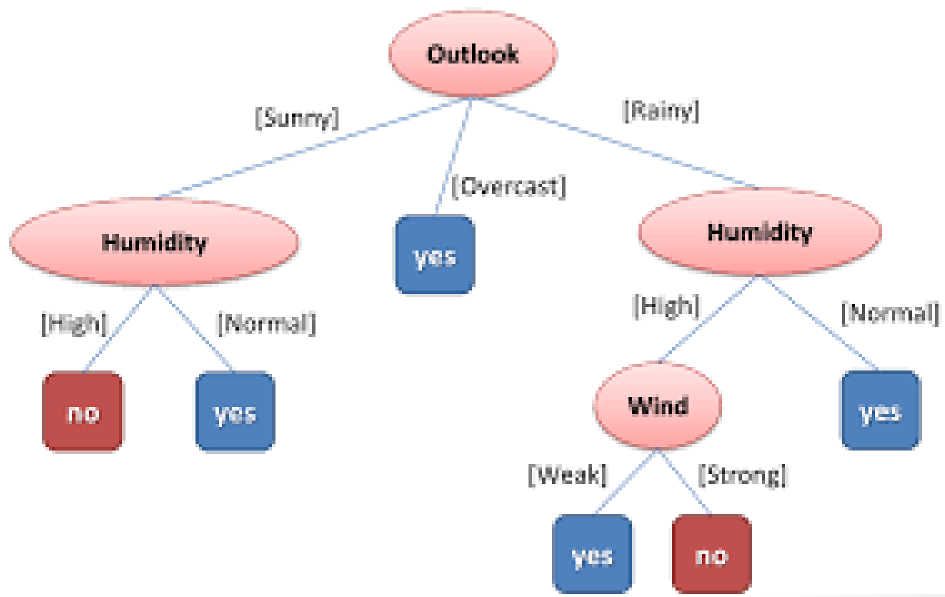
Supervised learning algorithm

Root Node - Outlook

Decision node - Humidity/Wind

Leaves - Yes/No

Structure of a Tree



HOW DECISION TREE ALGORITHM WORKS

HOW TO FIND ROOT (2 WAYS)

- **Information gain**
- **Gini index**

Information Gain & Entropy

Information Gain -> Information theory -> Entropy

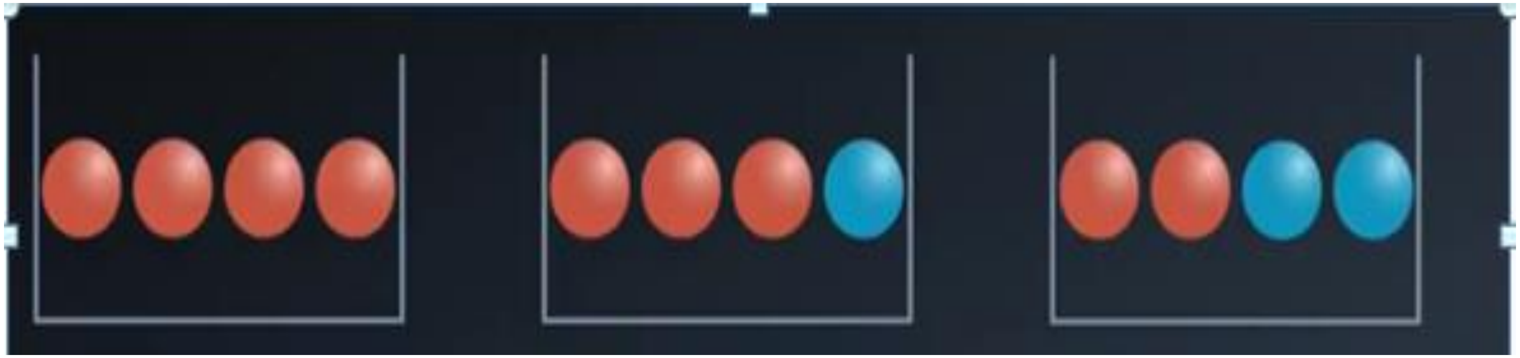
Entropy = **Randomness** or **Uncertainty** of a random variable.

There are **2 steps for calculating information gain** for each attribute:

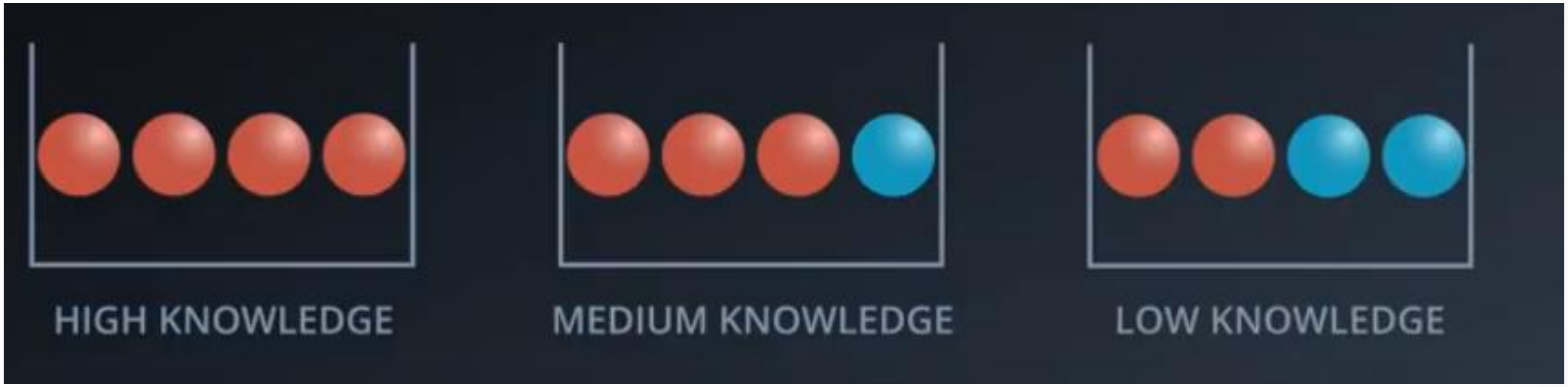
- Calculate entropy of Target.
- Calculate the Entropy for every attribute.

Information gain = Entropy of target - Entropy of attribute

Entropy - The measure of uncertainty

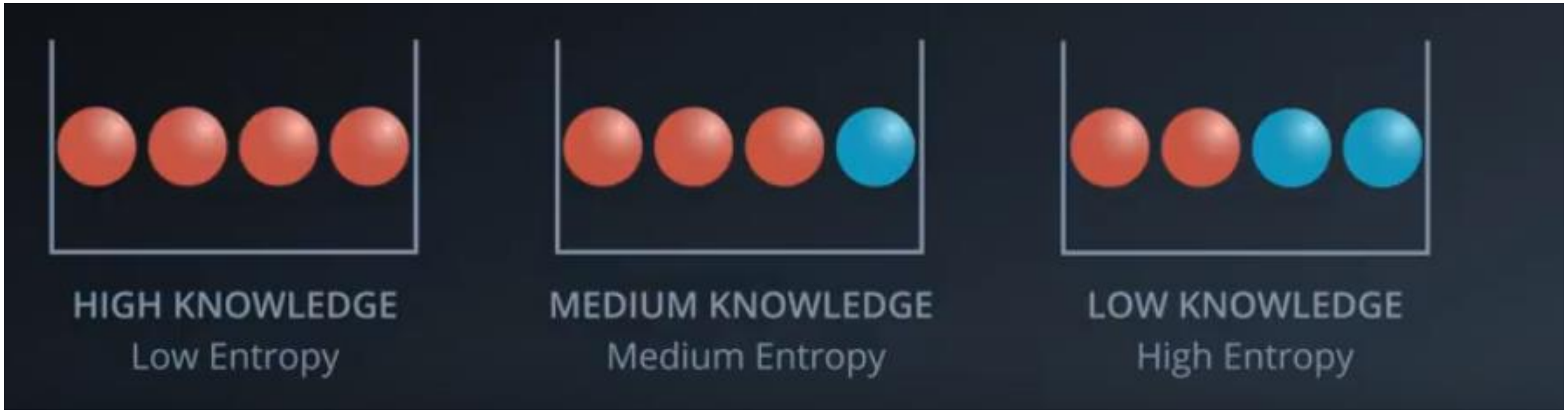


Entropy - The measure of uncertainty



Entropy - The measure of uncertainty

Entropy - The measure of uncertainty



$$H(X) = \mathbb{E}_X[I(x)] = -\sum_{x \in \mathbb{X}} p(x) \log p(x).$$

Case Study – Golf Play Dataset

Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Entropy of Target

Play Golf
No
No
Yes
Yes
Yes
No
Yes
No
Yes
Yes
Yes
Yes
Yes
No

Sort

Play Golf
No
No
No
No
No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes

5 / 14 = 0.36

9 / 14 = 0.64

Entropy(PlayGolf) = Entropy (5,9)

$= \text{Entropy}\ (0.36, 0.64)$

$= -\ (0.36 \log_2 0.36) - (0.64 \log_2 0.64)$

$= 0.94$

Frequency Table – 4 Attributes

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3

Entropy - Outlook

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)***E**(3,2) + **P**(Overcast)***E**(4,0) + **P**(Rainy)***E**(2,3)

= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

Information Gain - Outlook

$$\mathbf{G}(\text{PlayGolf, Outlook}) = \mathbf{E}(\text{PlayGolf}) - \mathbf{E}(\text{PlayGolf, Outlook})$$

$$= 0.940 - 0.693 = 0.247$$

Information Gain - All Attributes

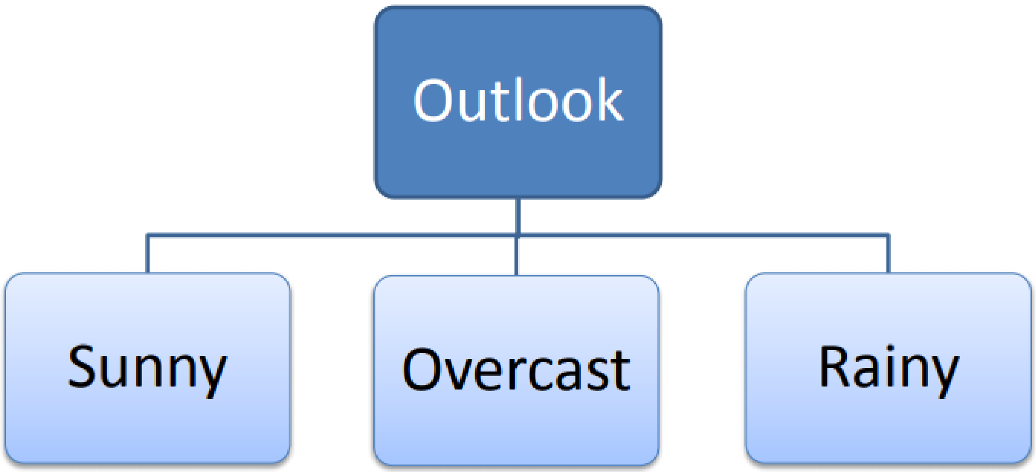
		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

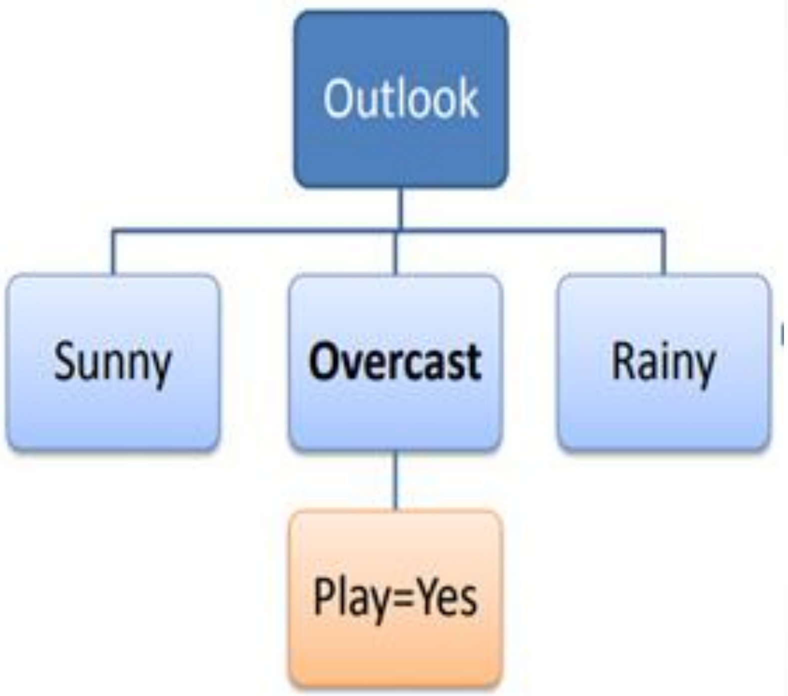
Construction of Tree



Outlook	Temp.	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes

Overcast

Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Sunny

Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	Normal	FALSE	Yes
Mild	High	TRUE	No

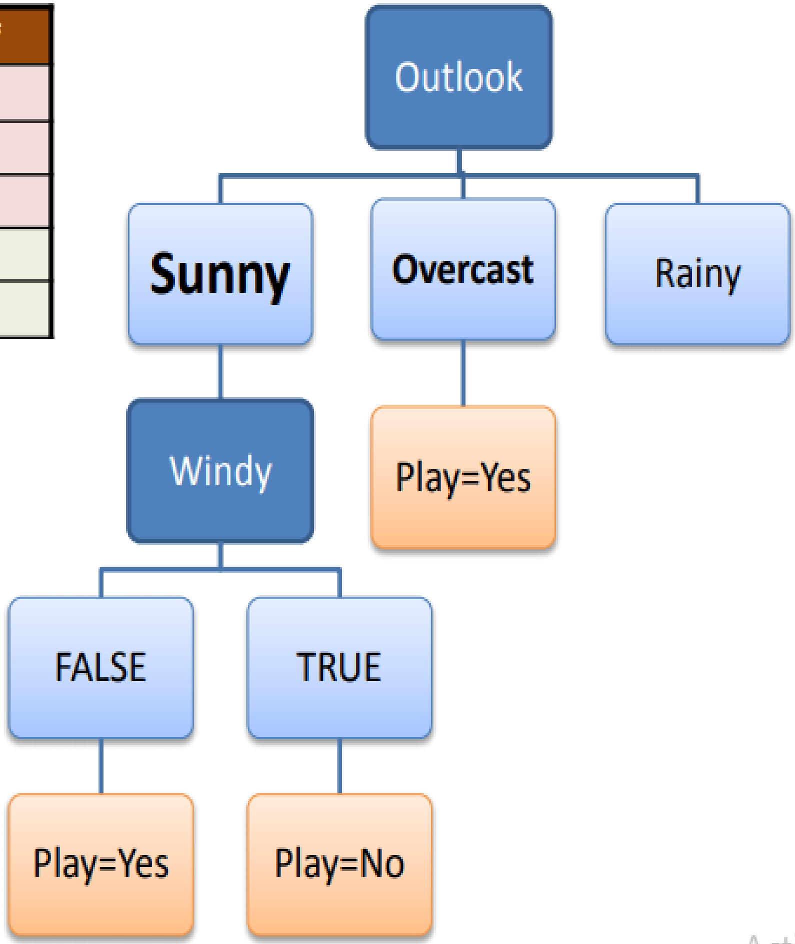
		Play Golf	
		Yes	No
Temp.	Mild	2	1
	Cool	1	1
Gain = 0.02			

		Play Golf	
		Yes	No
Humidity	High	1	1
	Normal	2	1
Gain = 0.02			

		Play Golf	
		Yes	No
Windy	False	3	0
	True	0	2
Gain = 0.97			

Construction of Tree

Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Rainy

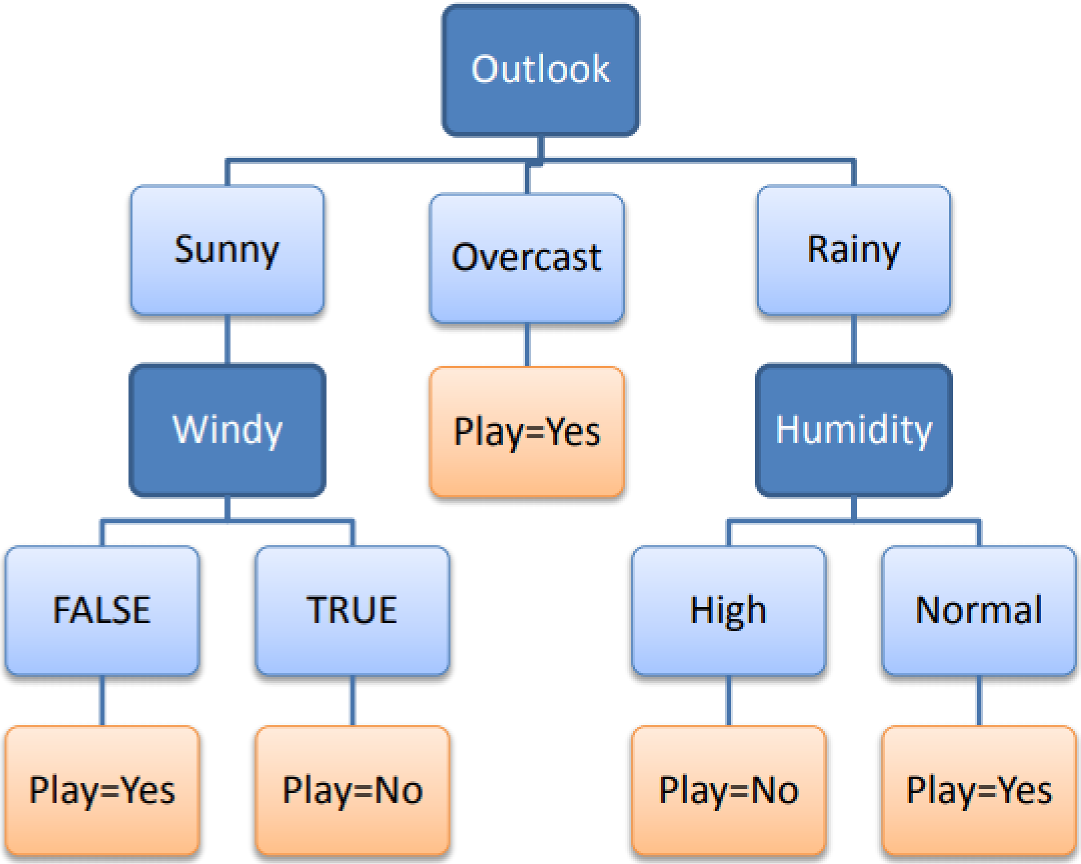
Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	No
Hot	High	TRUE	No
Mild	High	FALSE	No
Cool	Normal	FALSE	Yes
Mild	Normal	TRUE	Yes

		Play Golf	
		Yes	No
Temp.	Hot	0	2
	Mild	1	1
	Cool	1	0
Gain = 0.57			

		Play Golf	
		Yes	No
Humidity	High	0	3
	Normal	2	0
Gain = 0.97			

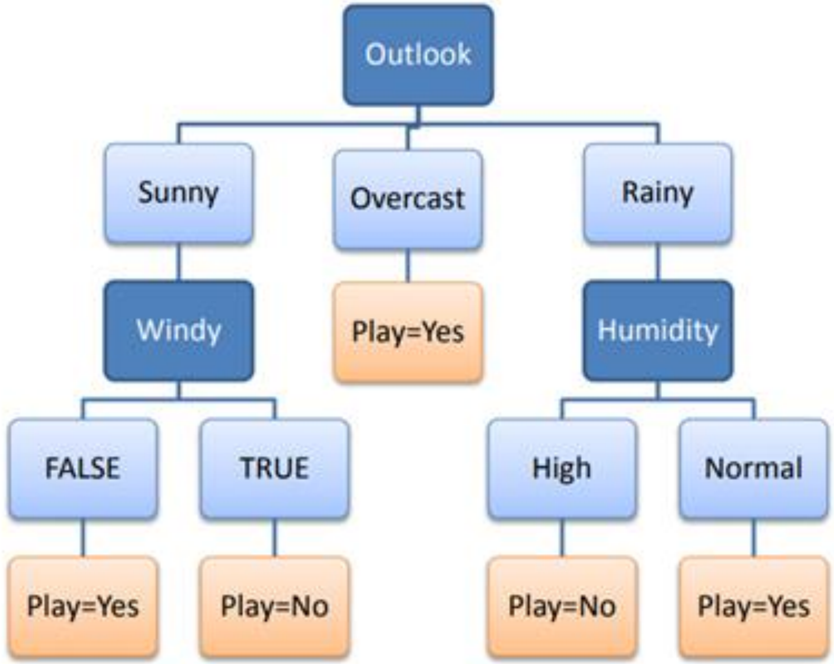
		Play Golf	
		Yes	No
Windy	False	1	2
	True	1	1
Gain = 0.02			

Final Tree Structure



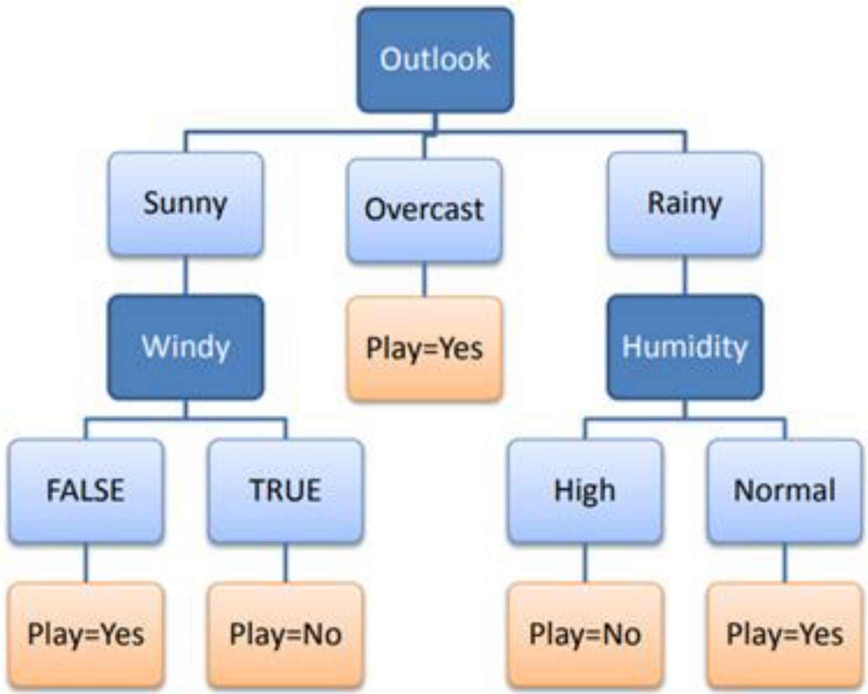
Predict the Play – D15 ?

Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Cool	Normal	FALSE	?



Predict the Play – D15 ?

Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Cool	Normal	FALSE	Yes



Decision Rules – Traditional approach

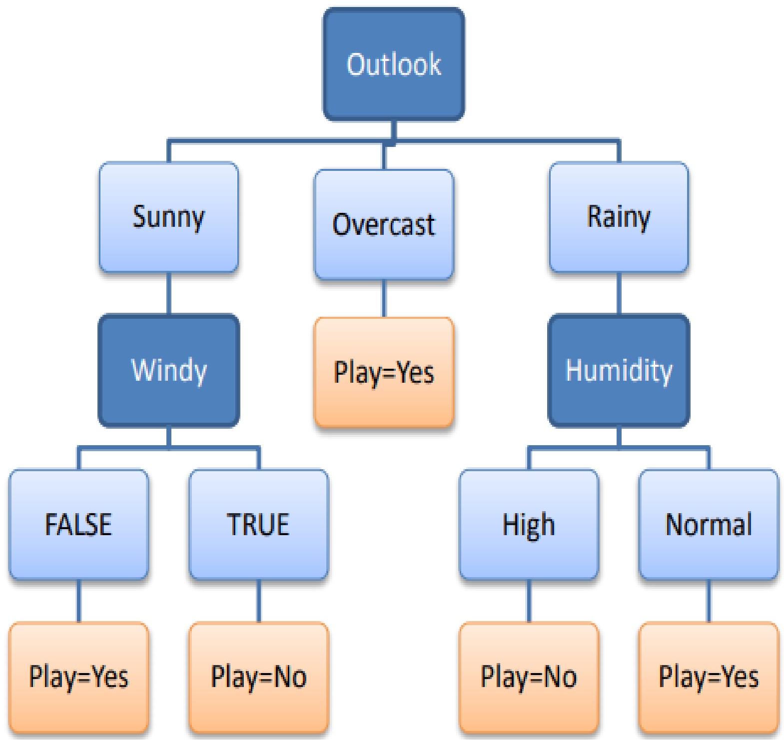
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

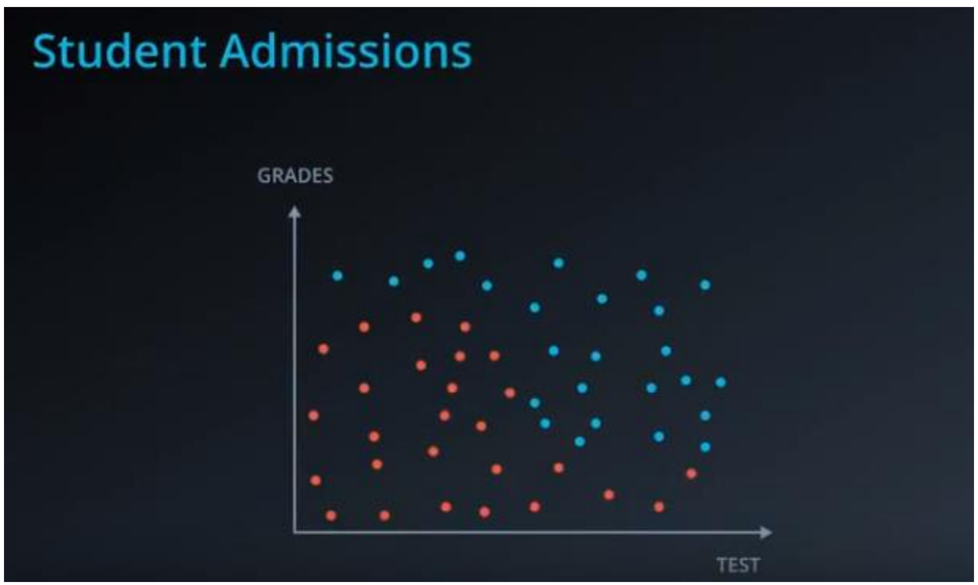


Finding Root using Gini Index

$$Gini\ Index = 1 - \sum_j p_j^2$$

1. The steps to build the tree using **Gini Index** approach is same as the Entropy with the only change in the Formula.
2. In Gini the attribute with the lowest Gini score is used as the ROOT
3. Gini Index is the default method of building the Decision Tree

Continuous Data



Quiz: Between grades and test, which one determines student acceptance better?

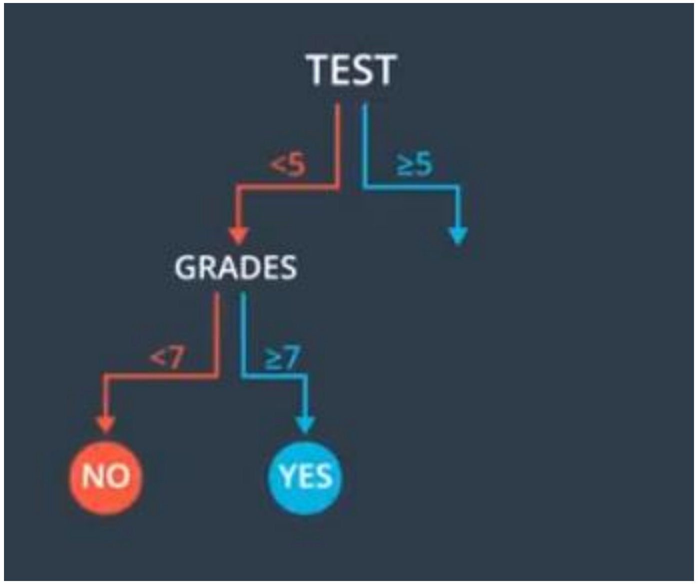
Or

Quiz: Between a horizontal and a vertical line, which one would cut the data better?

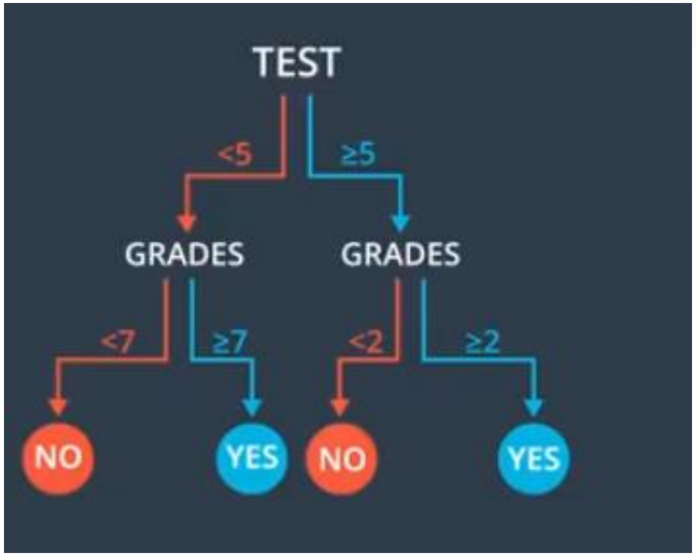
- ○ **Horizontal**
- ○ **Vertical**

Horizontal vs Vertical

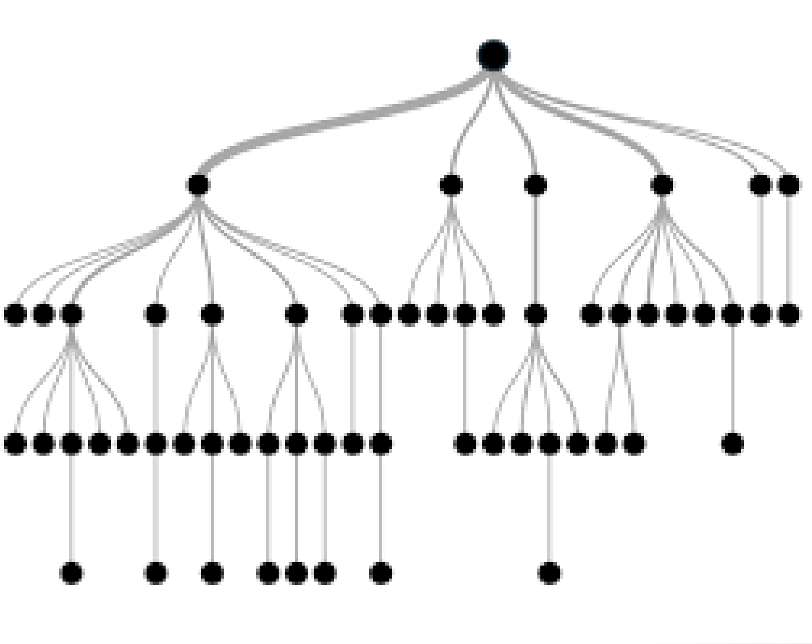
Construction of a Tree



Decision Tree – Manual Structure



When to stop splitting ?
Overfitting



How to overcome Overfitting?

Pruning

1. **Pre-pruning**
2. **Post-pruning**

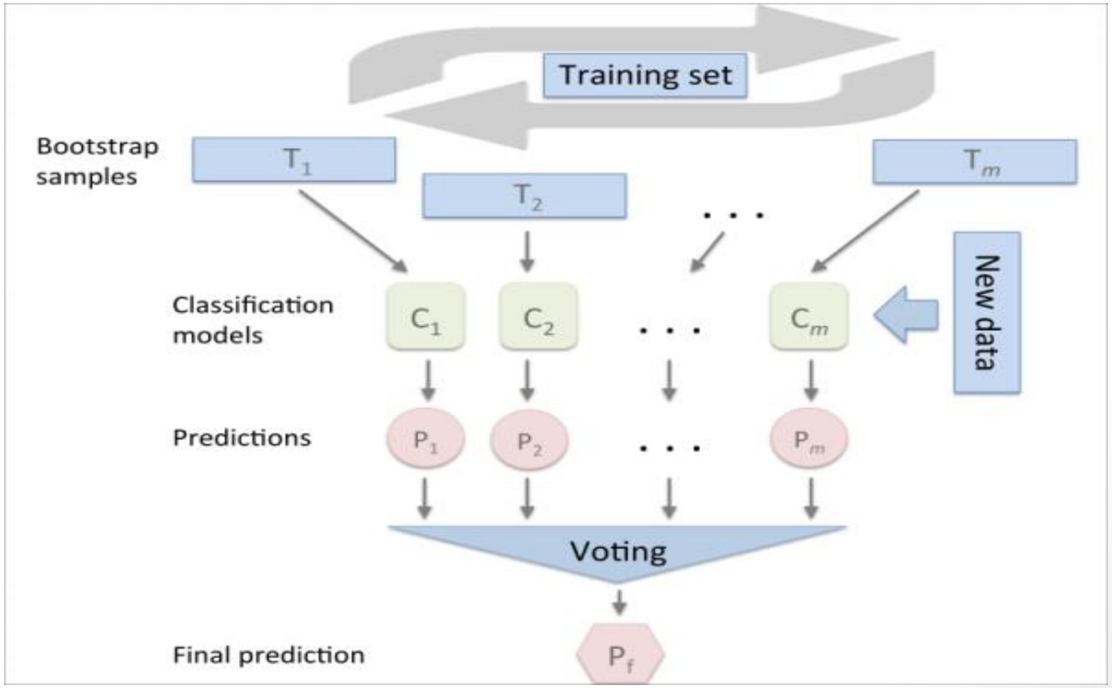
Ensemble

1. Bagging
2. Boosting

Ensemble

Machine learning paradigm which combine weak learners to become a strong learner

Model1	Model2	Model3	VotingPrediction
1	0	1	1



Random Forest *(Most used algorithm)*

- Bagging Technique (**B**ootstrap **agg**regat**ing** - **Bagging**)

Why Random Forest?

No overfitting

Use of multiple trees reduce the risk of overfitting

Training time is less

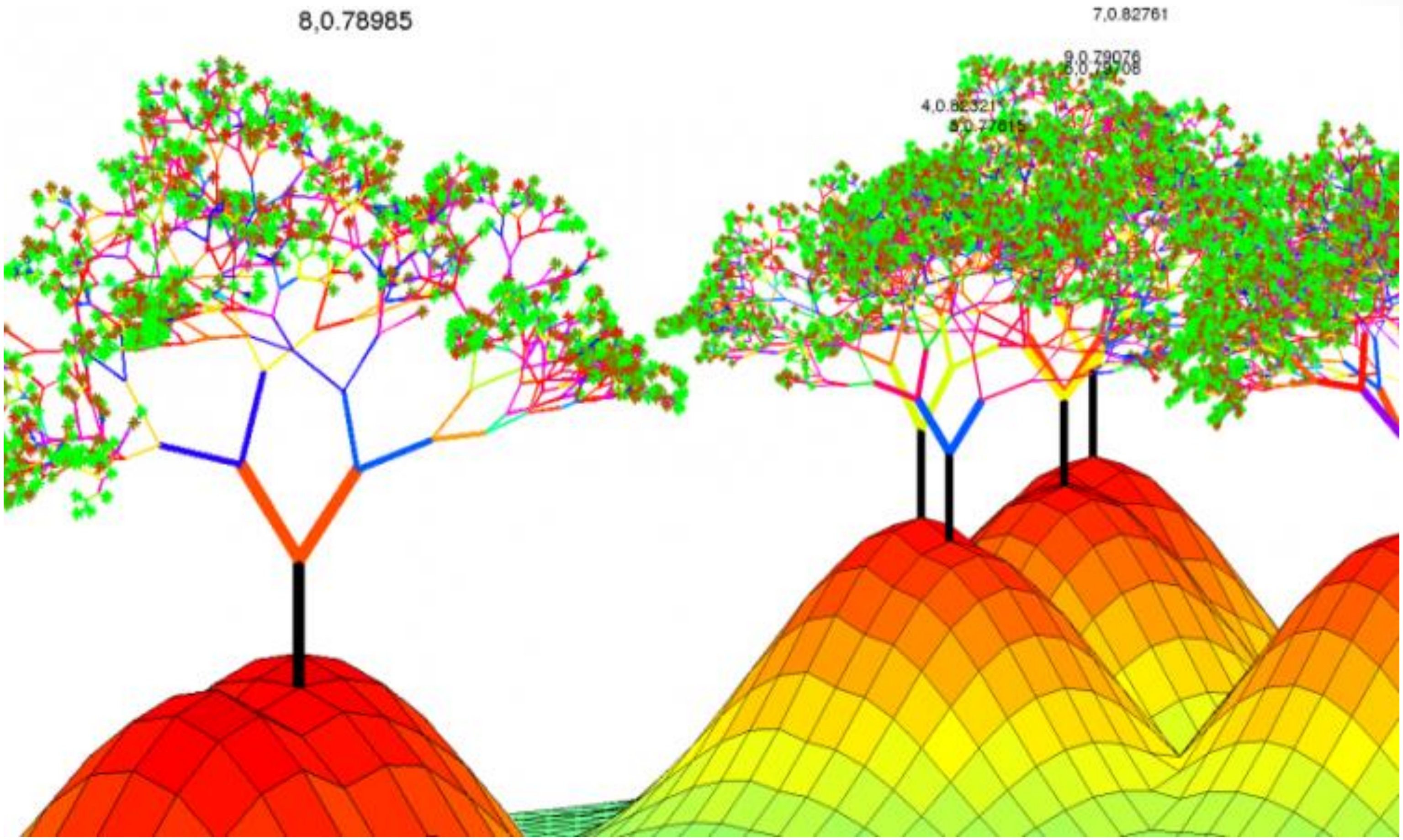
High accuracy

Runs efficiently on large database

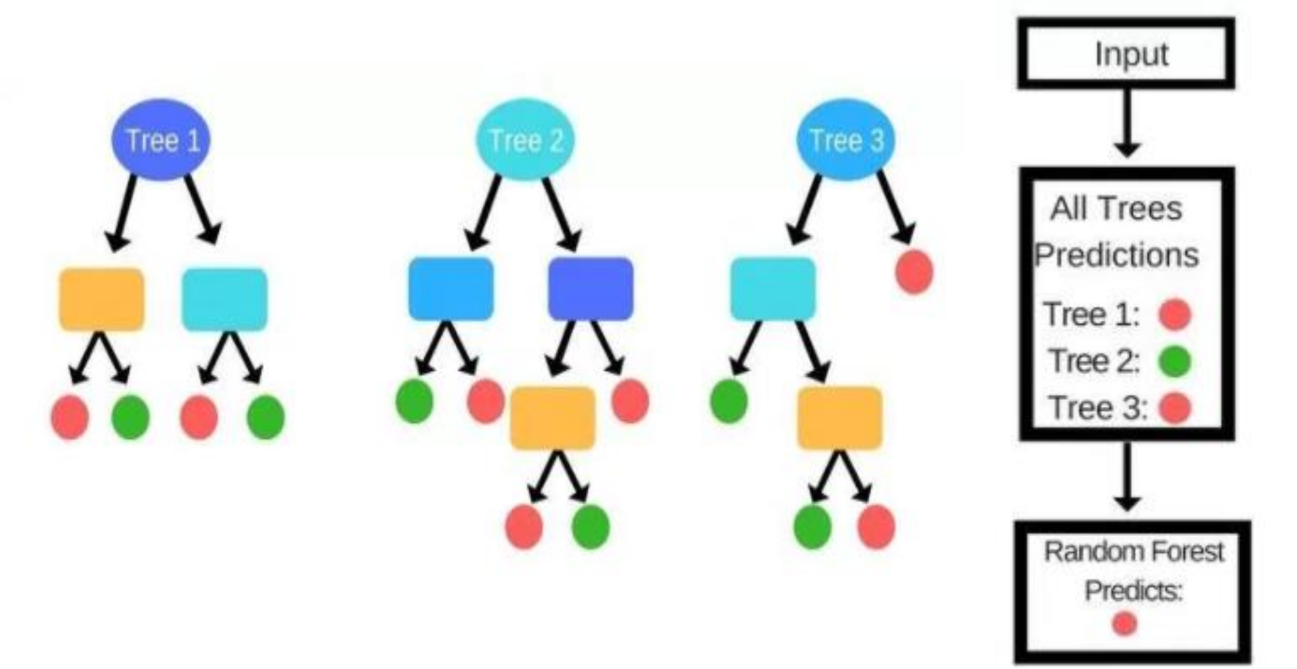
For large data, it produces highly accurate predictions

Estimates missing data

Random Forest can maintain accuracy when a large proportion of data is missing


8,0.78985
7,0.82761
9,0.79076
4,0.82321

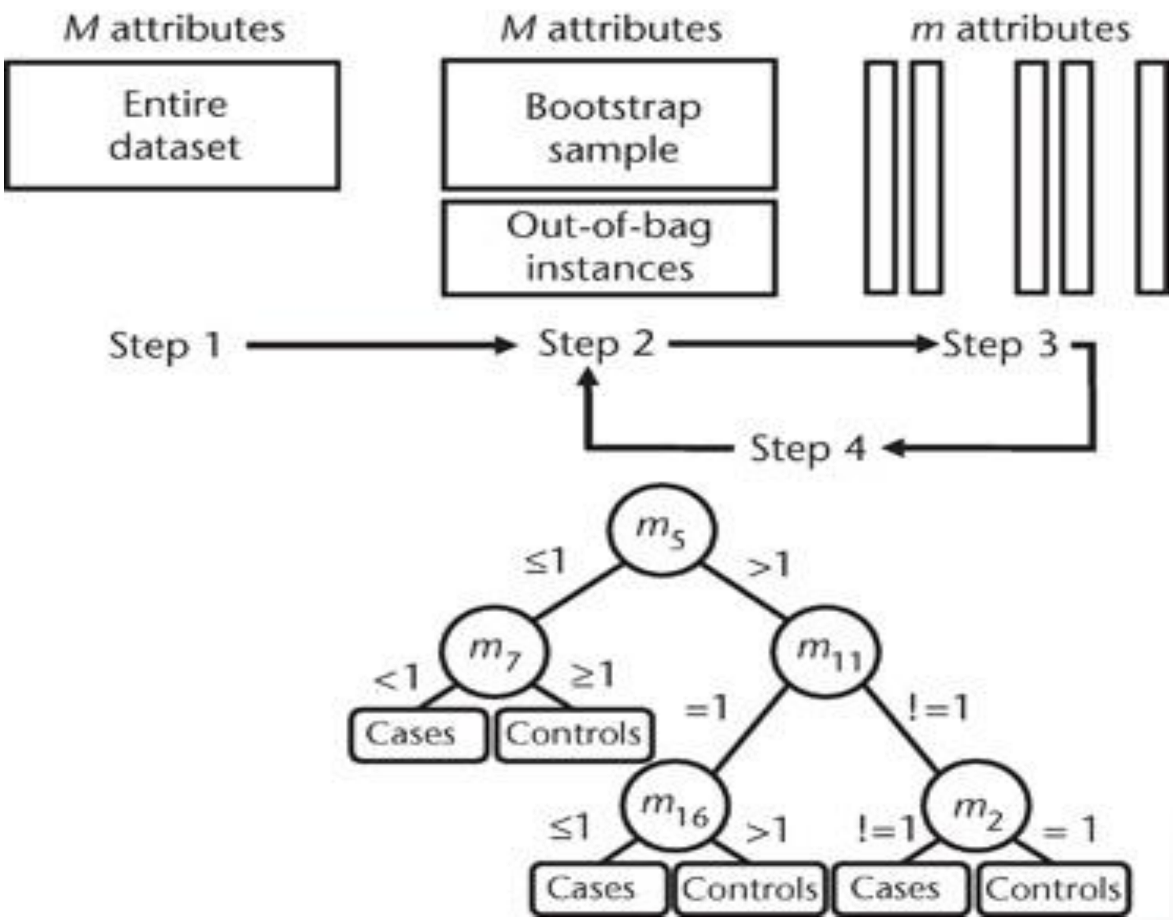
HOW THE RANDOM FOREST ALGORITHM WORKS IN MACHINE LEARNING



- Supervised learning algorithm

- Regression and classification problems

Bagging



M attributes
Entire dataset
M attributes
Bootstrap sample
Out-of-bag instances
m attributes
Step 1
Step 2
Step 3
Step 4
m_5
≤1
>1
m_7
m_{11}
<1
≥1
Cases
Controls
=1
!=1
m_{16}
m_2
≤1
>1
!=1
= 1
Cases
Controls
Cases
Controls

Random Forest pseudocode

- Randomly select **“k”** features from total **“m”** features.

  Where **k << m**

  For classification a good default is: k = sqrt(m)
  For regression a good default is: k = m/3

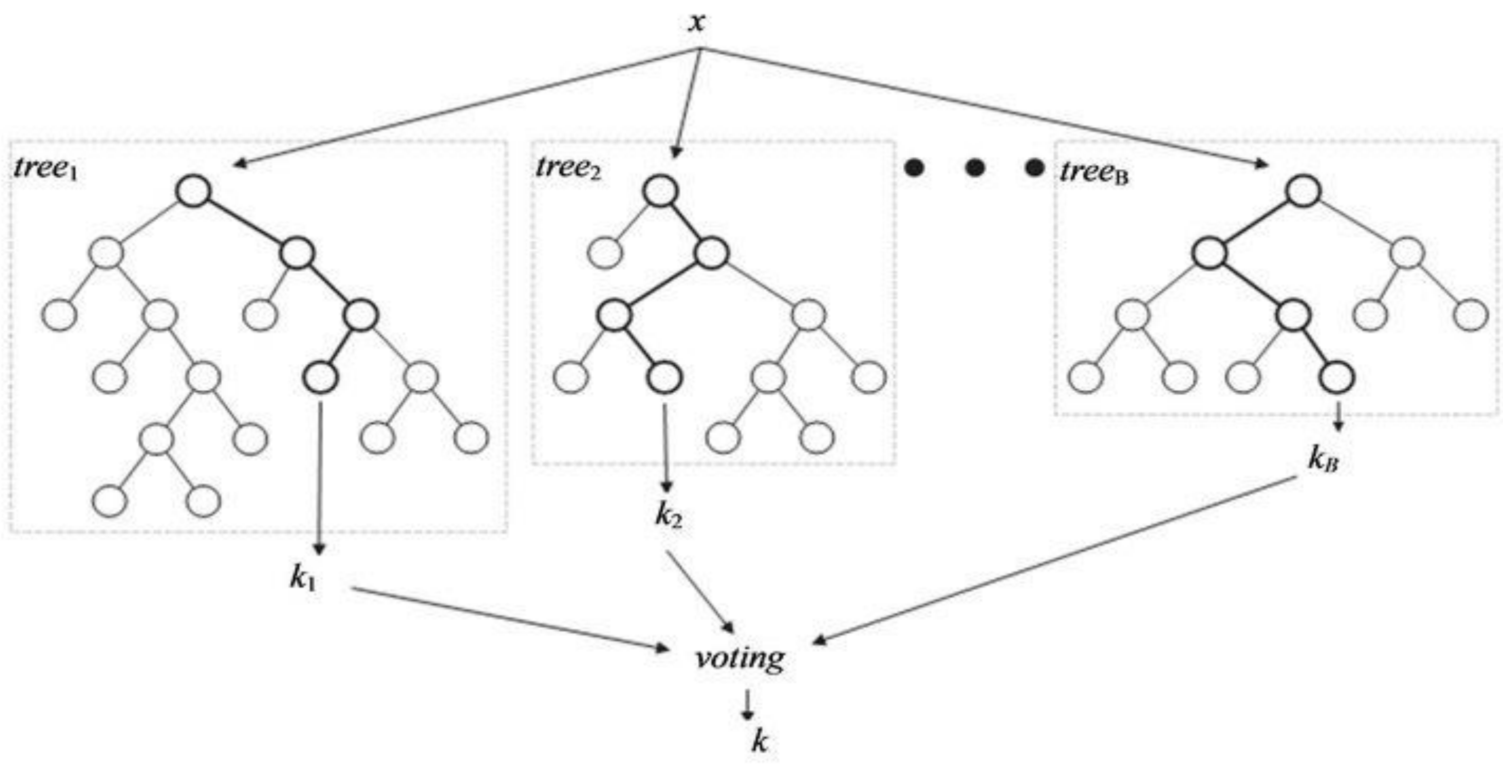
- Among the **“k”** features, calculate the node **“d”**.
- Split the node into **daughter nodes**.
- Repeat **1 to 3** steps
- Build forest by repeating steps **1 to 4** for “n” number times to create **“n” number of trees**.

Key Points

- **Majority voting**.
- **Higher the number** of trees in the forest = **High accuracy**.

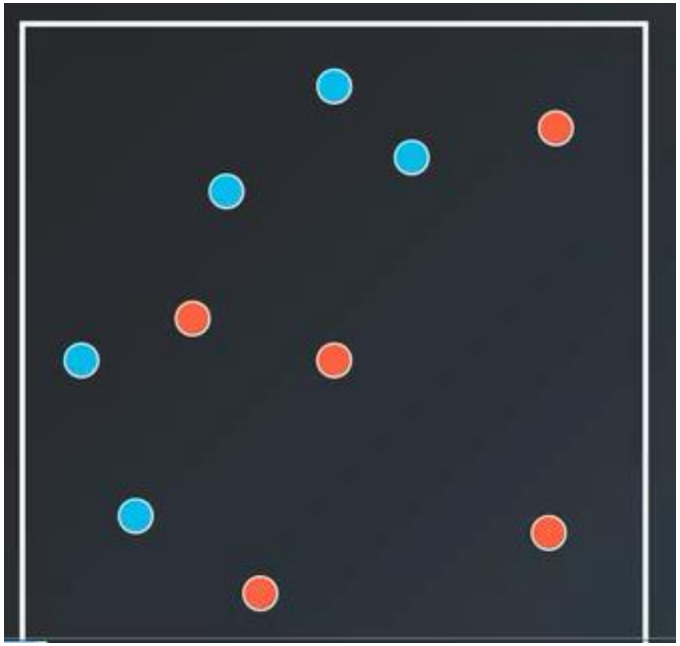
- When we have more trees in the forest, random forest classifier won't **overfit** the model.
- For each bootstrap sample taken from the training data, there will be samples left behind that were not included. These samples are called **Out-Of-Bag samples** or OOB.
- The performance of each model on its left out samples when averaged can provide an estimated accuracy of the bagged models. This estimated performance is often called the **OOB estimate of performance**.

Random Forest - Skeleton

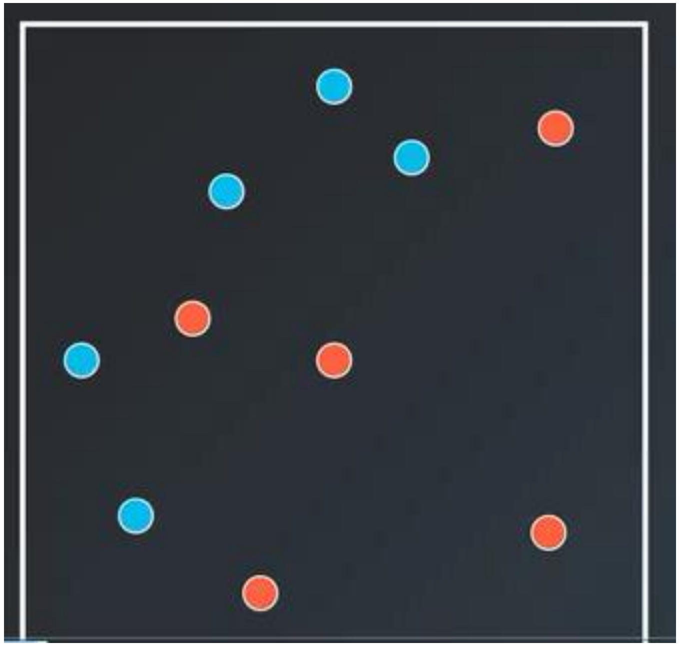


Boosting

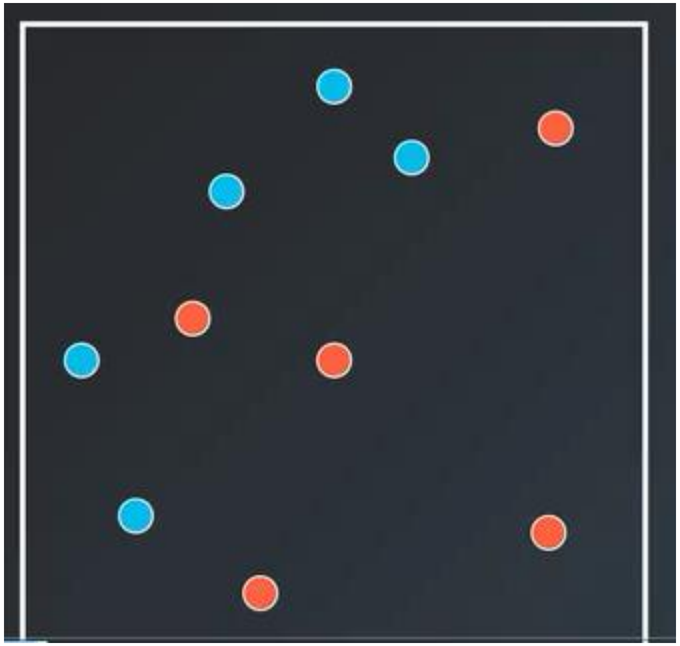
AdaBoost (**Ada**ptive **Boost**ing)

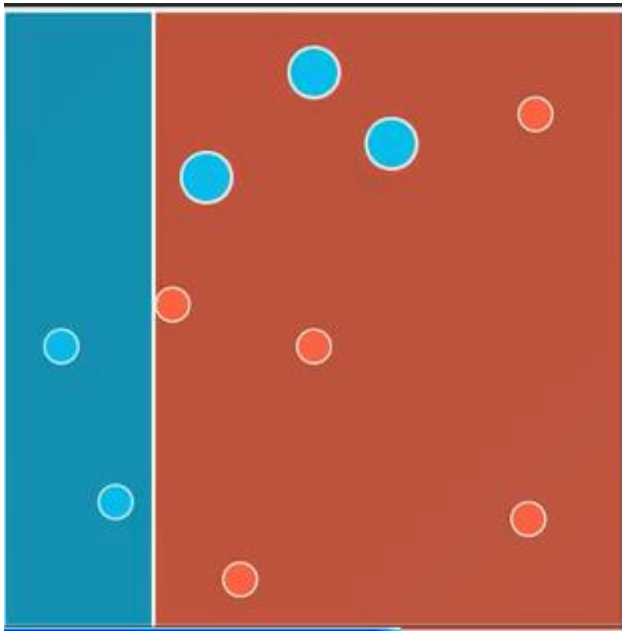


AdaBoost – Pattern 1

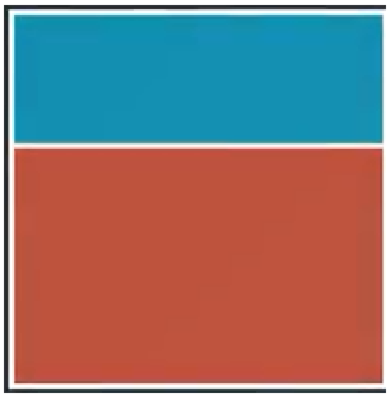


AdaBoost – Pattern 1





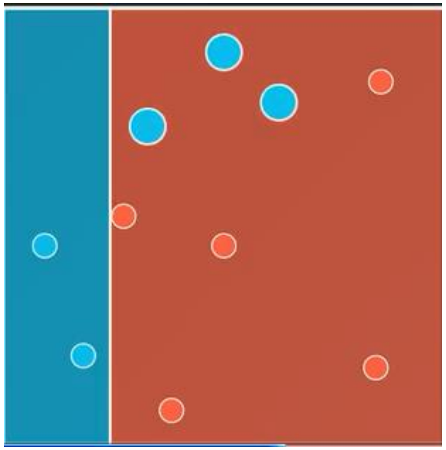
AdaBoost – Pattern 2

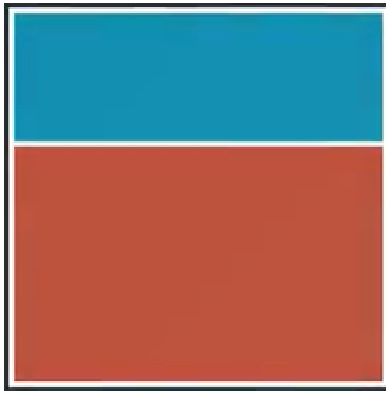


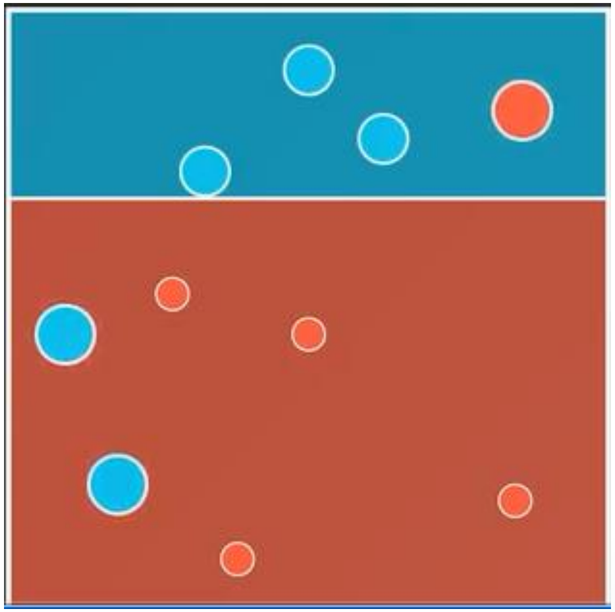
Apply pattern 2 on the Input Data from pattern 1

AdaBoost – Pattern 2

Input Data





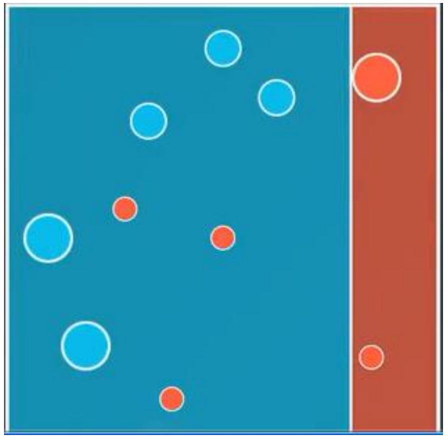


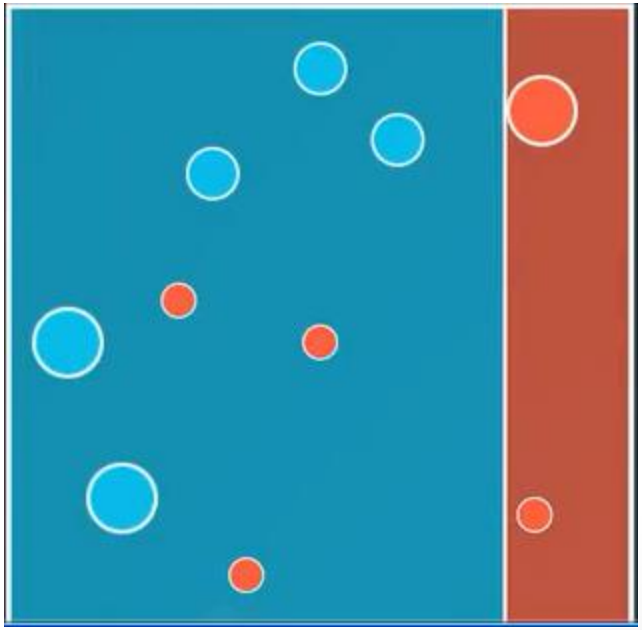
AdaBoost – Pattern 3

Apply pattern 3 on the Input Data from pattern 2

AdaBoost – Pattern 3

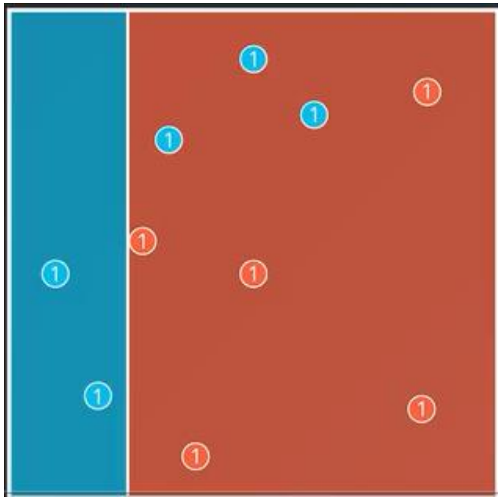
Input Data



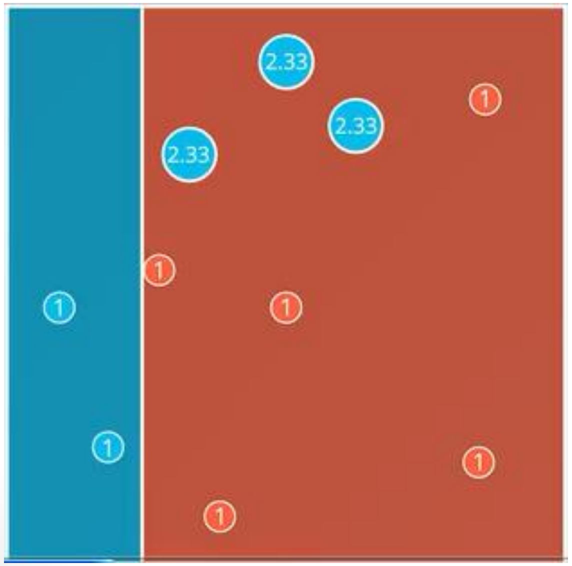


AdaBoost – Pattern 1

Weights after applying pattern 1



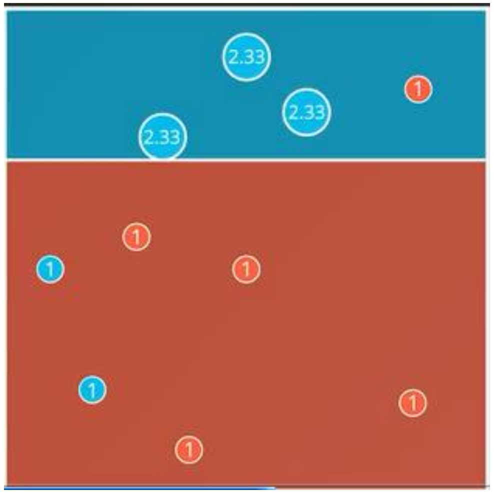
Correct: 7
Incorrect: 3



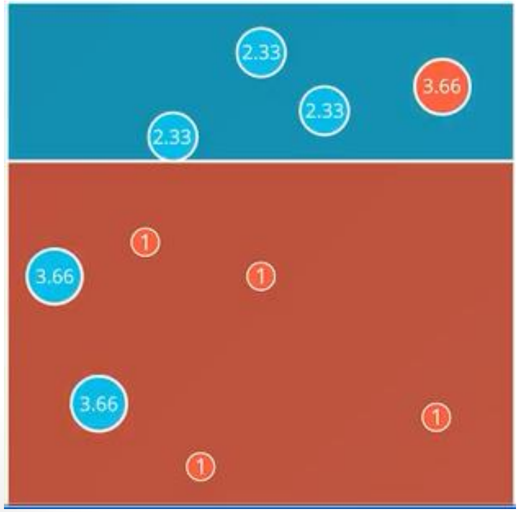
Correct: 7
Incorrect: 7

AdaBoost – Pattern 2

Weights after applying pattern 2



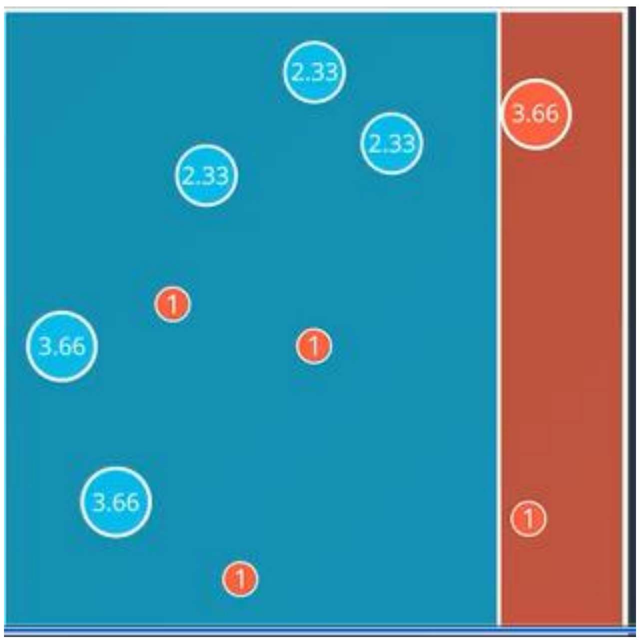
Correct: 11
Incorrect: 3



Correct: 11
Incorrect: 11

AdaBoost – Pattern 3

Weights after applying pattern 3



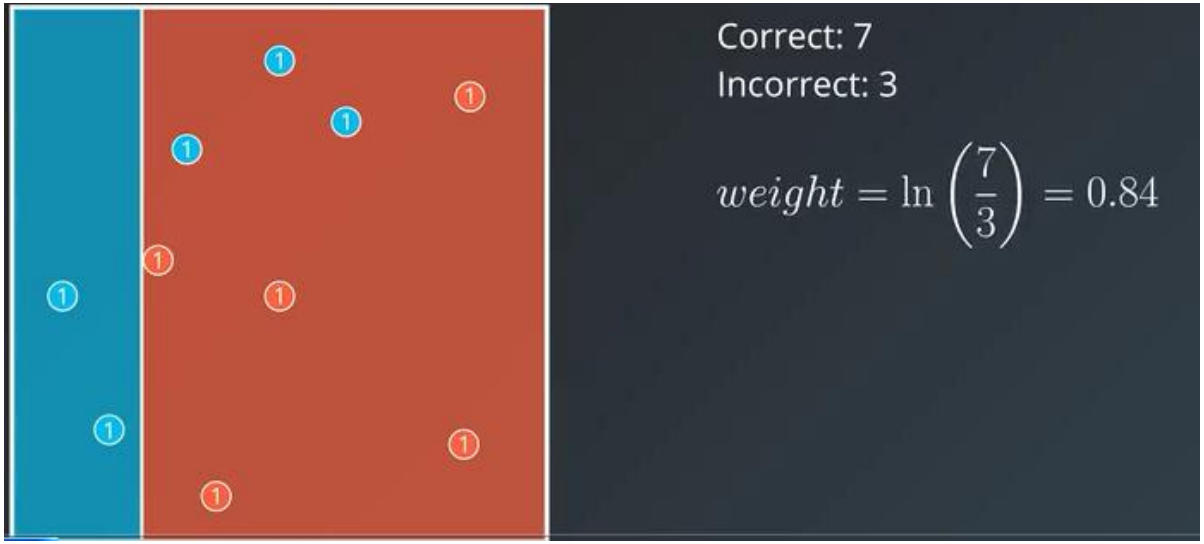
Correct: 19
Incorrect: 3

AdaBoost – 3 Models

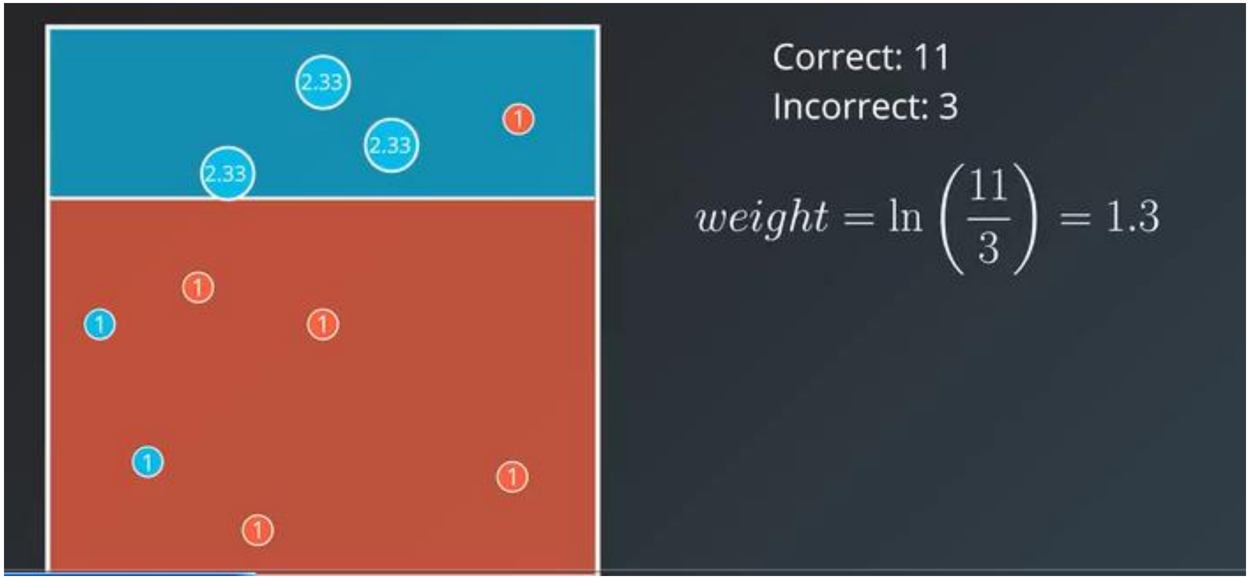
Weightage of a Model

$$weight = \ln\left(\frac{\#correct}{\#incorrect}\right)$$

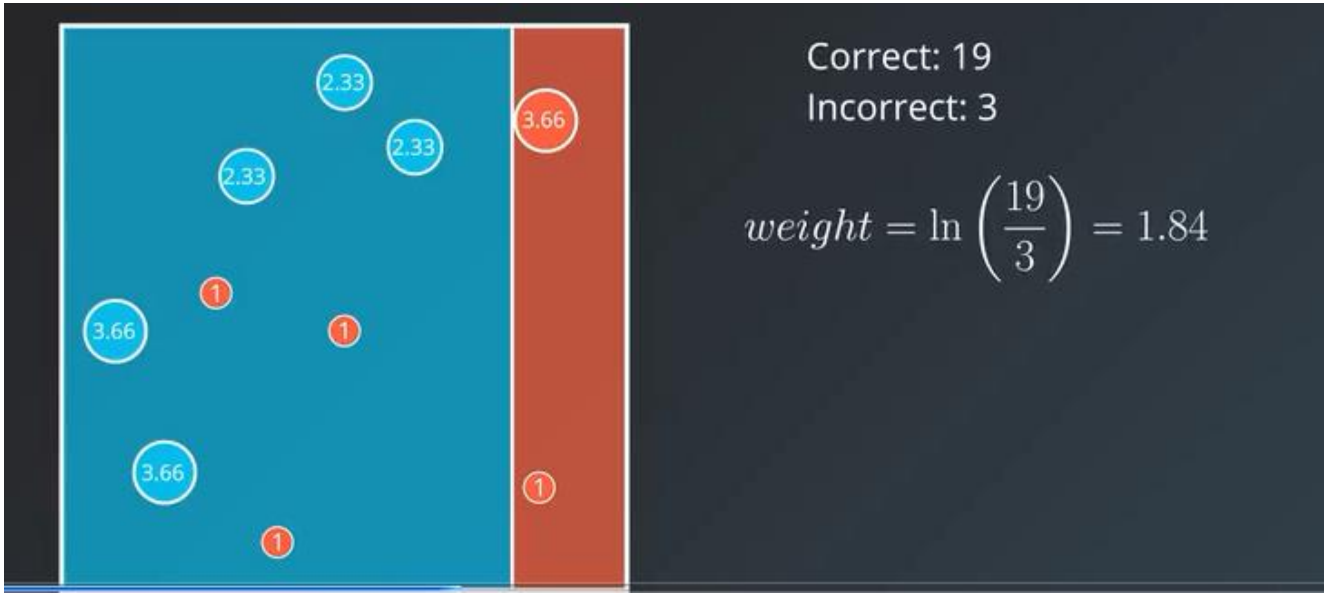
Weight of Model 1



Weight of Model 2



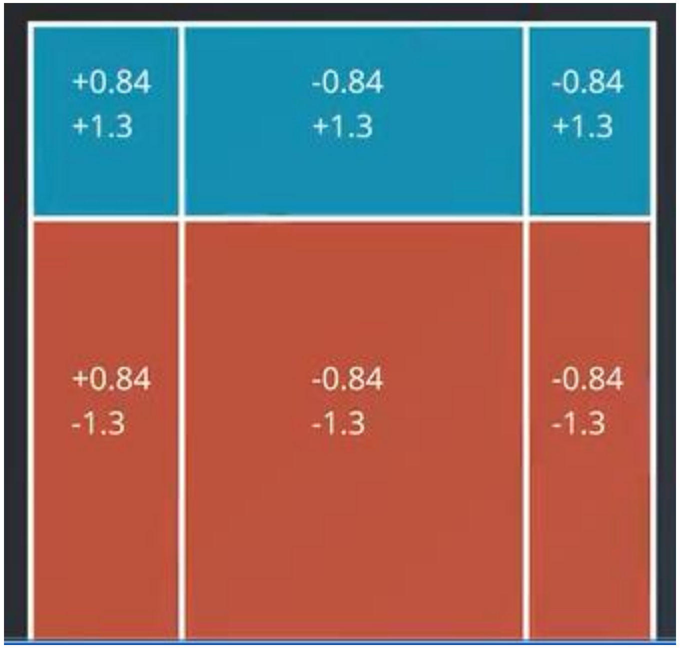
Weight of Model 3

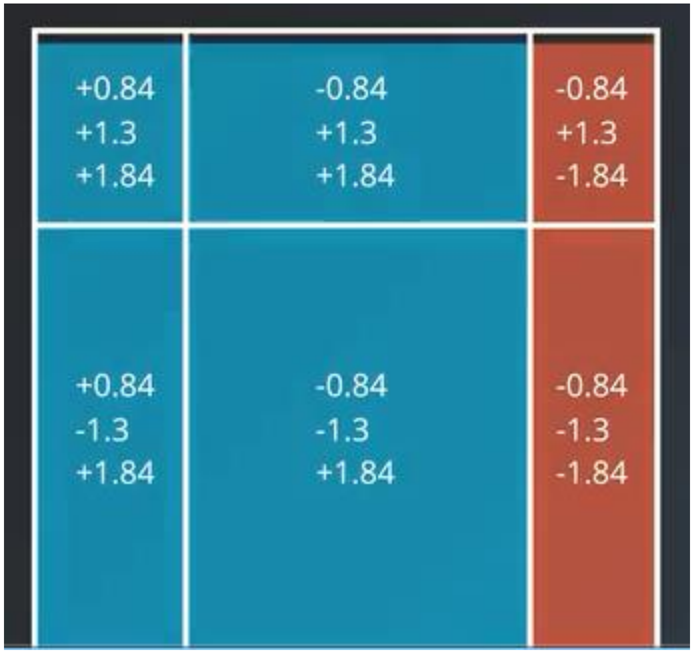


Weight of 3 Models

Assinging weights to 2 categories


+0.84
-0.84
-0.84
+0.84
-0.84
-0.84

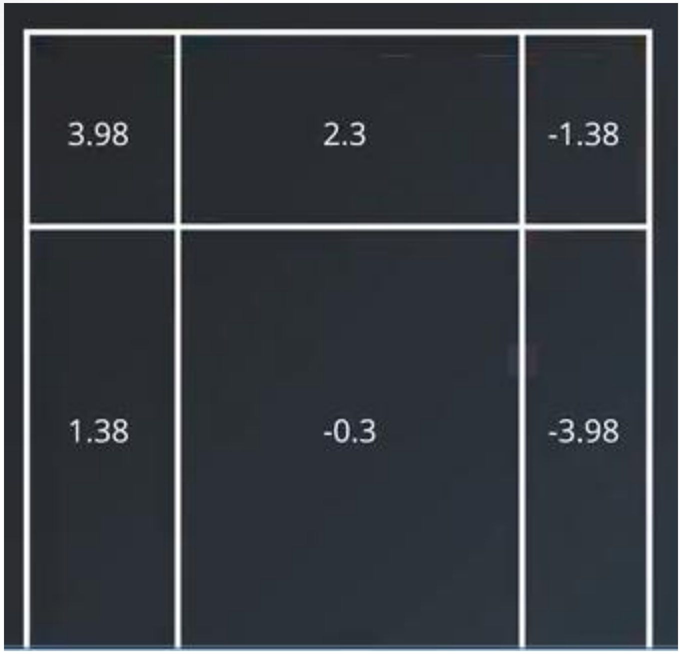

+0.84
+1.3
-0.84
+1.3
-0.84
+1.3
+0.84
-1.3
-0.84
-1.3
-0.84
-1.3

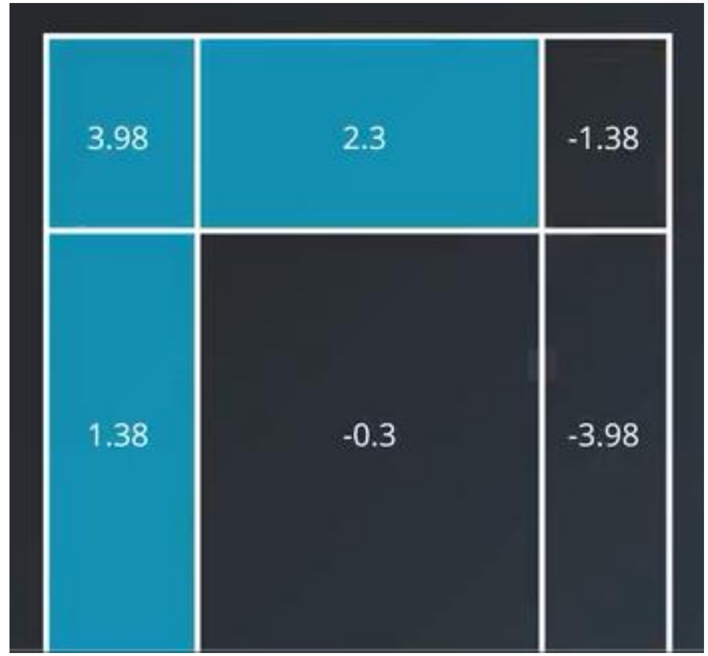

+0.84
+1.3
+1.84
-0.84
+1.3
+1.84
-0.84
+1.3
-1.84
+0.84
-1.3
+1.84
-0.84
-1.3
+1.84
-0.84
-1.3
-1.84

3.98	2.3	-1.38
1.38	-0.3	-3.98

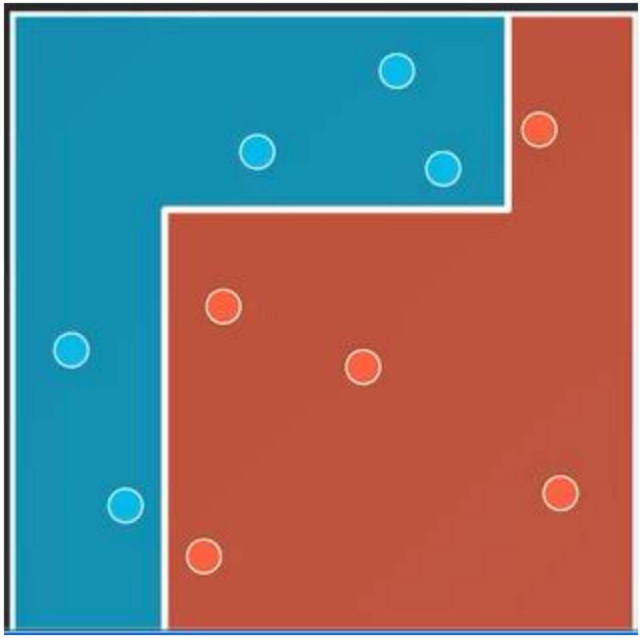
3.98	2.3	-1.38
1.38	-0.3	-3.98

3.98	2.3	-1.38
1.38	-0.3	-3.98


3.98
2.3
-1.38
1.38
-0.3
-3.98


3.98
2.3
-1.38
1.38
-0.3
-3.98


3.98
2.3
-1.38
1.38
-0.3
-3.98



K – Means

Un-Supervised learning algorithm

Clustering

No dependant variable

Pseudocode

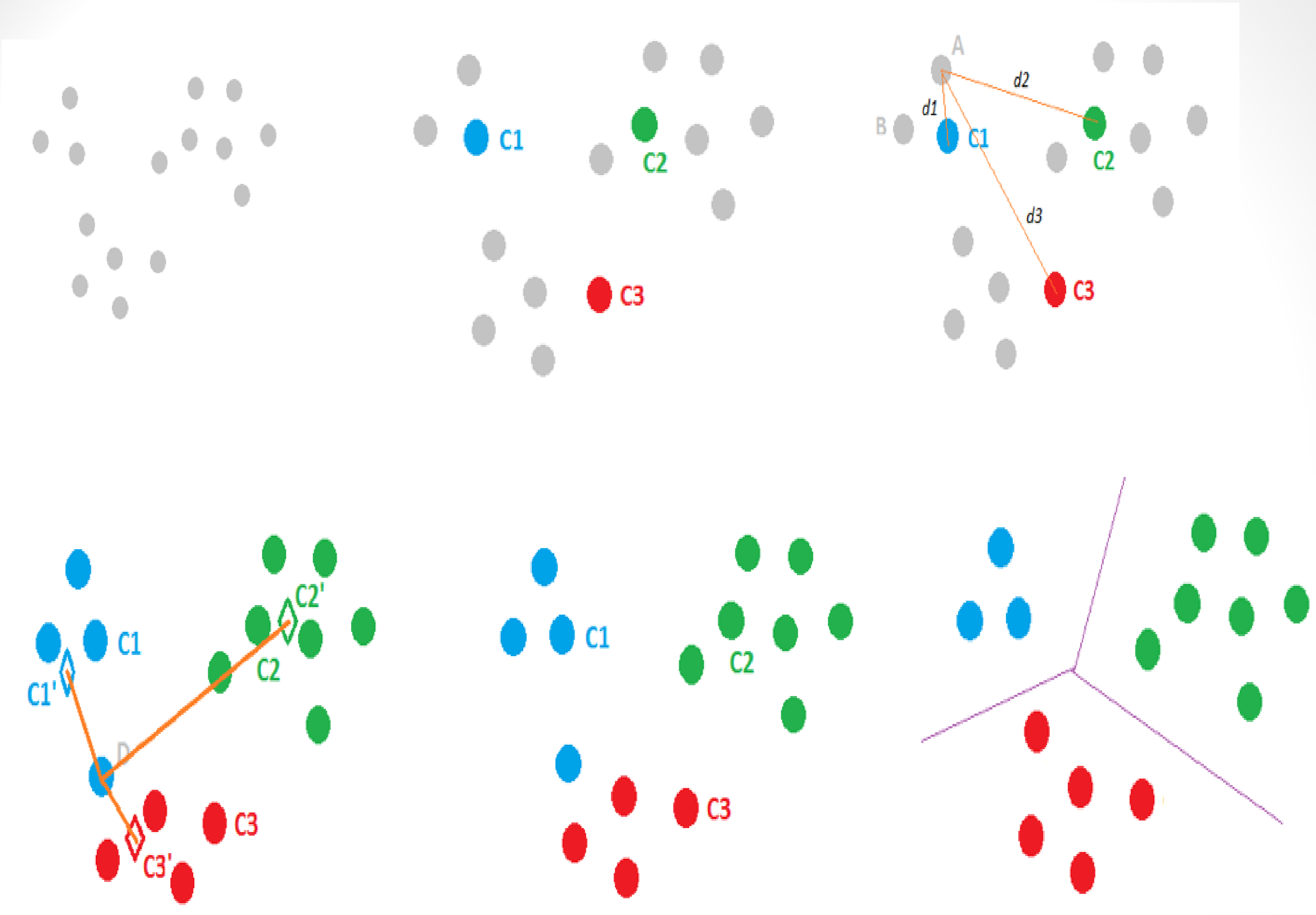
- Input the algorithm with the number of clusters **K** and the data set.
- Randomly generate or randomly select K centroids from the data set.

The algorithm then iterates between two steps:

1. Data assignment step

$$\underset{c_i \in C}{\operatorname{argmin}}\ dist(c_i, x)^2$$

where *dist*(·) is the standard (L_2)
Euclidean distance


C1
C2
C3
A
d2
d1
B
C1
C2
d3
C3
C2'
C1
C2
C1'
D
C3
C3'
C1
C2
C3

2. Centroid update step:

In this step, the centroids are recomputed. This is done by taking the mean of all data points assigned to that centroid's cluster.

$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

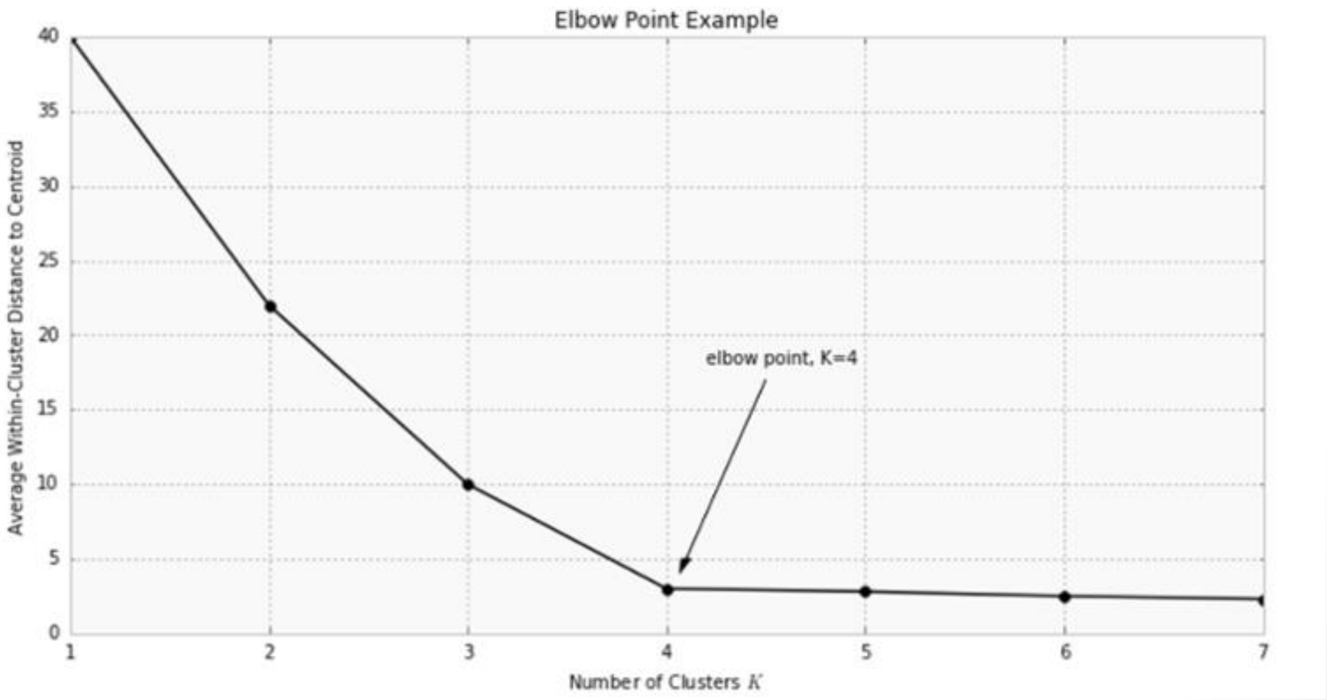
The algorithm iterates between steps one and two

1. No data points change clusters

2. The sum of the distances is minimized or some maximum number of iterations is reached

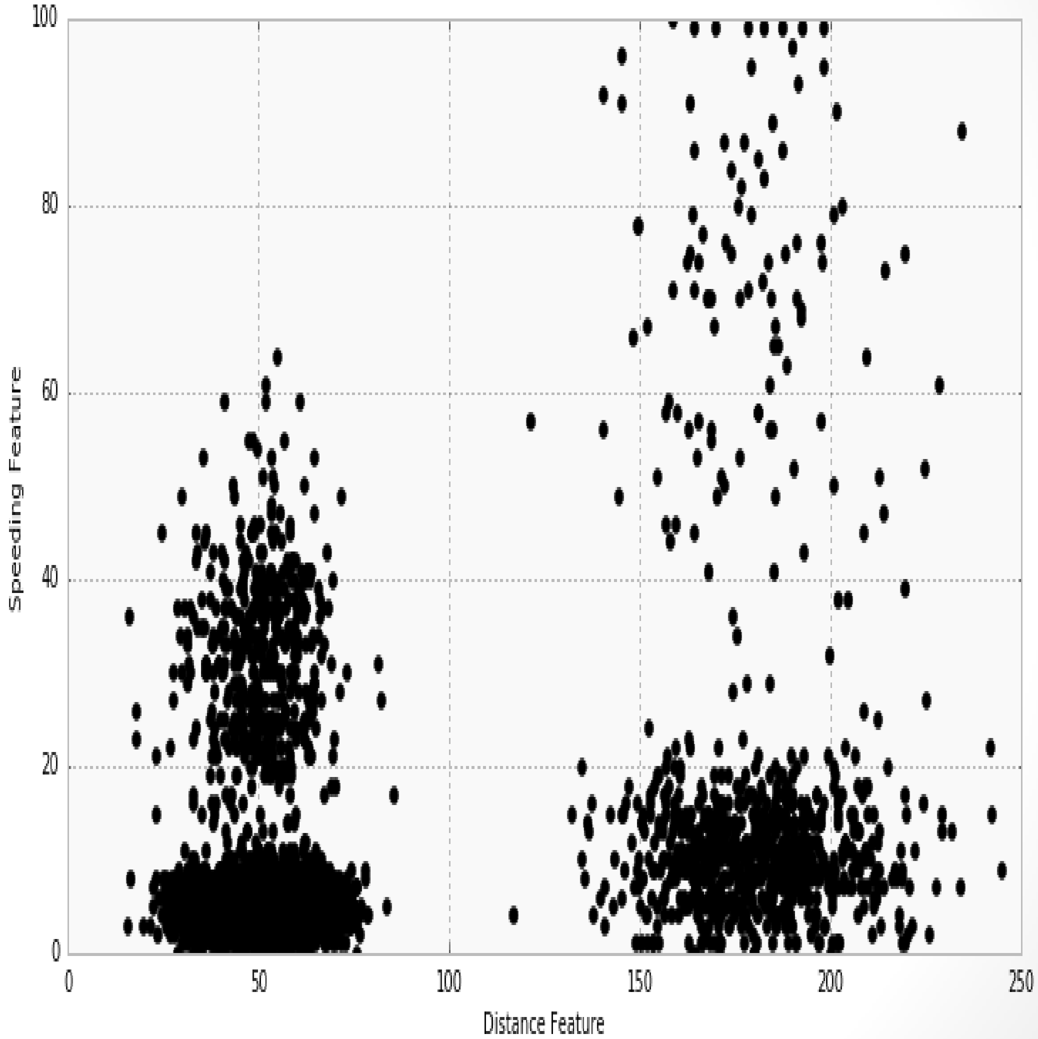
Choosing *K* – K Means ++

Run the *K*-means clustering algorithm for a range of *K* values

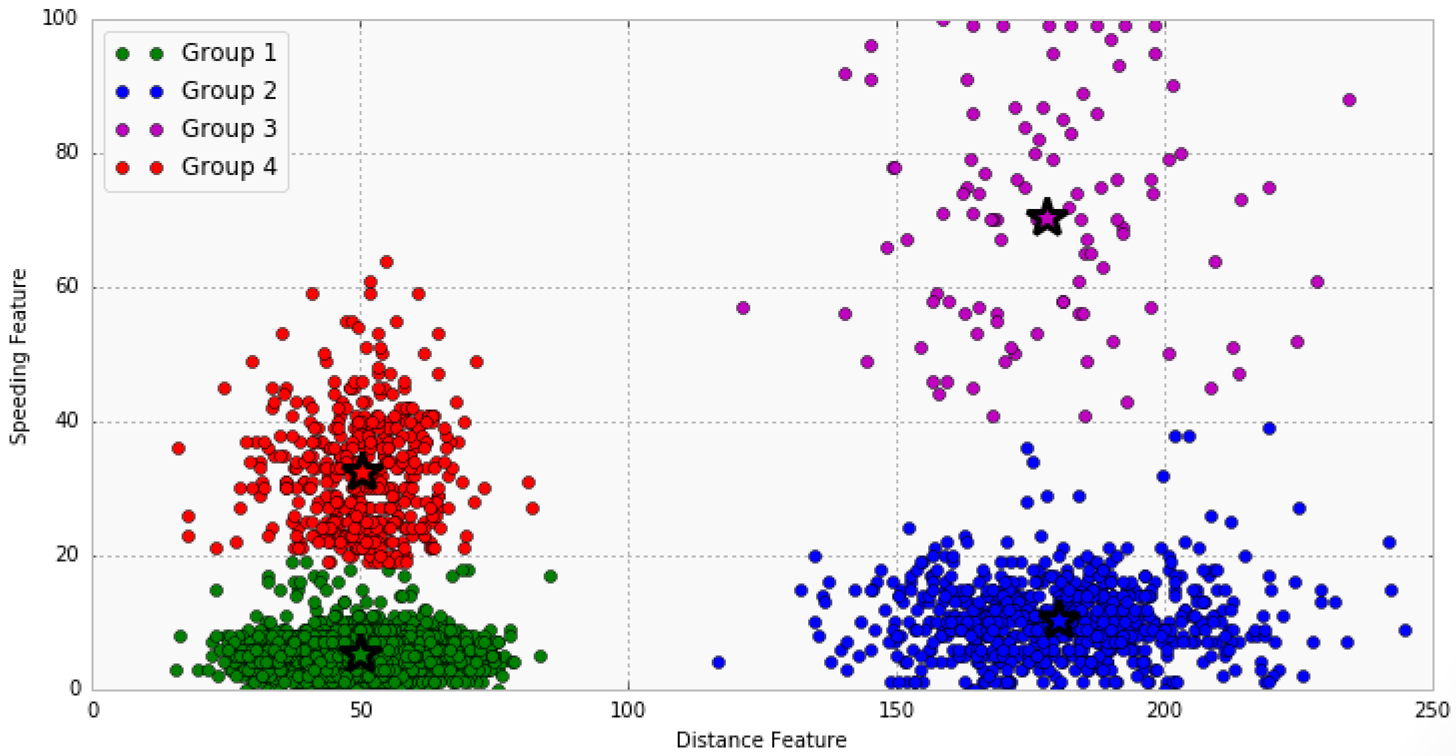


Distance and Speed

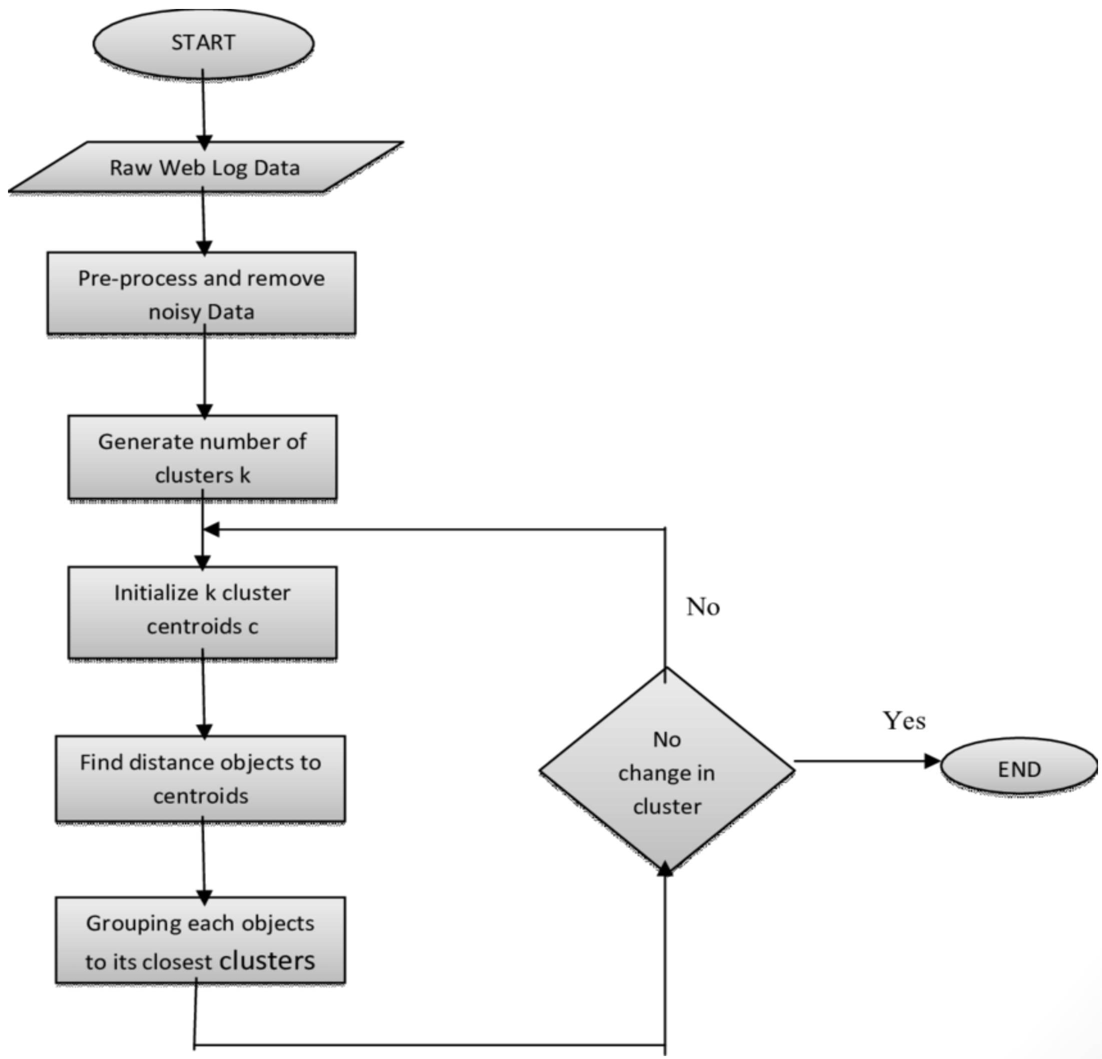
ID	Distance	Speed
1	75	60
2	55	50
3	64	55
4	20	30
5	45	40
.	.	.
.	.	.
.	.	.
.	.	.
.	.	.
4000	150	110



Graph


Group 1
Group 2
Group 3
Group 4
Speeding Feature
Distance Feature
0
20
40
60
80
100
0
50
100
150
200
250

Flow Chart


START
Raw Web Log Data
Pre-process and remove noisy Data
Generate number of clusters k
Initialize k cluster centroids c
Find distance objects to centroids
Grouping each objects to its closest clusters
No change in cluster
No
Yes
END

Key Points

- No prediction – The interest is group to similar kind of attributes to a common class

Example –

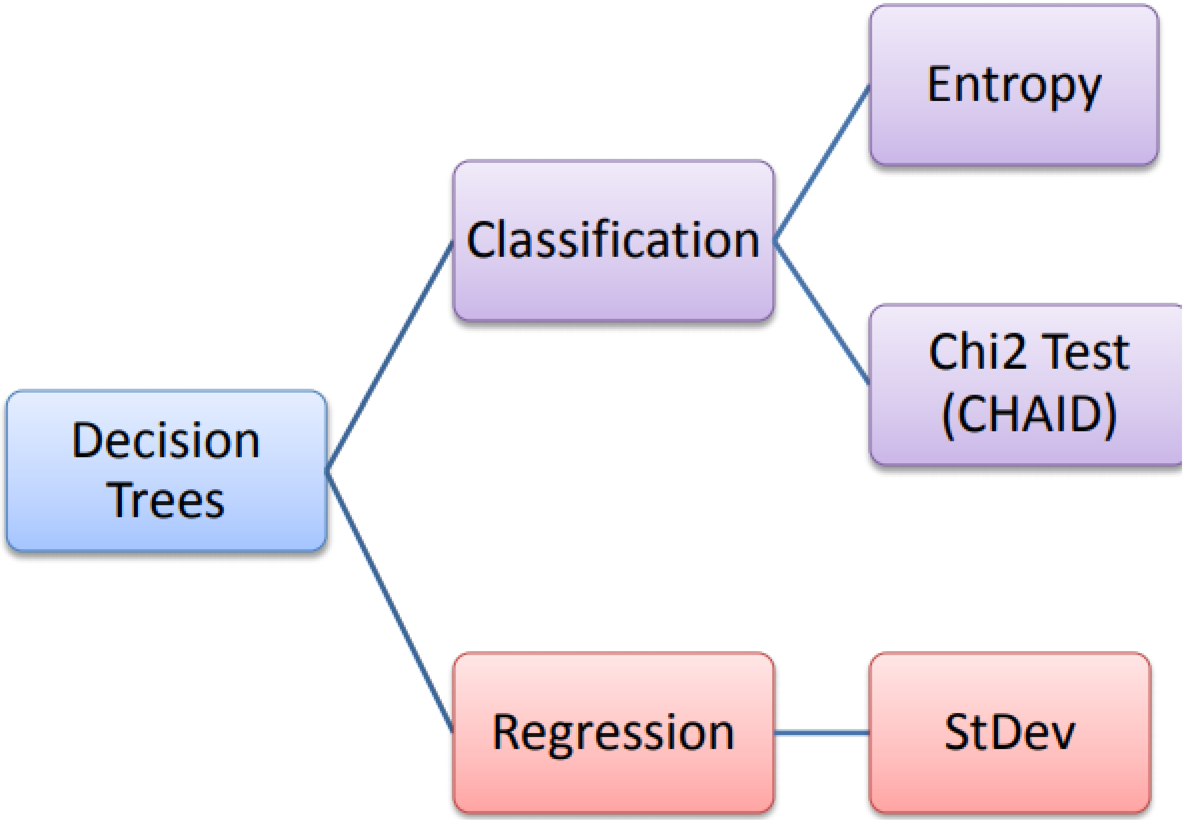
- Same language documents are one group.
- While categorising the news articles (Same news category(Sport) articles are one group)

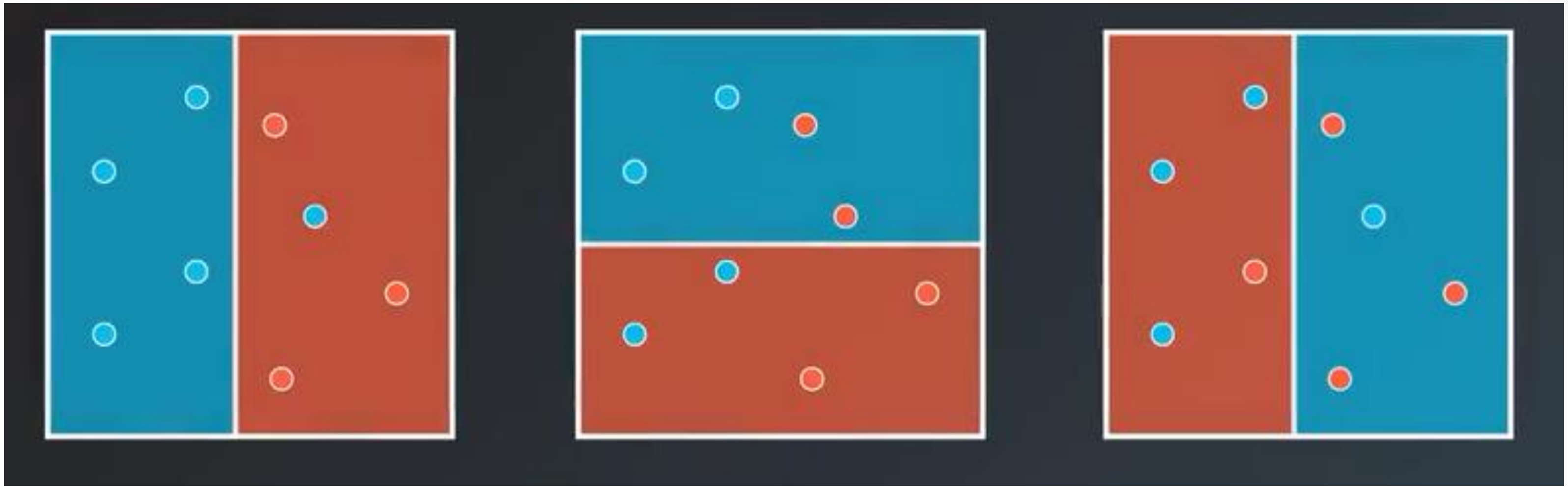
Result of K- means

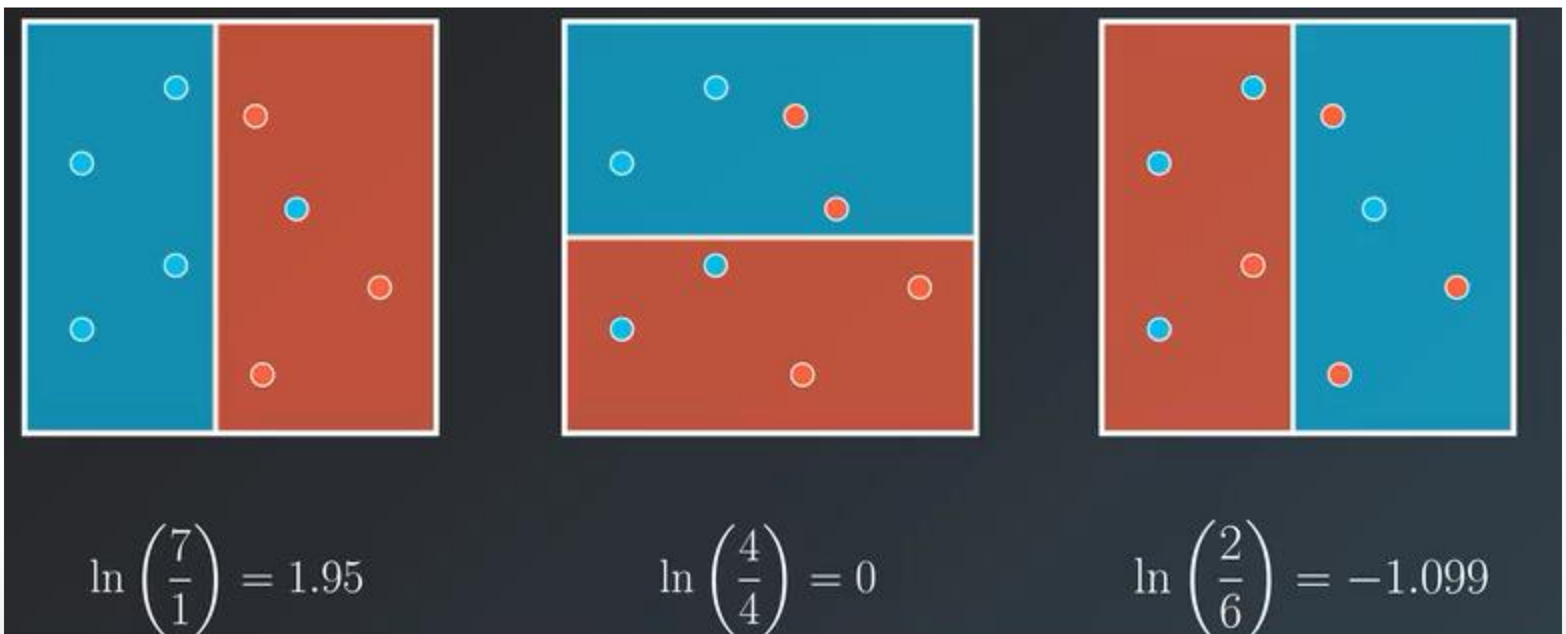
1. The centroids of the K clusters, which can be used to label new data

2. Labels for the training data (each data point is assigned to a single cluster)

Classification vs Regression Tree






$\ln\left(\frac{7}{1}\right) = 1.95$
$\ln\left(\frac{4}{4}\right) = 0$
$\ln\left(\frac{2}{6}\right) = -1.099$