DECISION TREE

- **C**l**A**ssification and **R**egression **T**ree (CART)

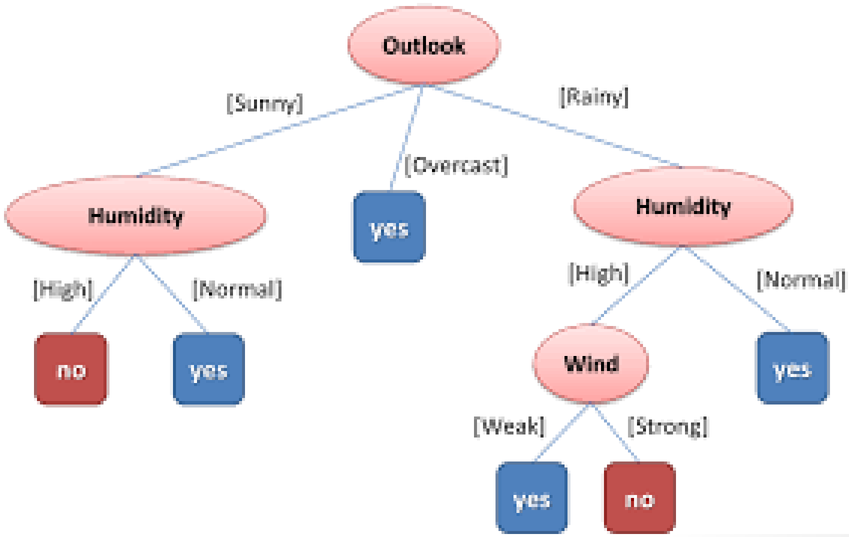
Supervised learning algorithm

Root Node - Outlook

Decision node - Humidity/Wind

Leaves - Yes/No

Structure of a Tree



HOW DECISION TREE ALGORITHM WORKS (HOW TO FIND ROOT)

Attribute selection measures

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

Information Gain

Information Gain -> Information theory -> 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

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

The measure of uncertainty

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)

= Entropy (0.36, 0.64)

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

= 0.94

Frequency Table

		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

Outlook - Entropy

		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

Outlook - Information Gain

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

$$= 0.940 - 0.693 = 0.247$$

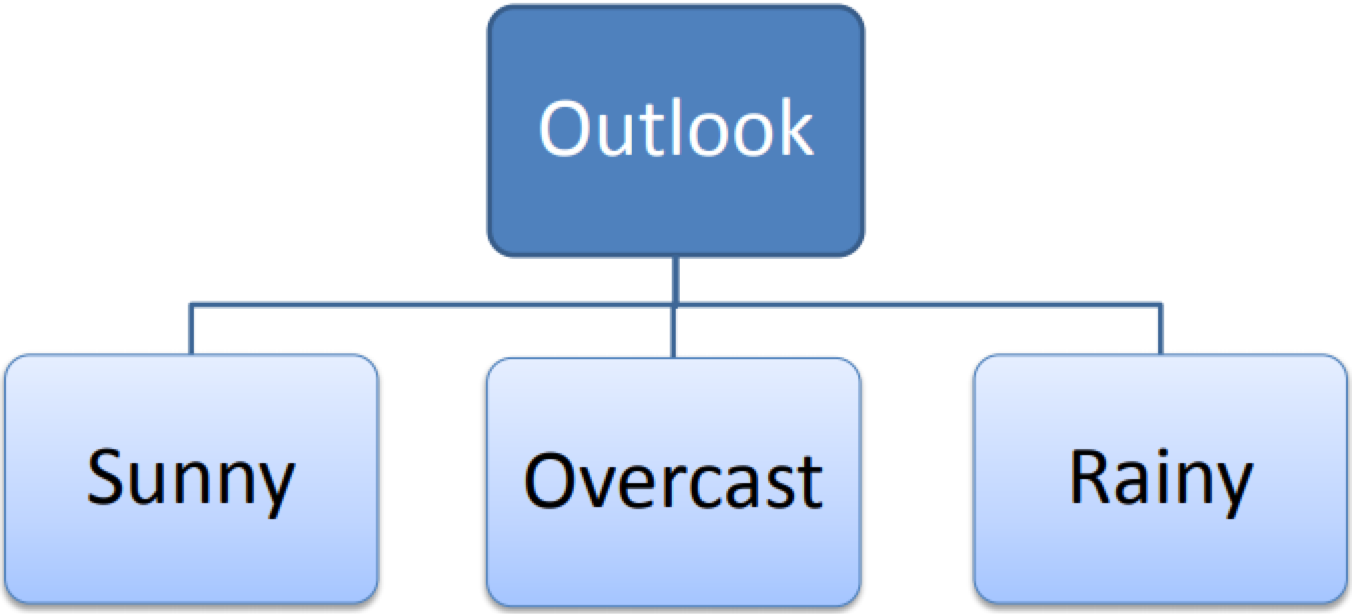
All Attributes - Information Gain

		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			

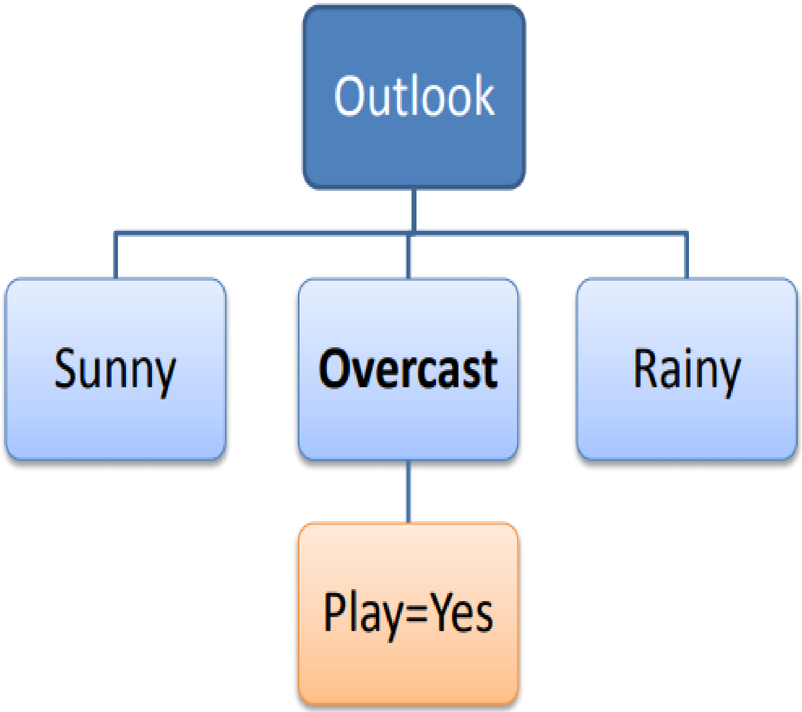
		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			


Outlook
Sunny
Overcast
Rainy

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
Hot	High	FALSE	Yes



Sunny

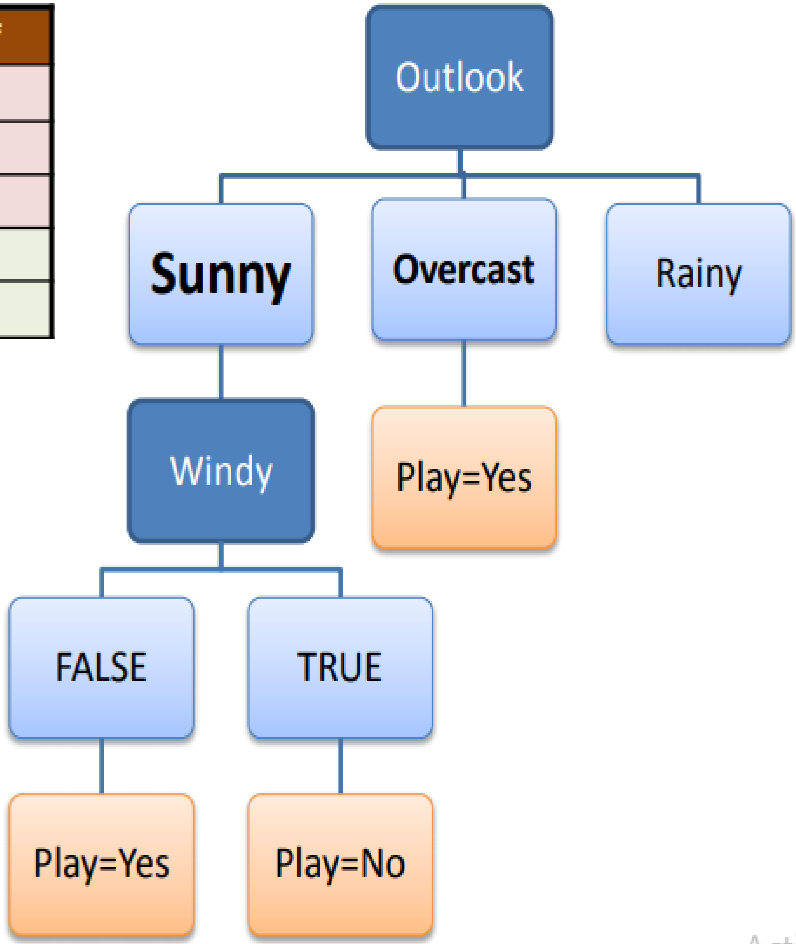
Temp.	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	Normal	FALSE	Yes
Mild	High	TRUE	No

		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			

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

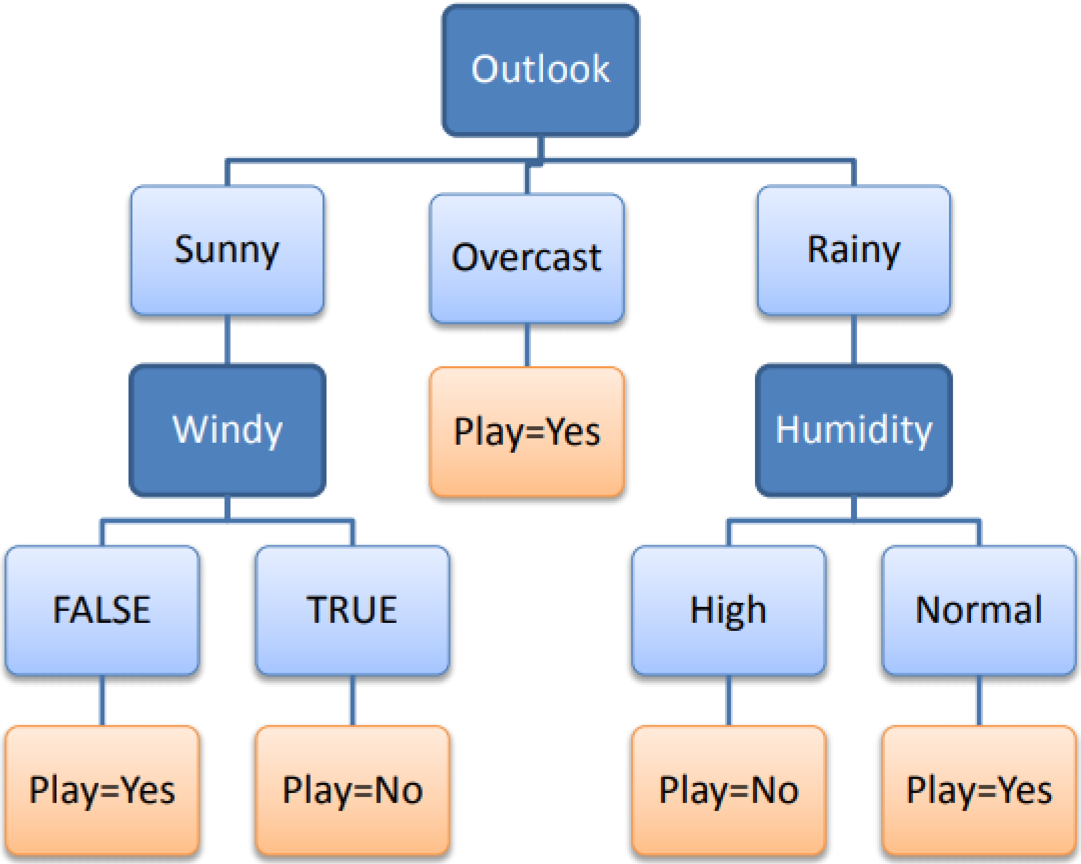
Temp.	Humidity	Windy	Play Golf
Hot	High	FALSE	No
Hot	High	TRUE	No
Mild	High	FALSE	No
Cool	Normal	FALSE	Yes
Mild	Normal	TRUE	Yes

		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			

Tree


Outlook
Sunny
Overcast
Rainy
Windy
Play=Yes
Humidity
FALSE
TRUE
High
Normal
Play=Yes
Play=No
Play=No
Play=Yes

Predit the Play – D15 ?

Sunny – Cool – Normal – False

Predit the Play – D15 ?

Sunny – Cool – Normal – False = **Play**

Decision Rules

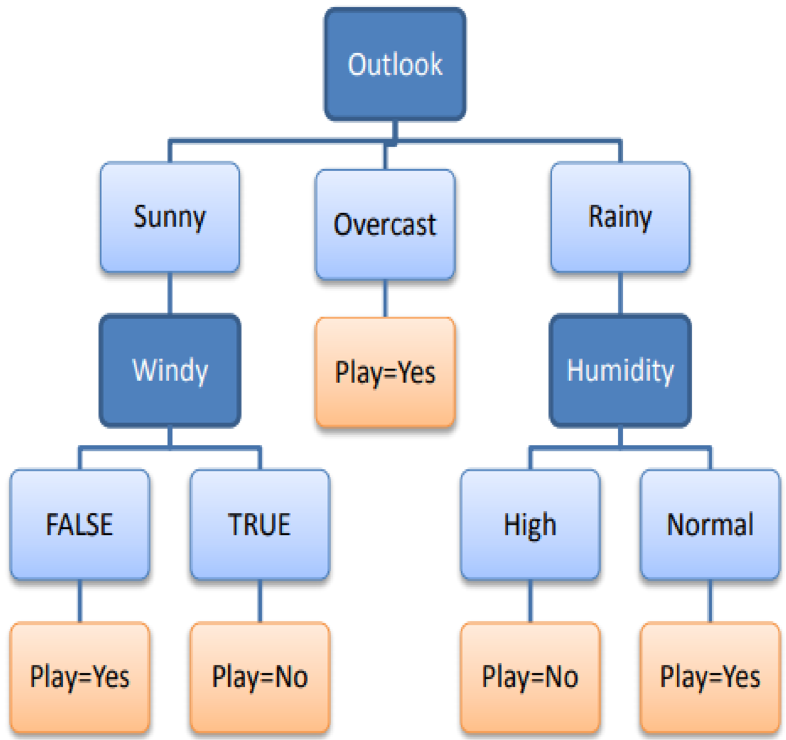
R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

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

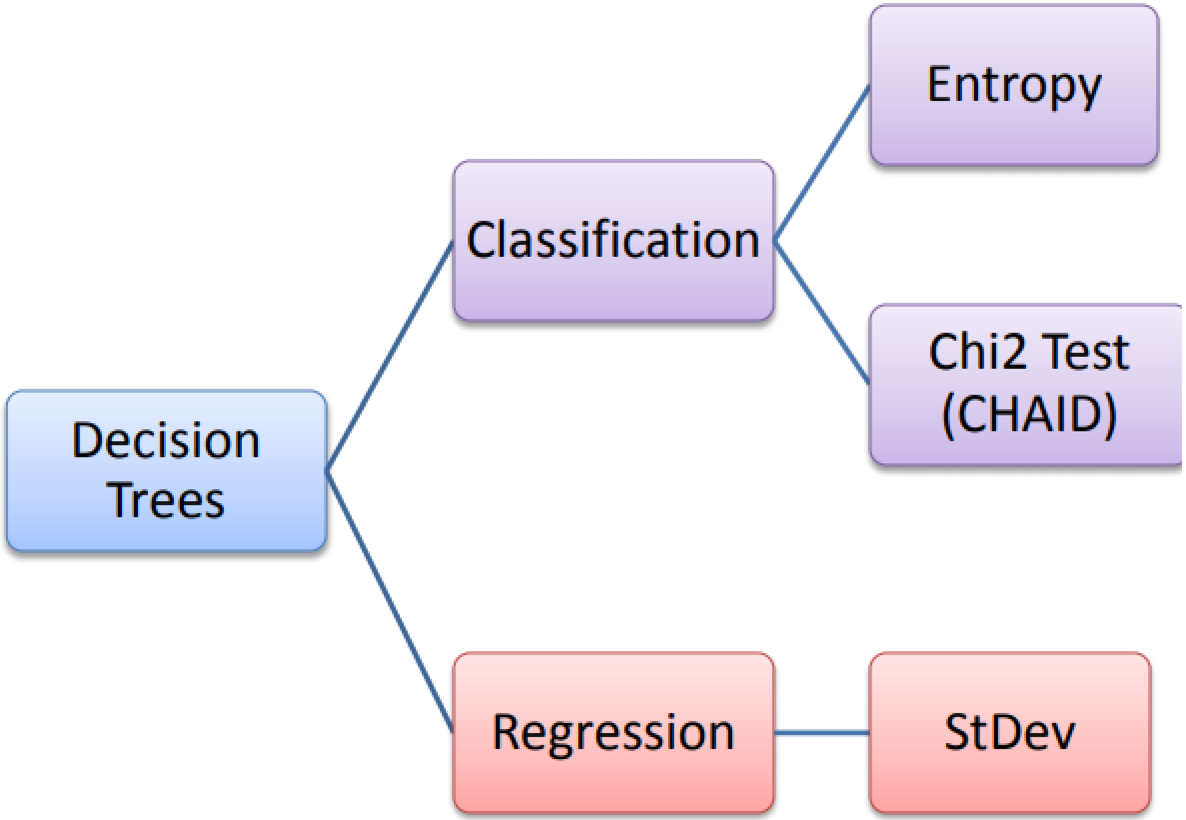
R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes

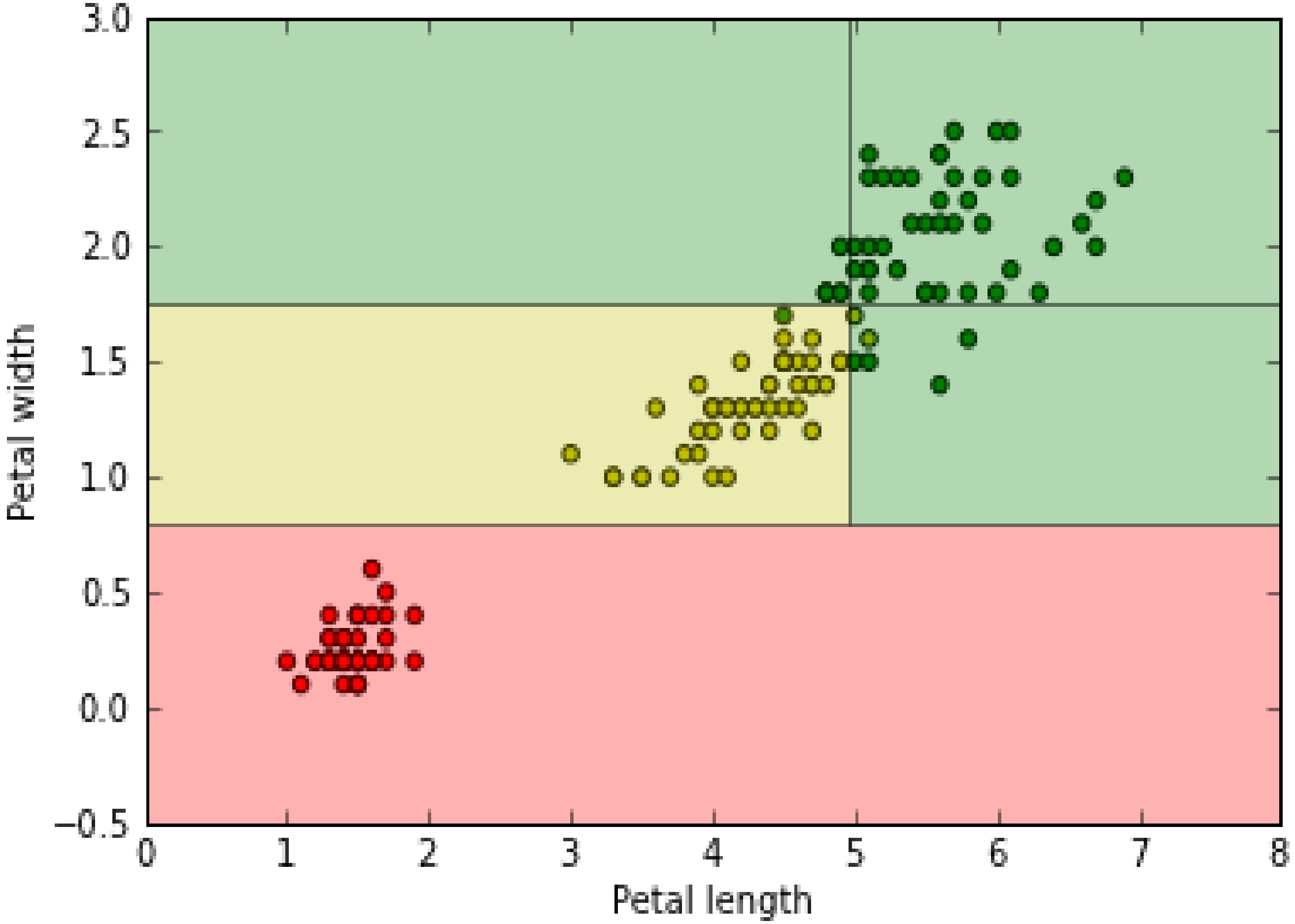


Decision Tree – Gini

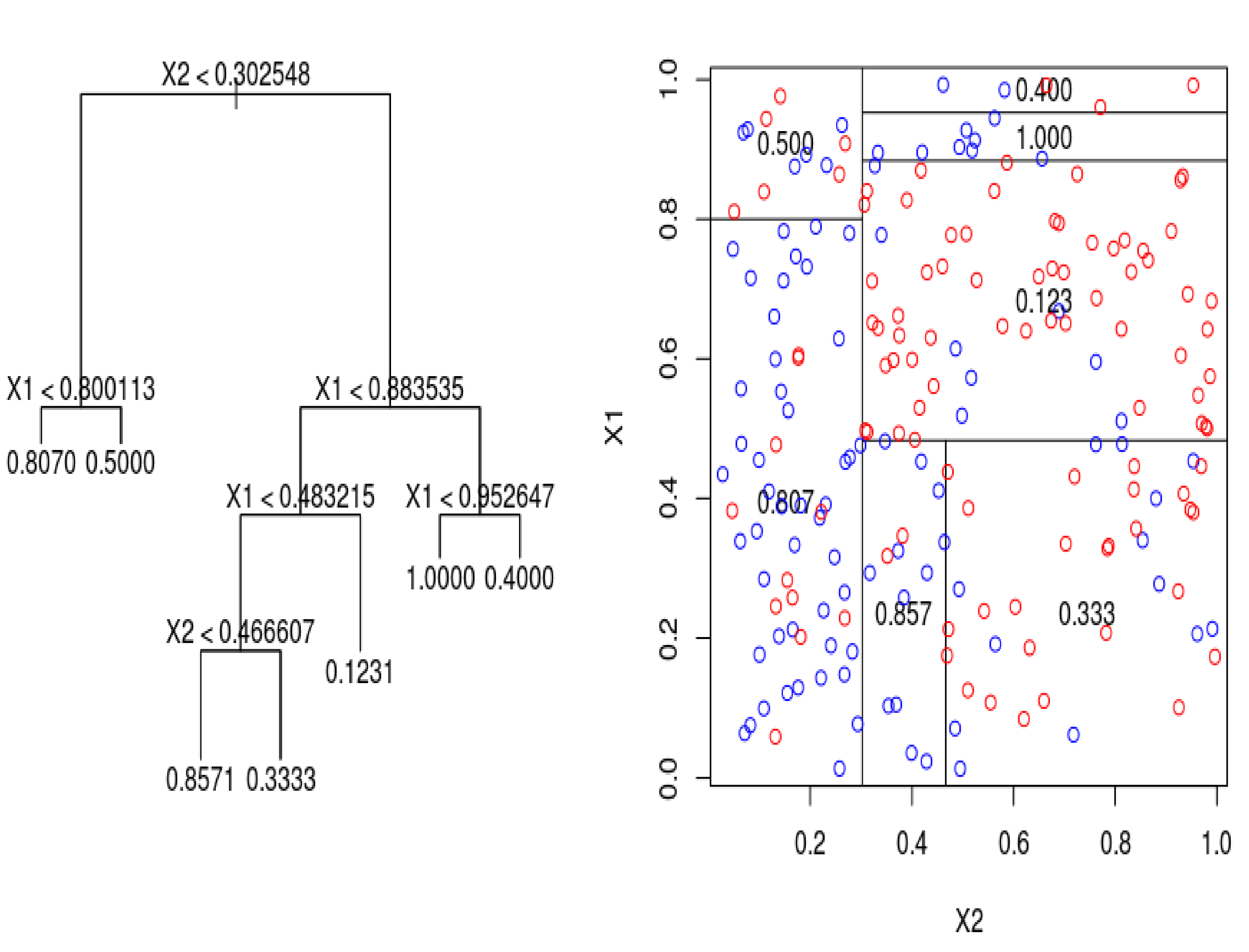
$$Gini\ Index = 1 - \sum_j p_j^2$$

Classification vs Regression Tree

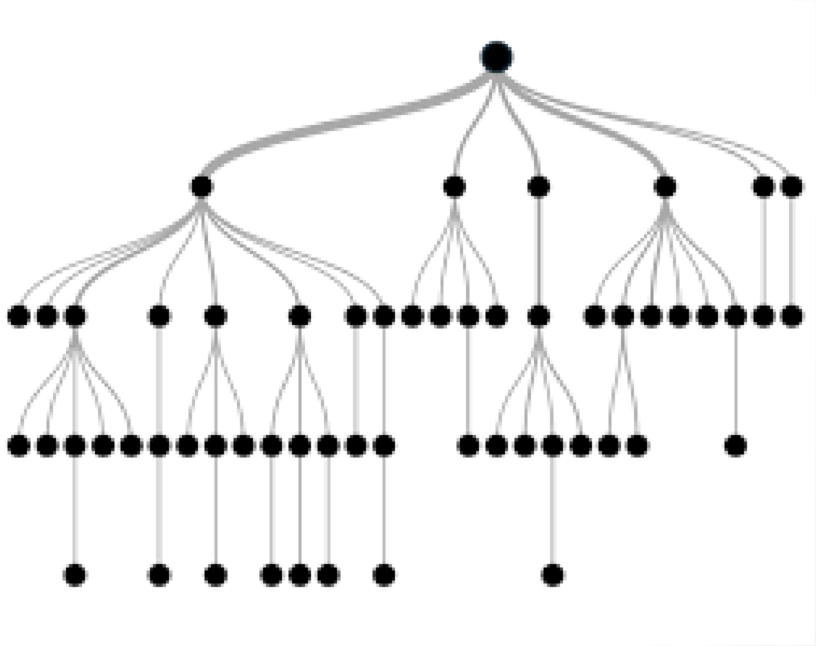




3.0
2.5
2.0
1.5
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
6
7
8
Petal width
Petal length


X2 < 0.302548
X1 < 0.800113
0.8070 0.5000
X1 < 0.883535
X1 < 0.483215
X1 < 0.952647
1.0000 0.4000
X2 < 0.466607
0.1231
0.8571 0.3333
0.500
0.400
1.000
0.123
0.807
0.857
0.333
X1
X2
0.0
0.2
0.4
0.6
0.8
1.0

When to stop splitting ? Overfitting



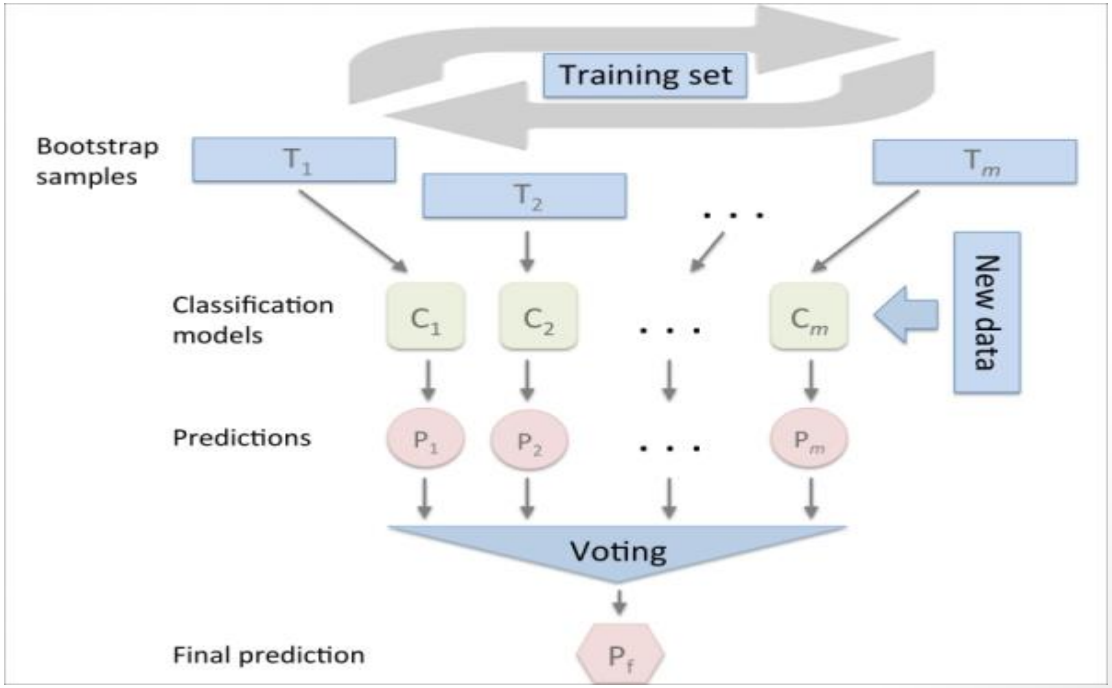
How to overcome ?
Pruning

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

Ensemble

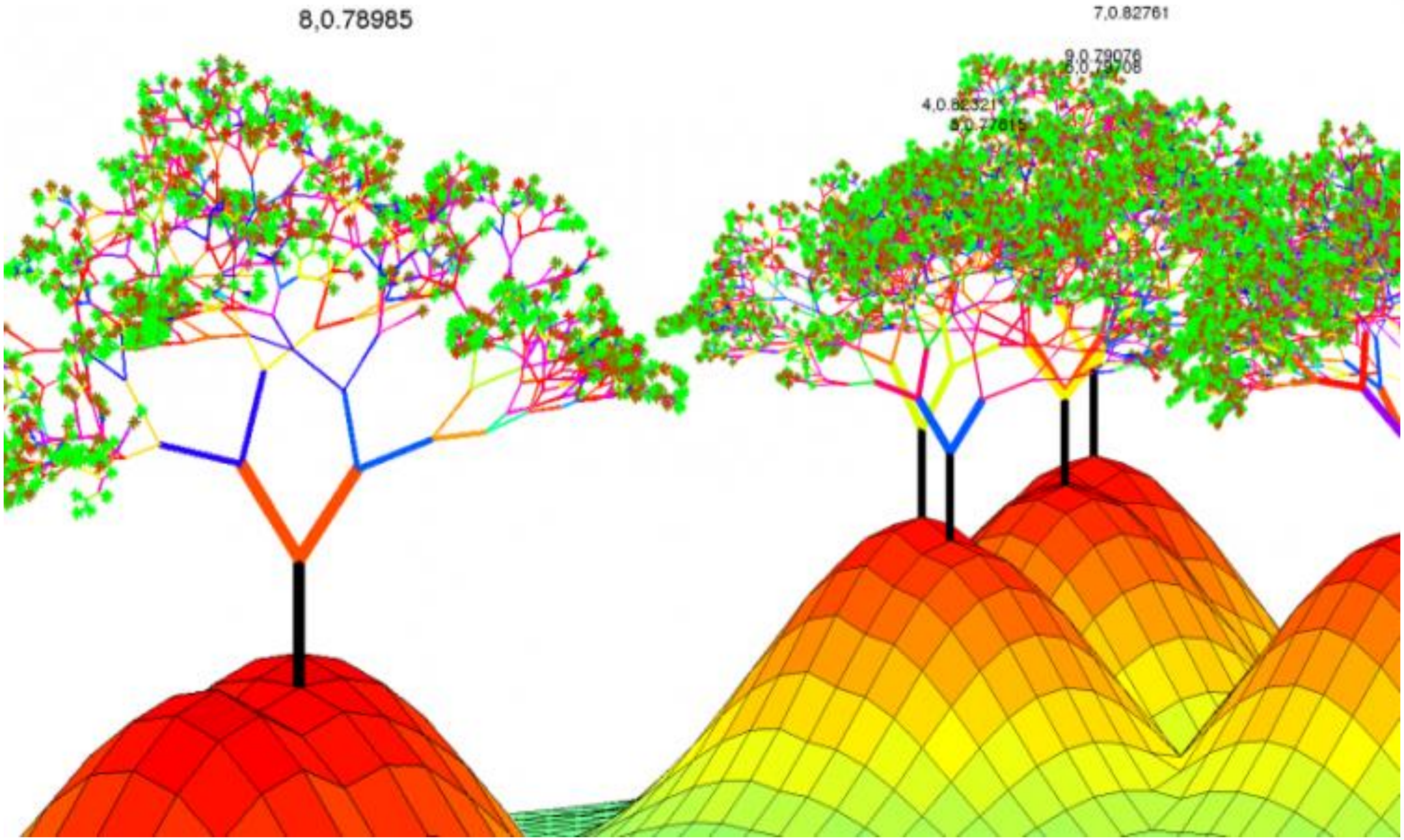
Machine learning paradigm which combine weak learners to become a strong learner

Model1	Model2	Model3	VotingPrediction
1	0	1	1

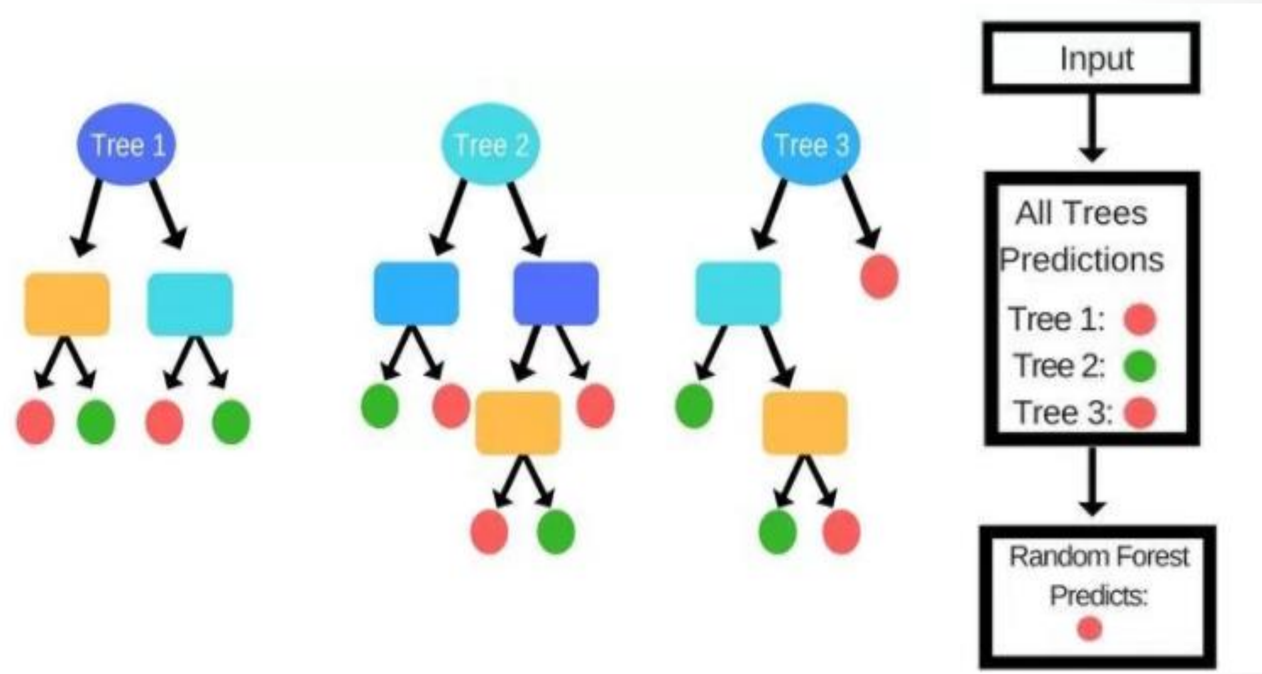


Random Forest

*Most used algorithm - Bagging Technique (***B**ootstrap **agg**regat**ing** - **bagging**)


8,0.78985
7,0.82761
9,0.79076

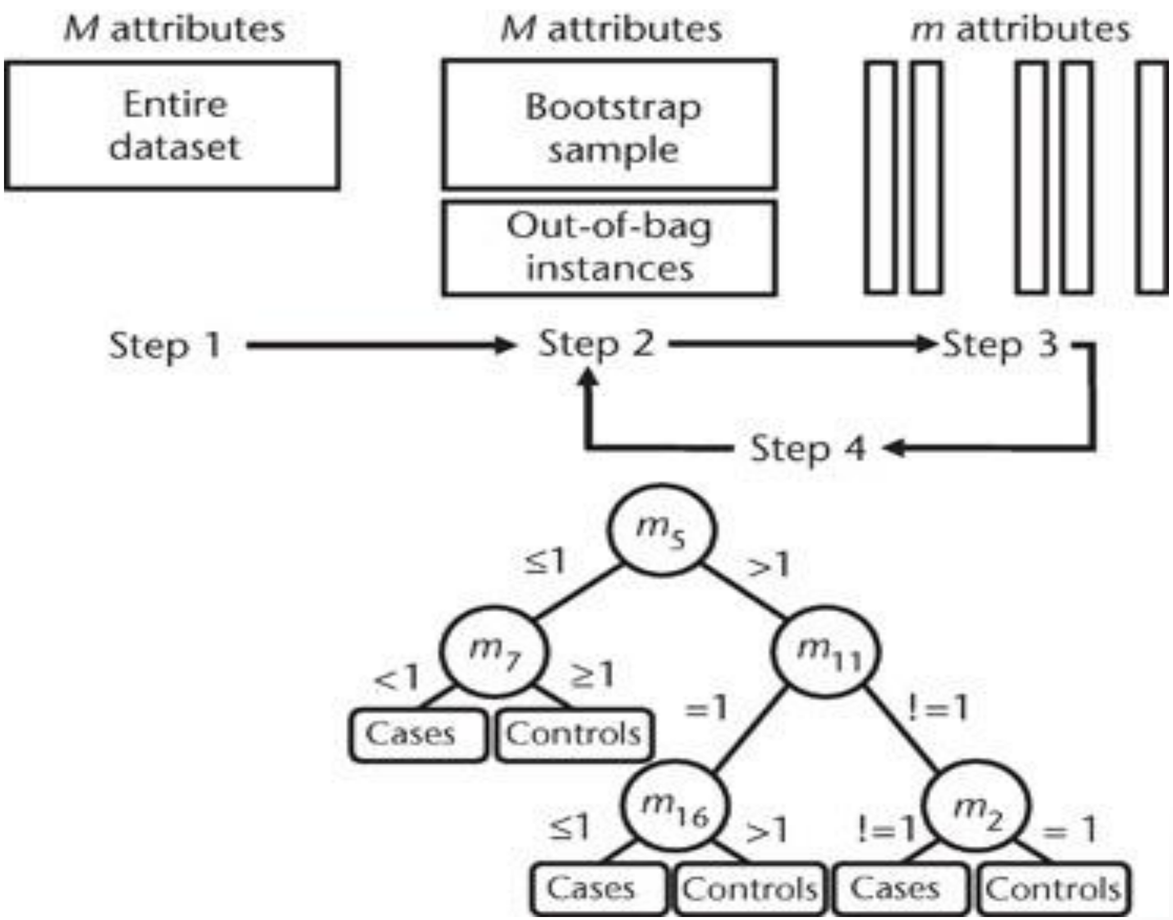
HOW THE RANDOM FOREST ALGORITHM WORKS IN MACHINE LEARNING



- 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
<1
≥1
m_{11}
=1
!=1
Cases
Controls
m_{16}
≤1
>1
m_2
!=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

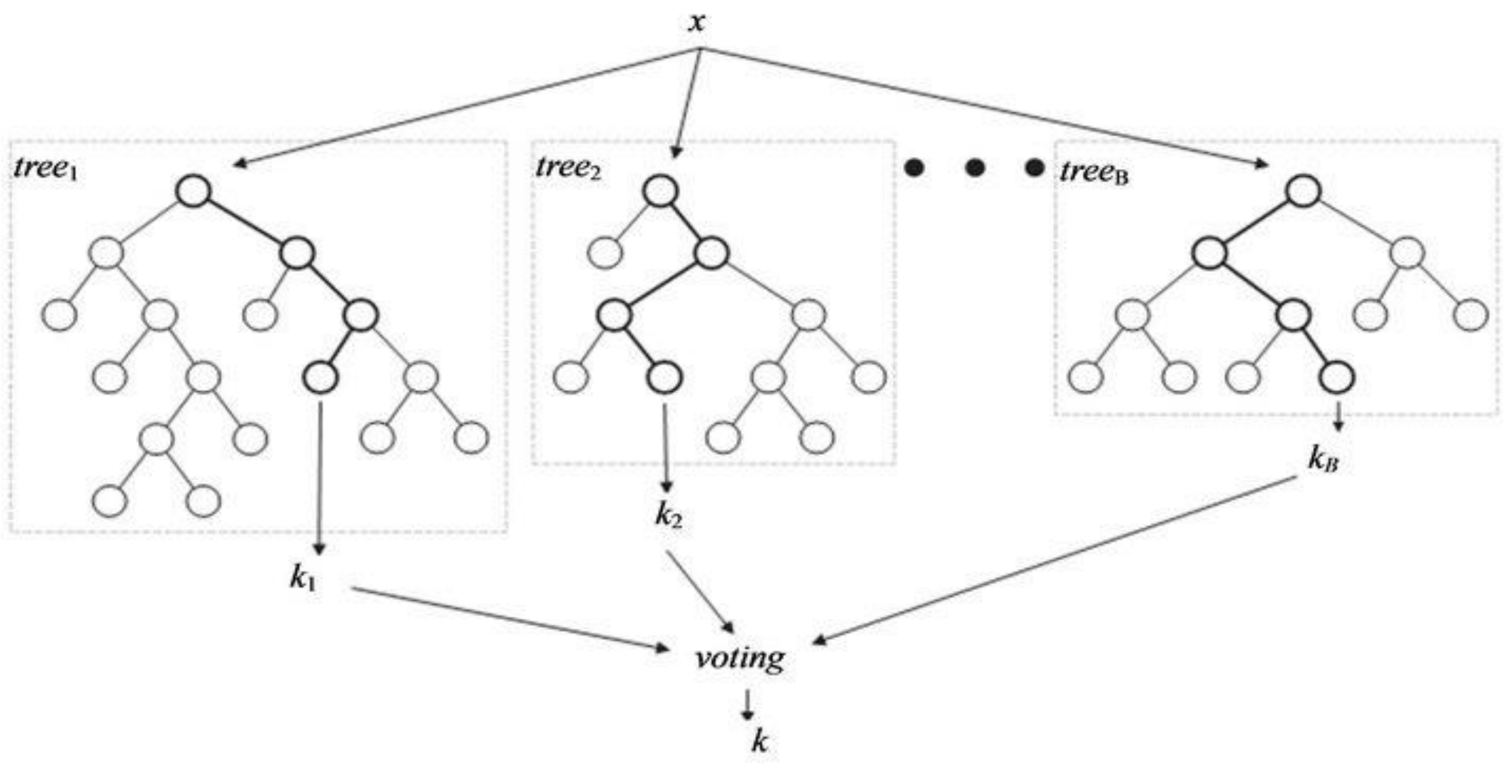
- Among the **“k”** features, calculate the node **“d”**.
- Split the node into **daughter nodes**.
- Repeat **1 to 3** steps
- Build forest by repeating steps **1 to 4** for “n” number times to create **“n” number of trees**.

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



K – Means

Un-Supervised learning algorithm

Clustering

No dependant variable

Pseudocode

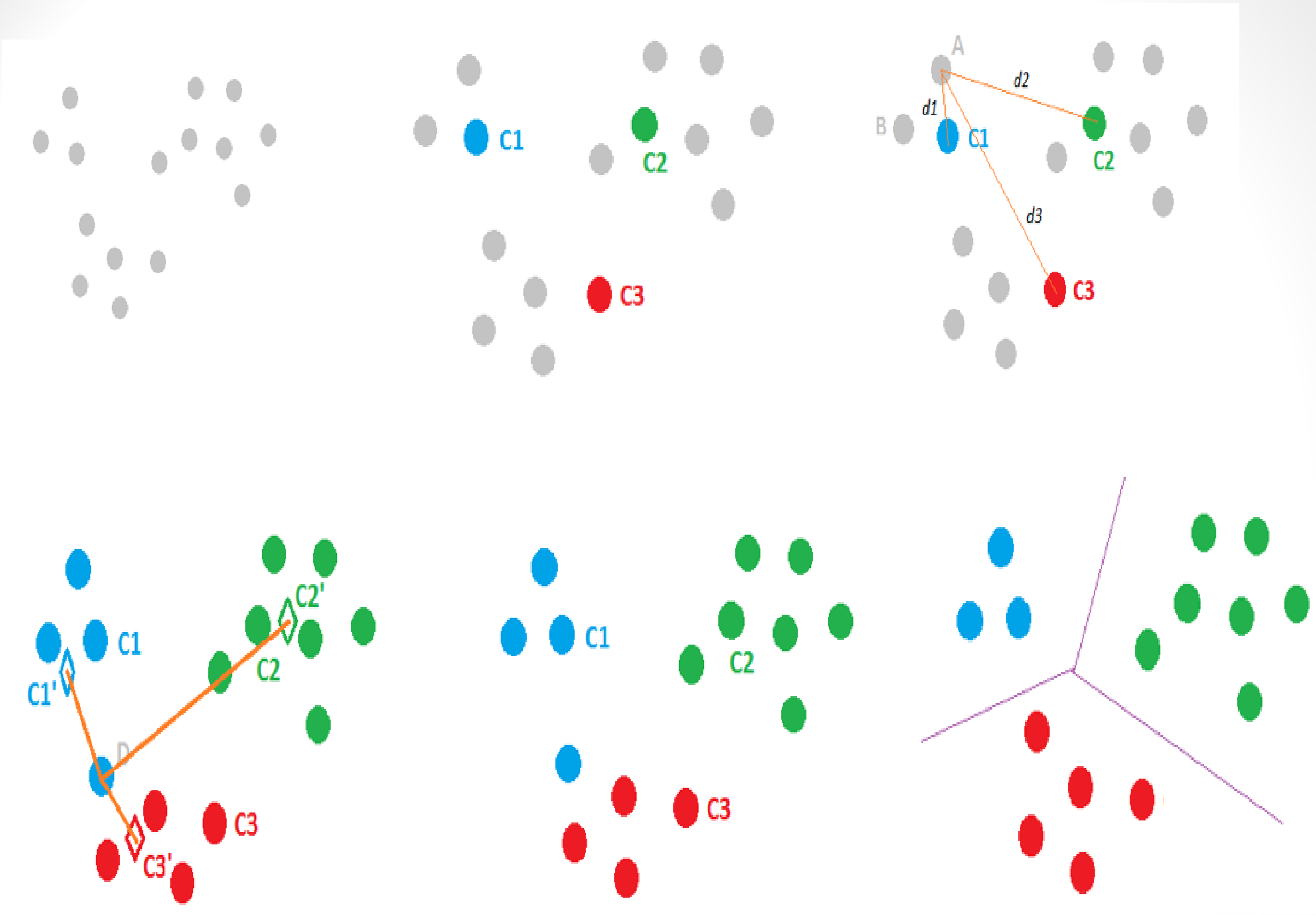
- Input the algorithm with the number of clusters **K** and the data set.
- Randomly generate or randomly select K centroids from the data set.

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
B
d1
d2
d3
C1
C2
C3
C1'
C2'
C3'
C1
C2
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.

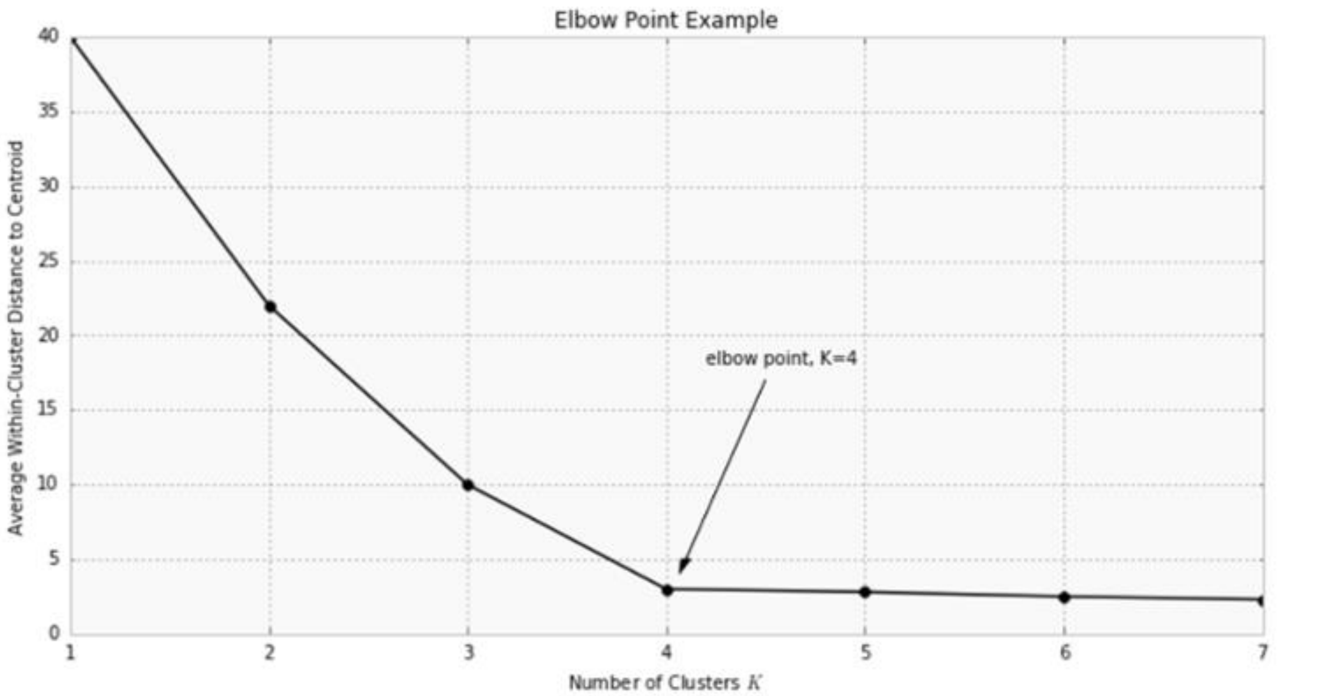
$$c_i = \frac{1}{|S_i|}\sum_{x_i \in S_i} x_i$$

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

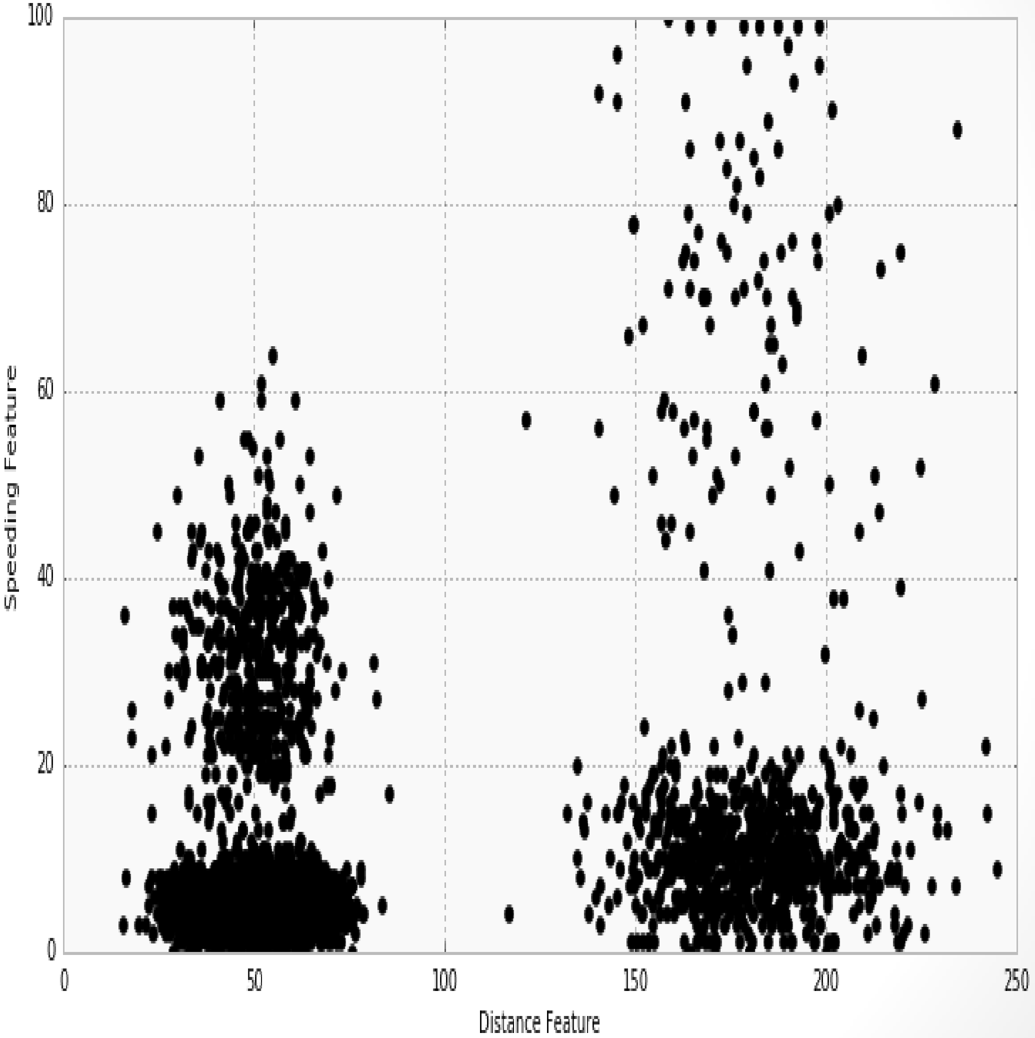
Choosing *K* – K Means ++

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

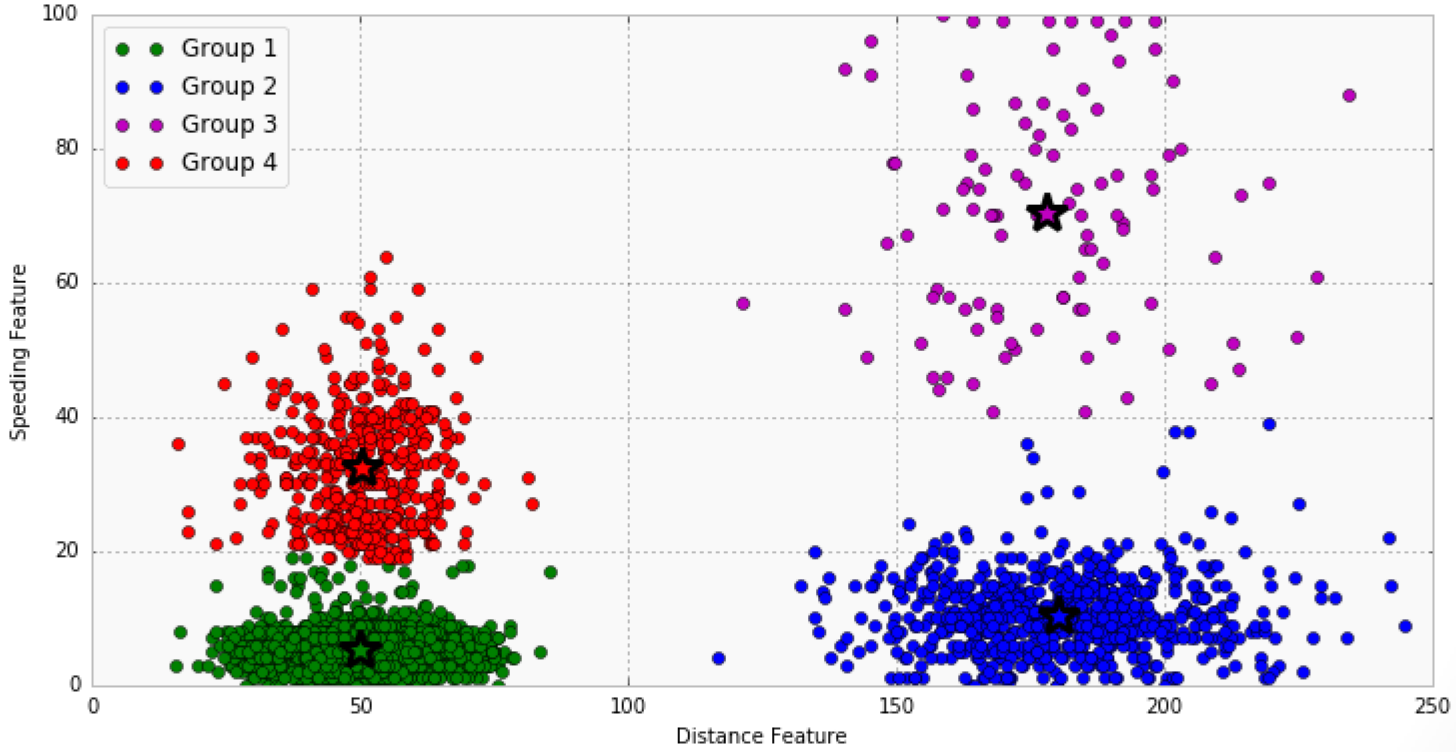


Distance and Speed

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

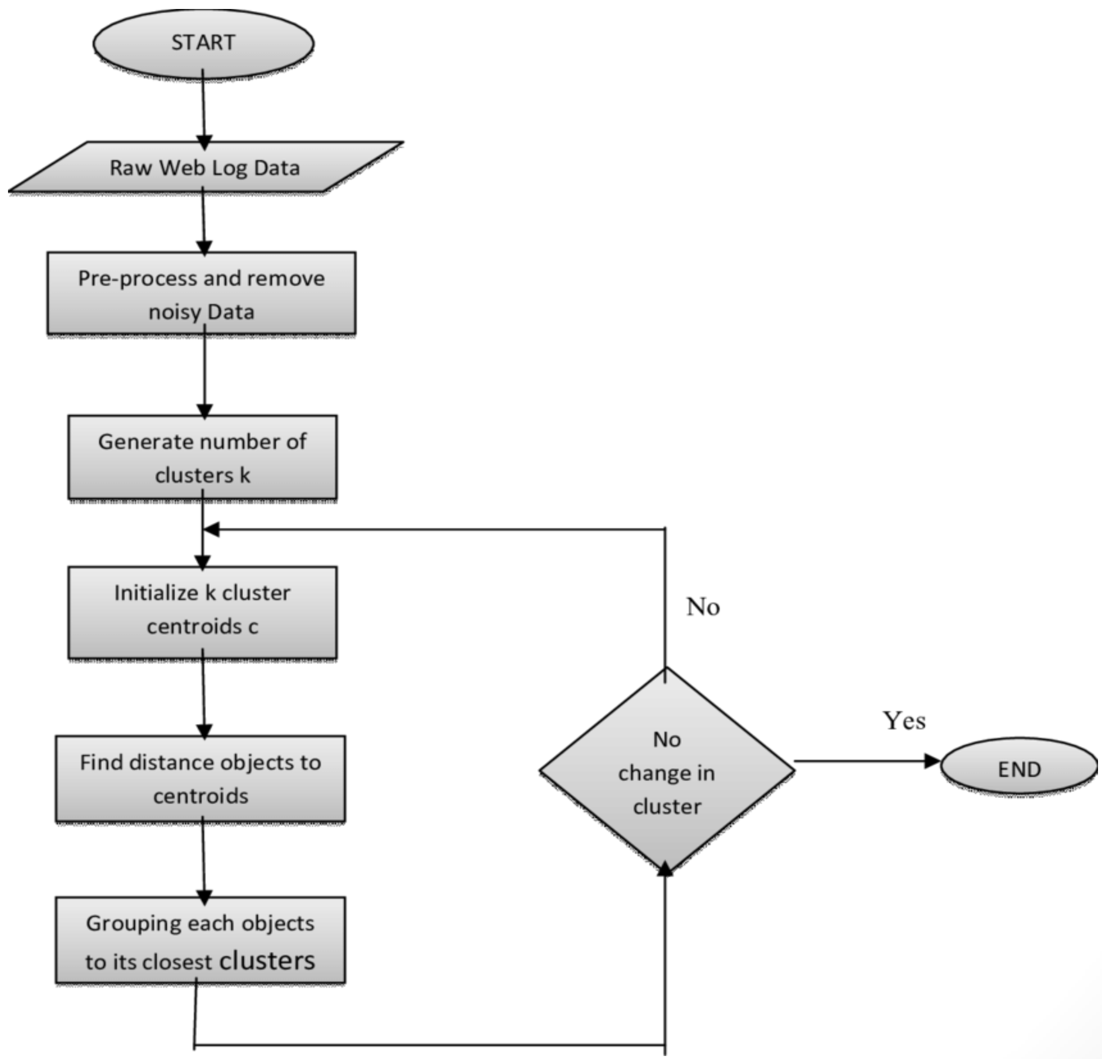


Graph



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)