# DECISION TREE

# DECISION TREE

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

**You are building a  recommendation system and you are suppose to provide recommendations of suggesting App based on Gender and occupation**

**Which one you would suggest for the following people ?**

# 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:** 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

**That was pretty Easy right …**

That was pretty Easy right …

But what if we had to choose between Gender and Occupation to suggest an App

# 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 | |

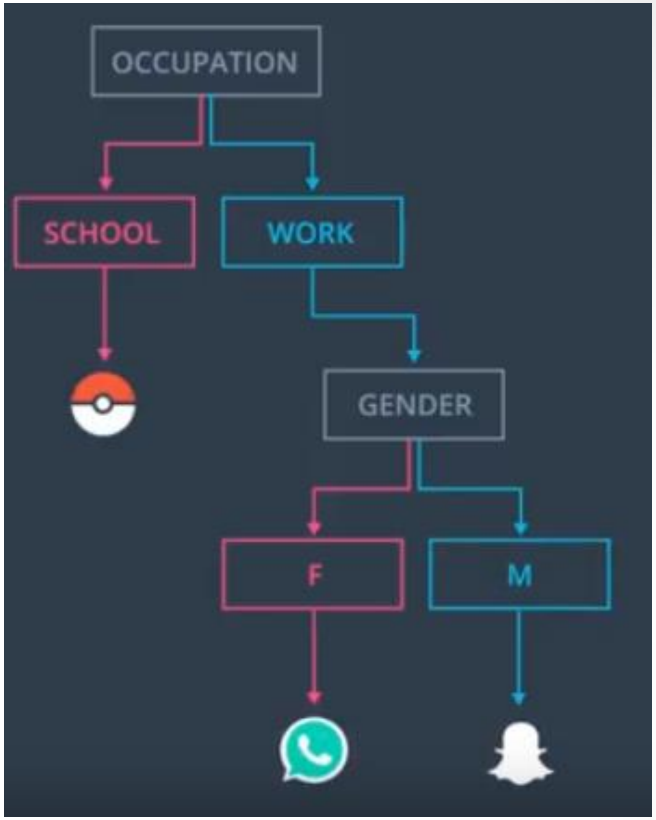
| Gender | Occupation | App |
|---|---|---|
| F | Study | |
| F | Work | |
| M | Work | |
| F | Work | |
| M | Study | |
| M | Study | |

**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 | |



# Terminologies

**Root Node**

**Decision node**

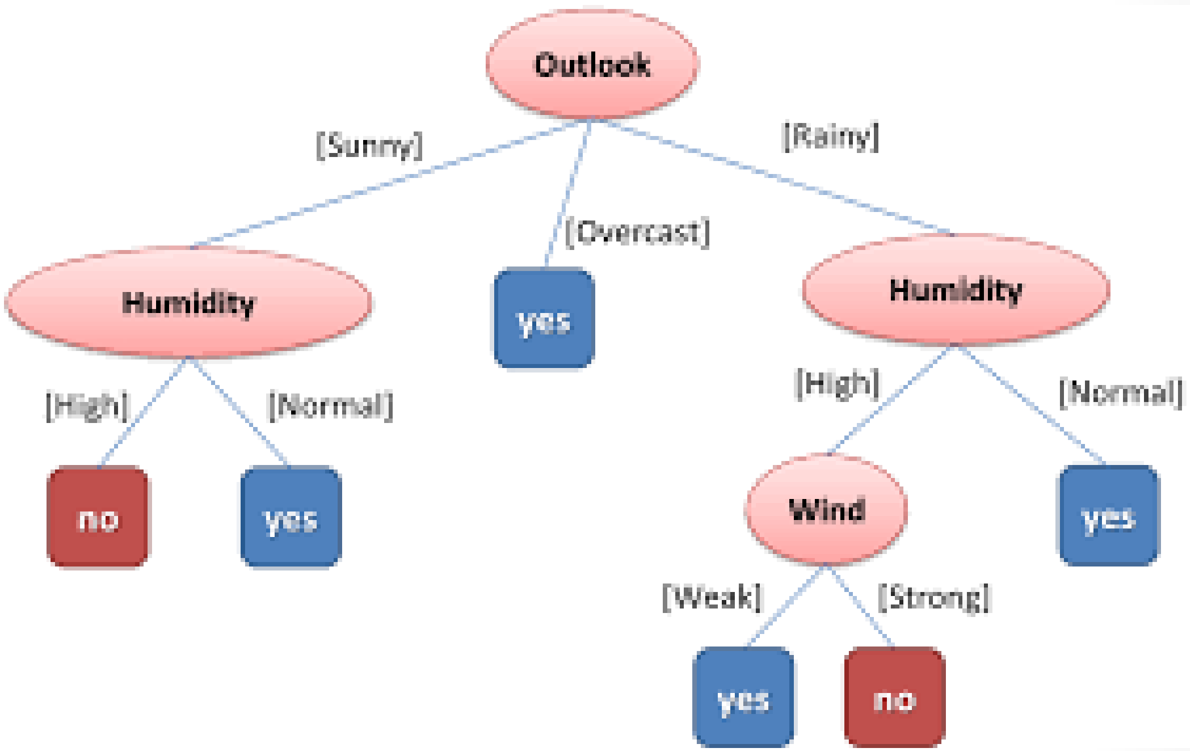
**Leaves**

# Supervised learning algorithm

**Root Node -**

**Decision node -**

**Leaves -**



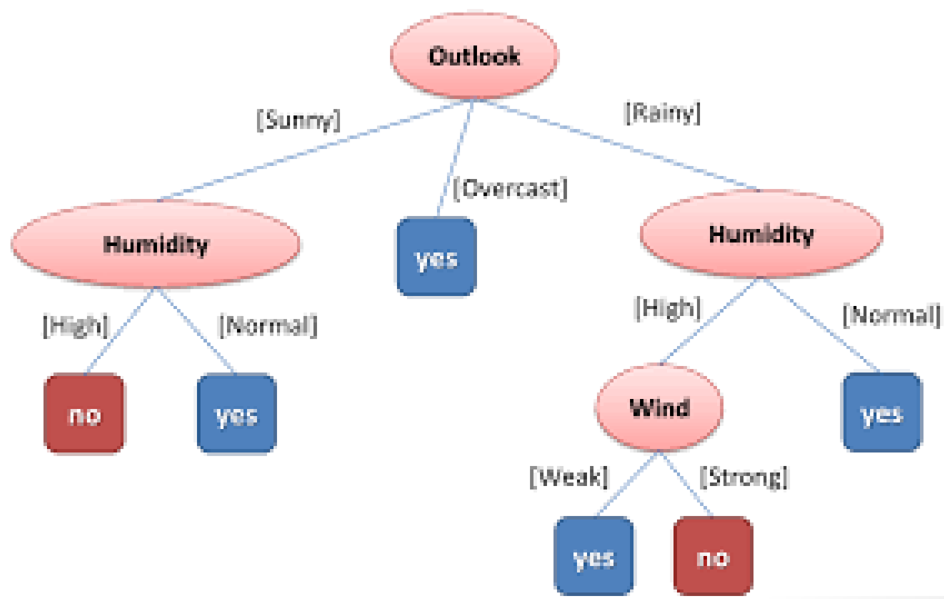
# Supervised learning algorithm

**Root Node** - Outlook

**Decision node** - Humidity/Wind

**Leaves** - Yes/No

Structure of a Tree



# How do we find the Root node ?

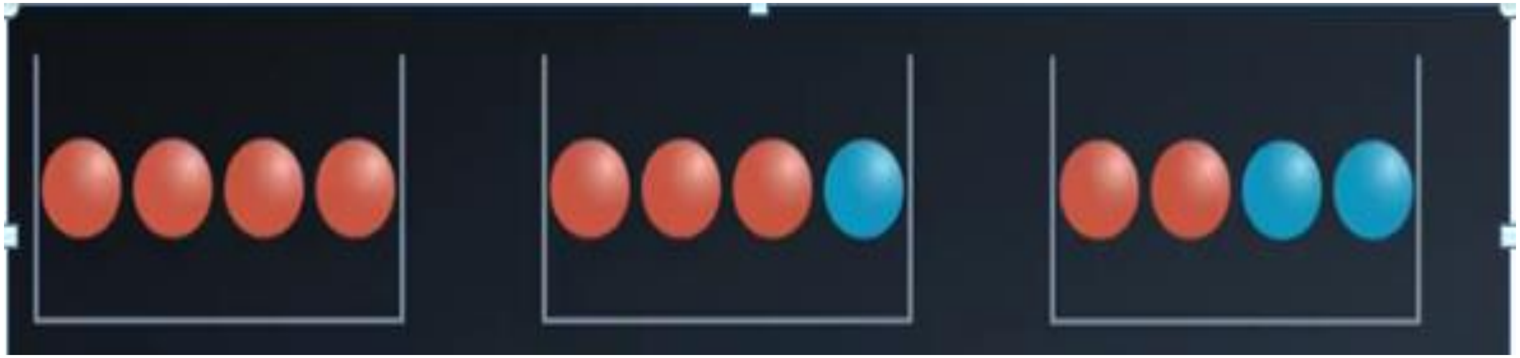
# HOW TO FIND ROOT (2 WAYS)

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

**Understand Entropy you will get to understand Information Gain**

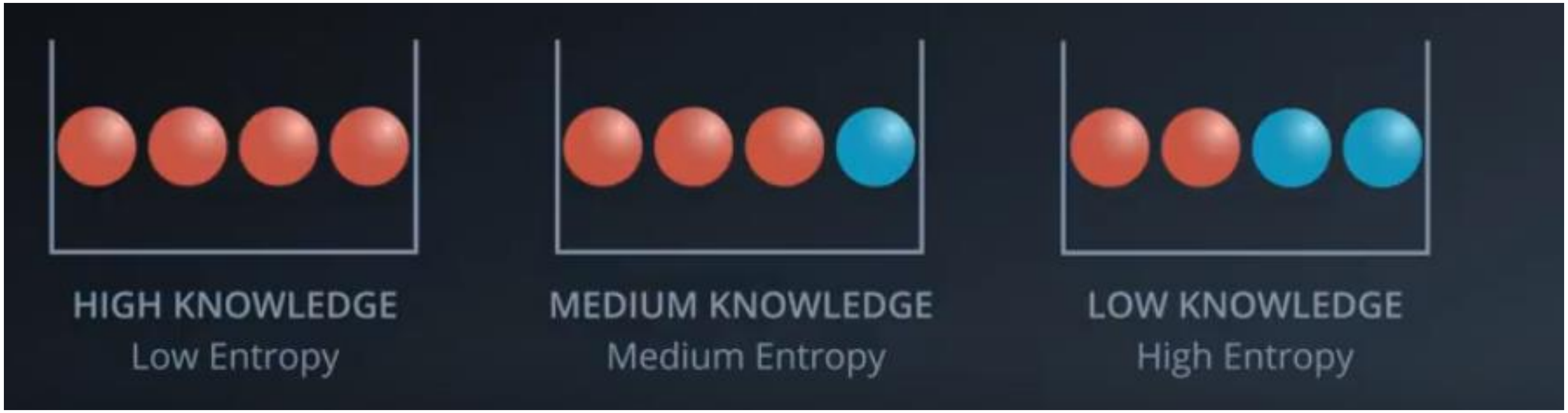
# Entropy or Randomness

- The measure of uncertainty



# Entropy - The measure of uncertainty

# Entropy - The measure of uncertainty




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

# 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**

# 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)

= 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

**G**(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

# Information Gain - All Attributes

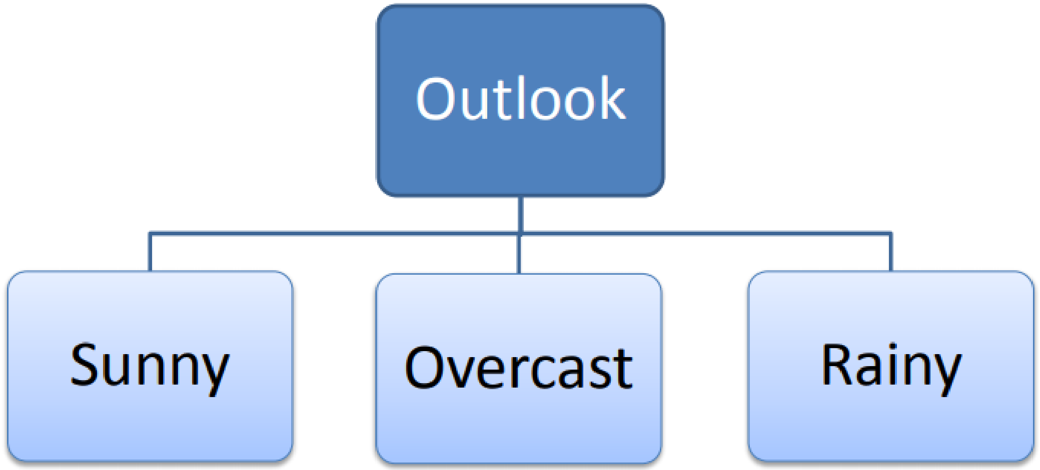
| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Outlook | Sunny | 3 | 2 |
| | Overcast | 4 | 0 |
| | Rainy | 2 | 3 |
| Gain = 0.247 | | | |

| | | 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 | | | |

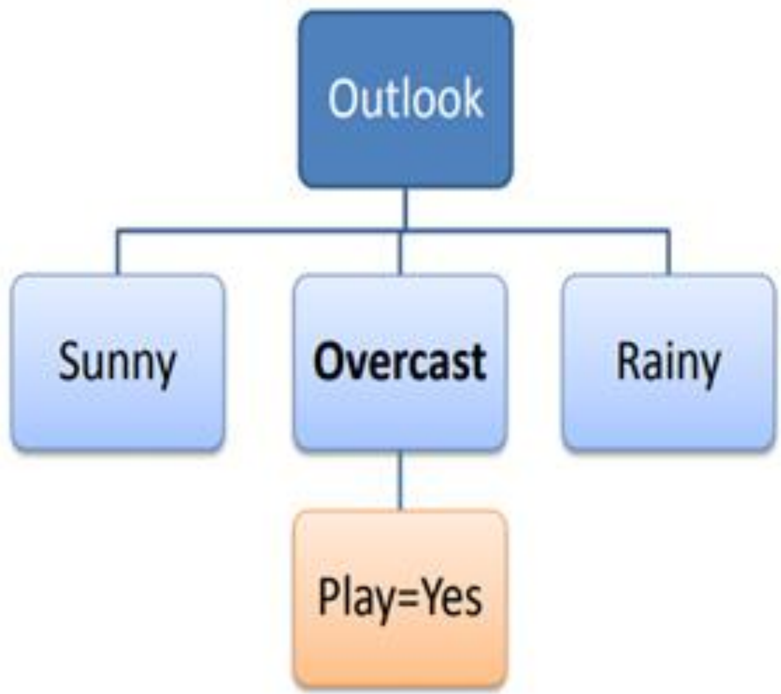
# Construction of Tree



| 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 |

| 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 |

# 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 |

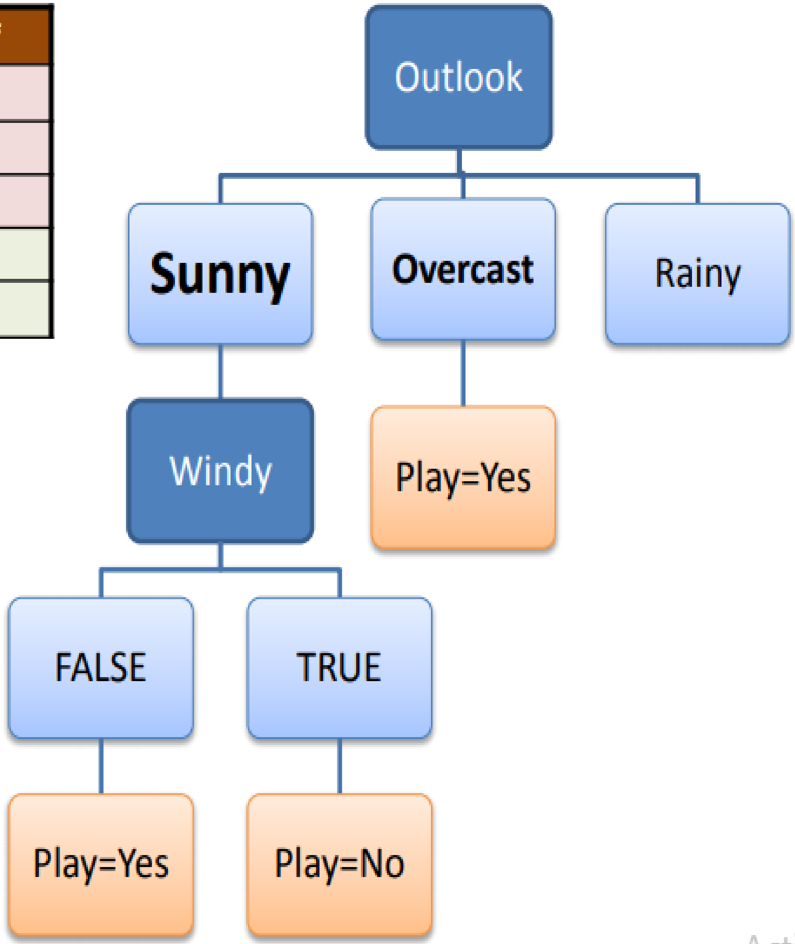
| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Temp. | Mild | 2 | 1 |
| | Cool | 1 | 1 |
| Gain = 0.02 | | | |

| | | 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

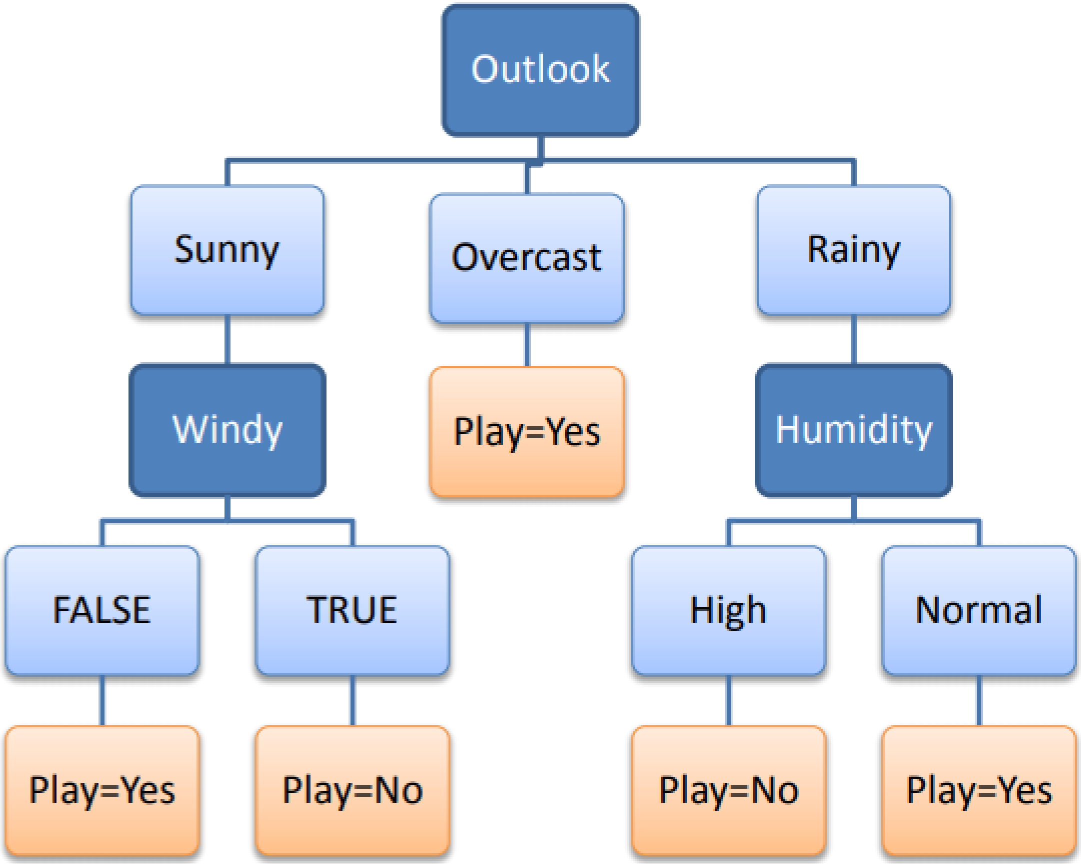
| Temp. | Humidity | Windy | Play Golf |
|---|---|---|---|
| Hot | High | FALSE | No |
| Hot | High | TRUE | No |
| Mild | High | FALSE | No |
| Cool | Normal | FALSE | Yes |
| Mild | Normal | TRUE | Yes |

| | | 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 | | | |

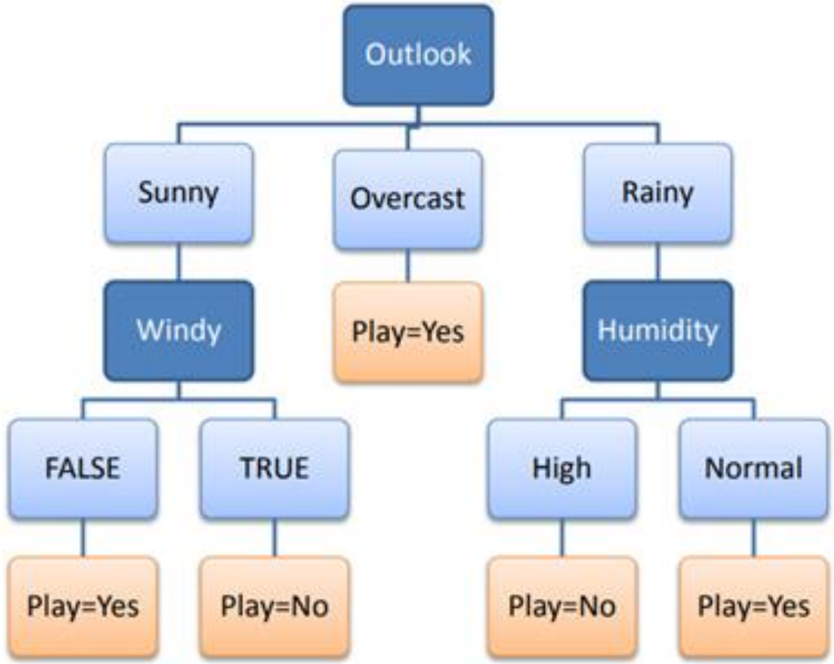
| | | Play Golf | |
|---|---|---|---|
| | | Yes | No |
| Windy | False | 1 | 2 |
| | True | 1 | 1 |
| Gain = 0.02 | | | |

# Final Tree Structure



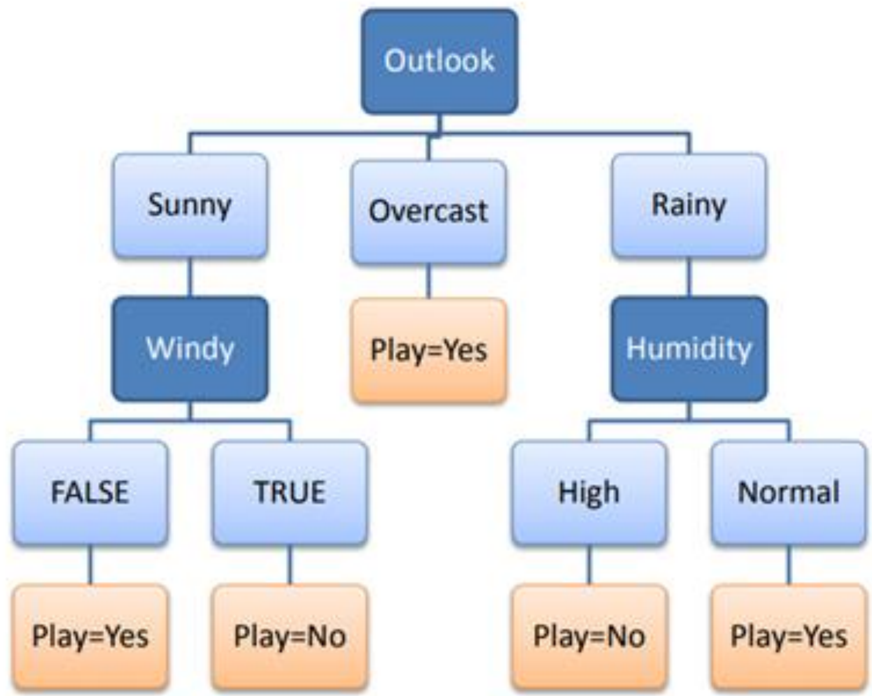
# Predict the Play – D15 ?

| 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

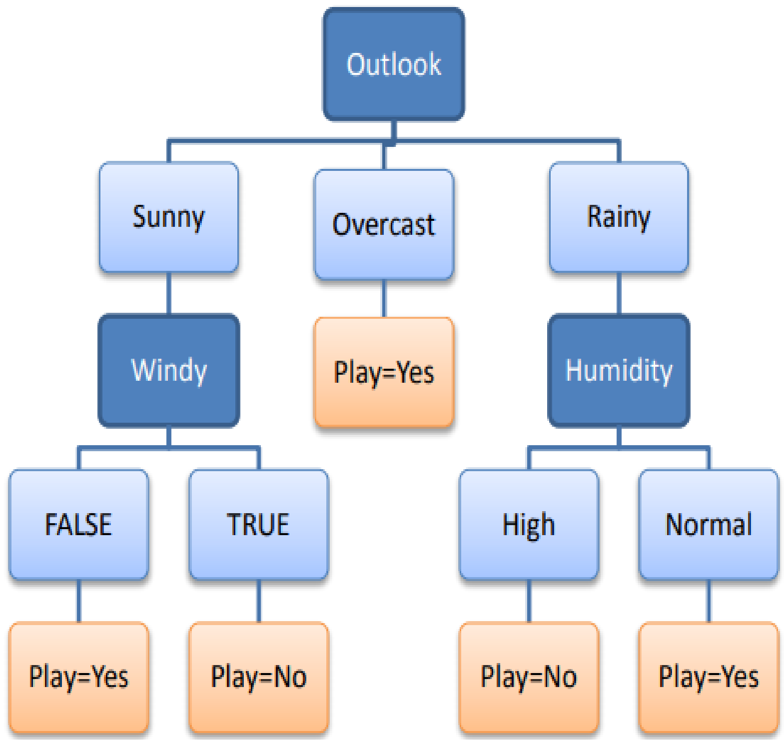
**$R_1$**: IF (Outlook=Sunny) AND (Windy=FALSE) **THEN** Play=Yes

**$R_2$**: 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

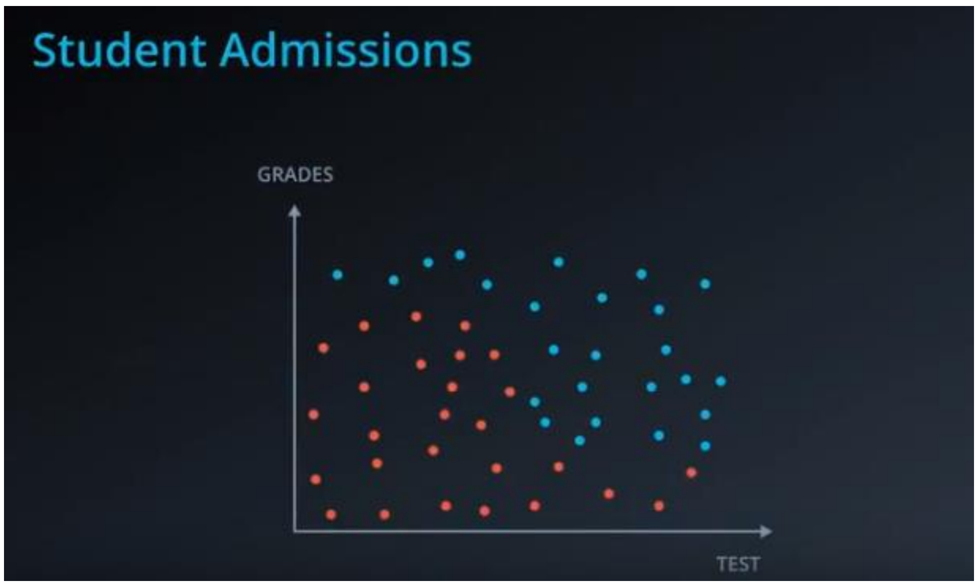


# Finding Root using Gini Index

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

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?

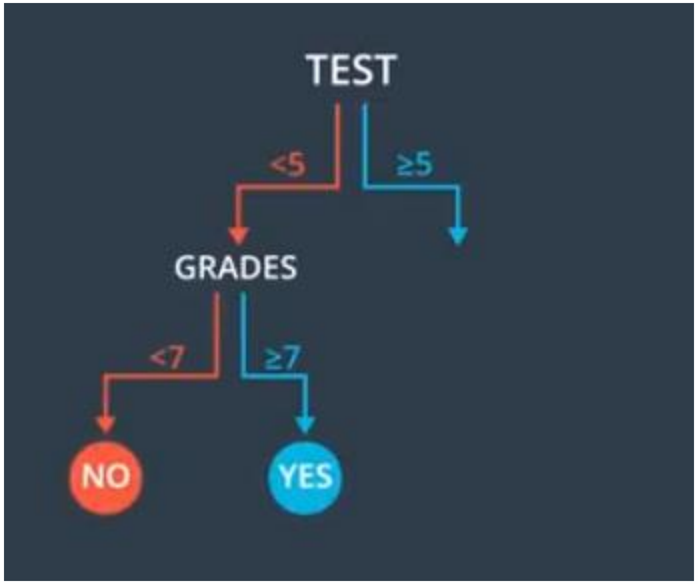
**Or**

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

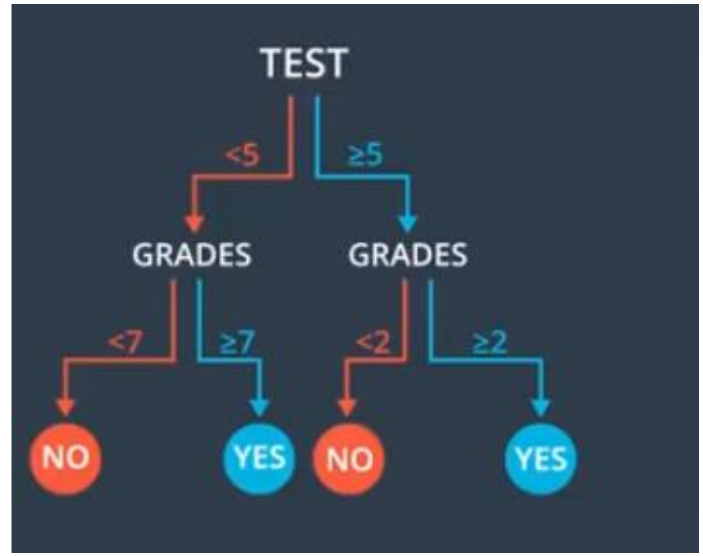
- ○ **Horizontal**
- ○ **Vertical**

# Horizontal vs Vertical


# Construction of a Tree

# Decision Tree – Manual Structure

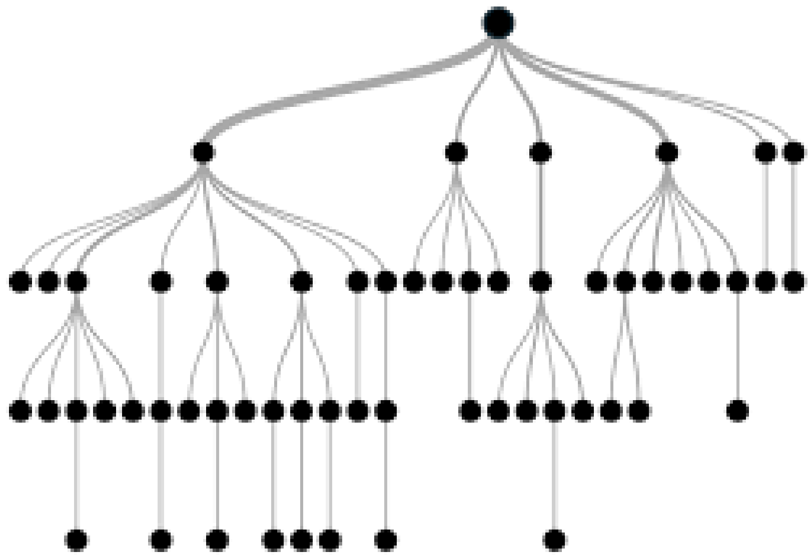
# Problem with Trees

# Problem with Trees

**How will a tree structure look if there are N columns**

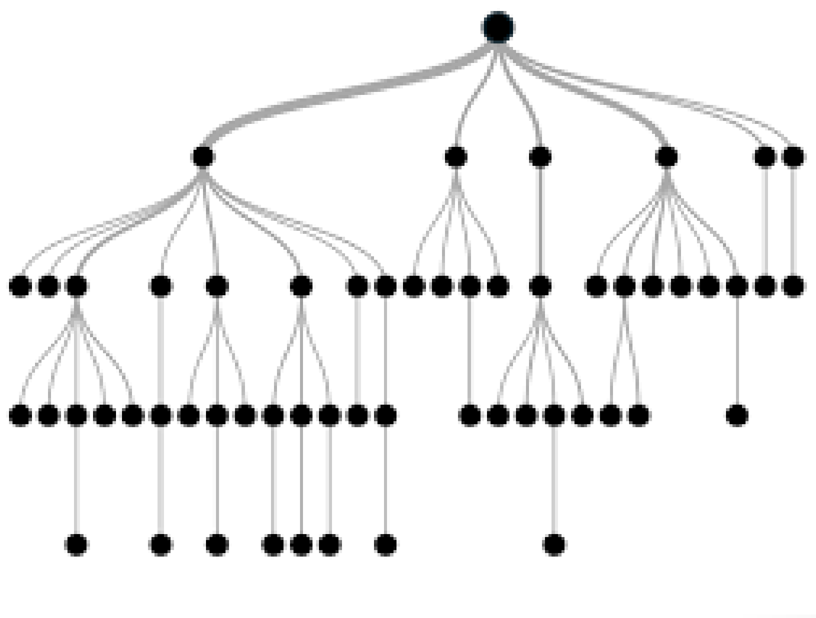
# Problem with Trees

**How will a tree structure look if there are N columns**

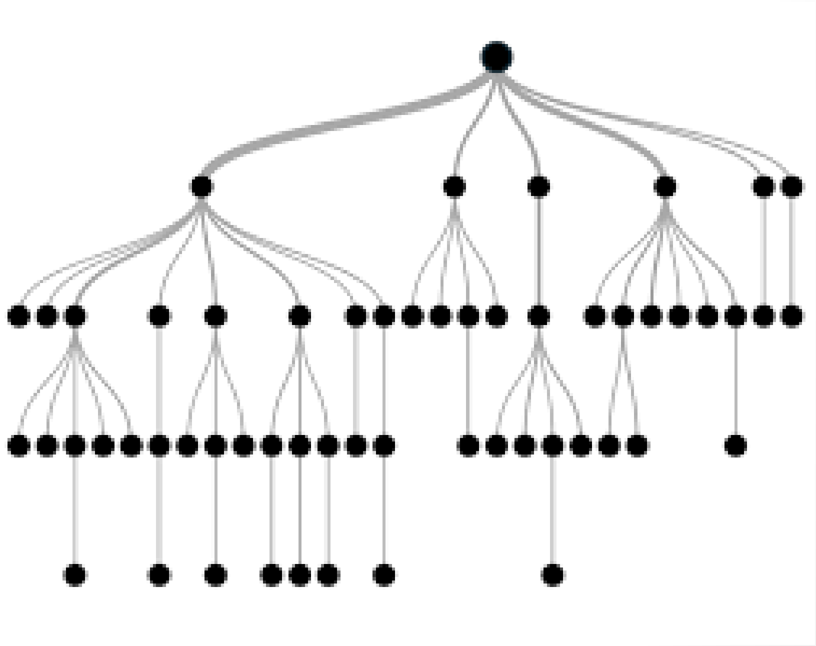


# Problem with Trees - Overfitting

**How will a tree structure look if there are 30 columns**



# When to stop splitting ?



# Pruning -

## To Avoid Overfitting

# Pruning

1. Pre-pruning
2. Post-pruning

# Ensemble

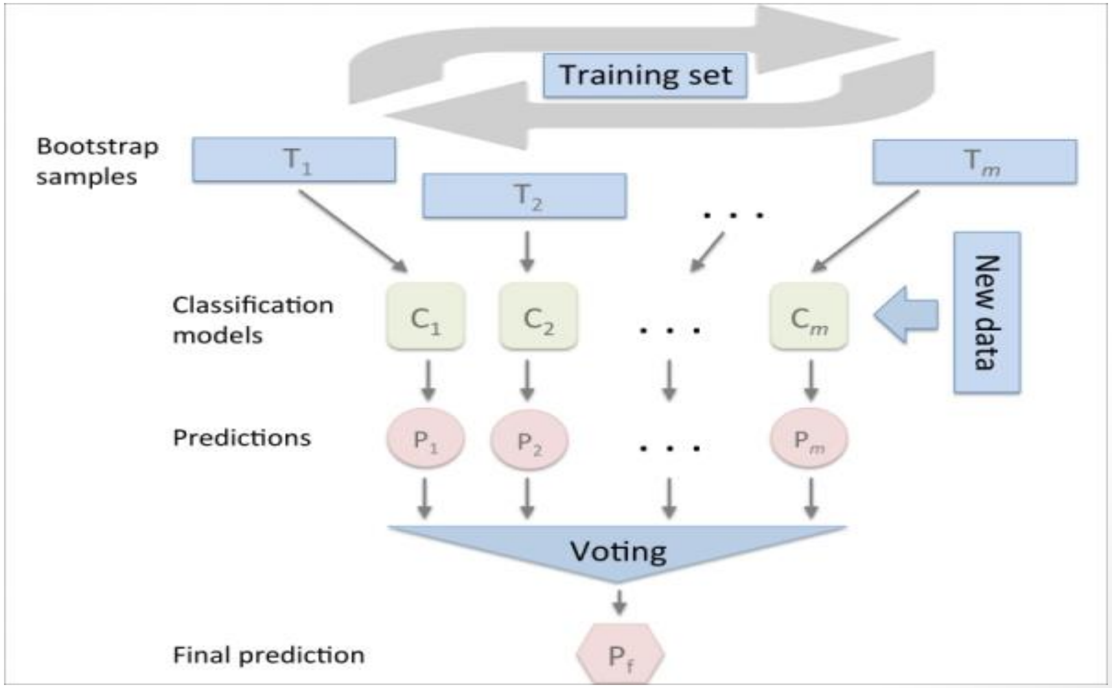
# Ensemble

1. Bagging
2. Boosting

# Ensemble

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

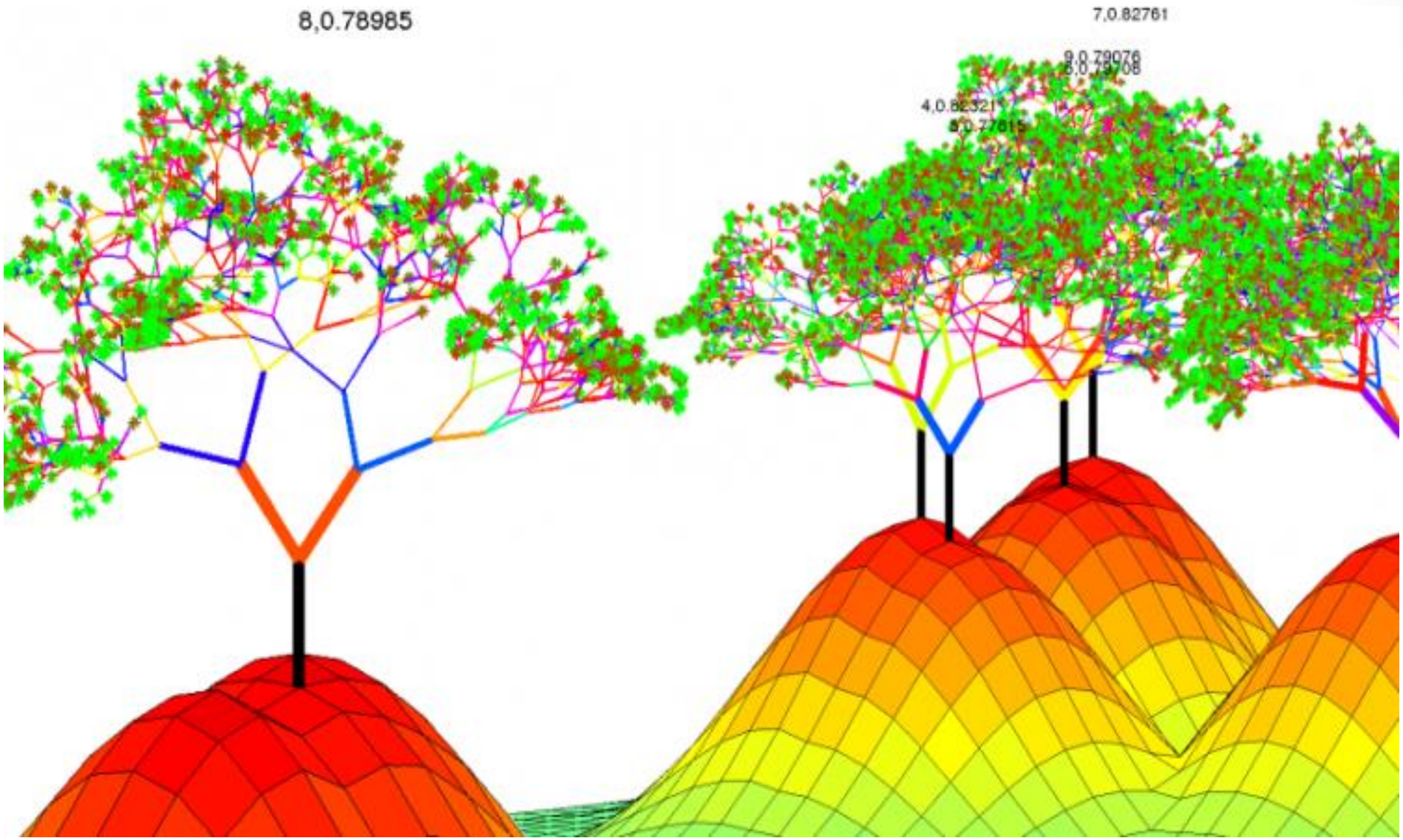
| Model1 | Model2 | Model3 | VotingPrediction |
|---|---|---|---|
| 1 | 0 | 1 | 1 |



# **Random Forest** (*Most used algorithm)*

# **Random Forest** *(Most used algorithm)*

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


8,0.78985
7,0.82761
9,0.79076

# Why Random Forest?

No overfitting

Use of multiple trees reduce the risk of overfitting

Training time is less

High accuracy

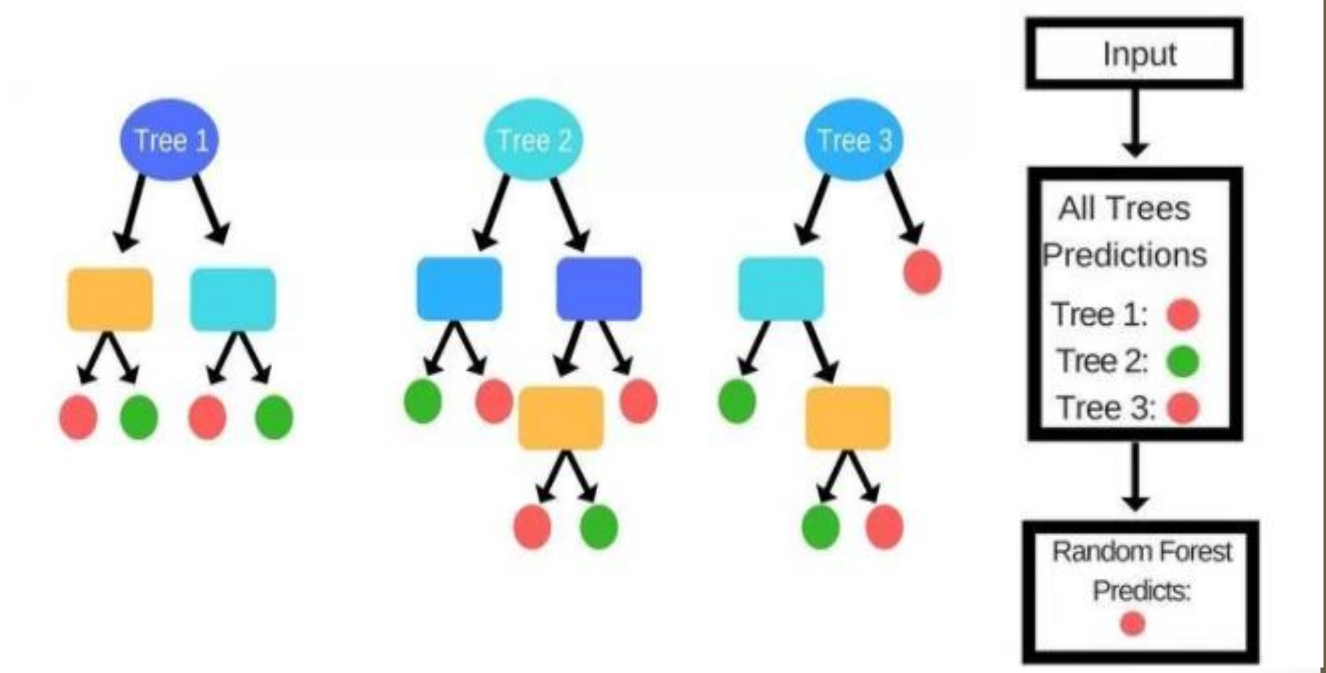
Runs efficiently on large database

For large data, it produces highly accurate predictions

Estimates missing data

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

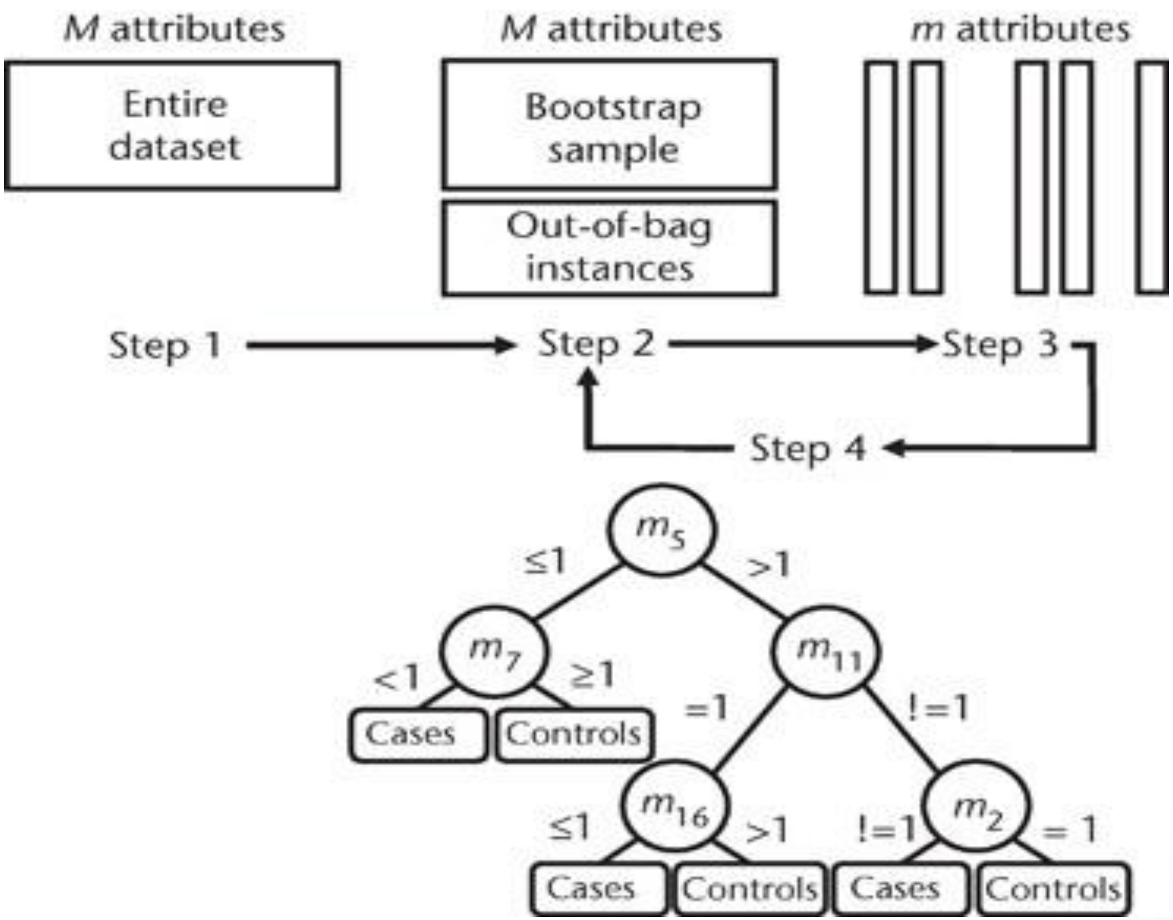
# HOW THE RANDOM FOREST ALGORITHM WORKS IN MACHINE LEARNING




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

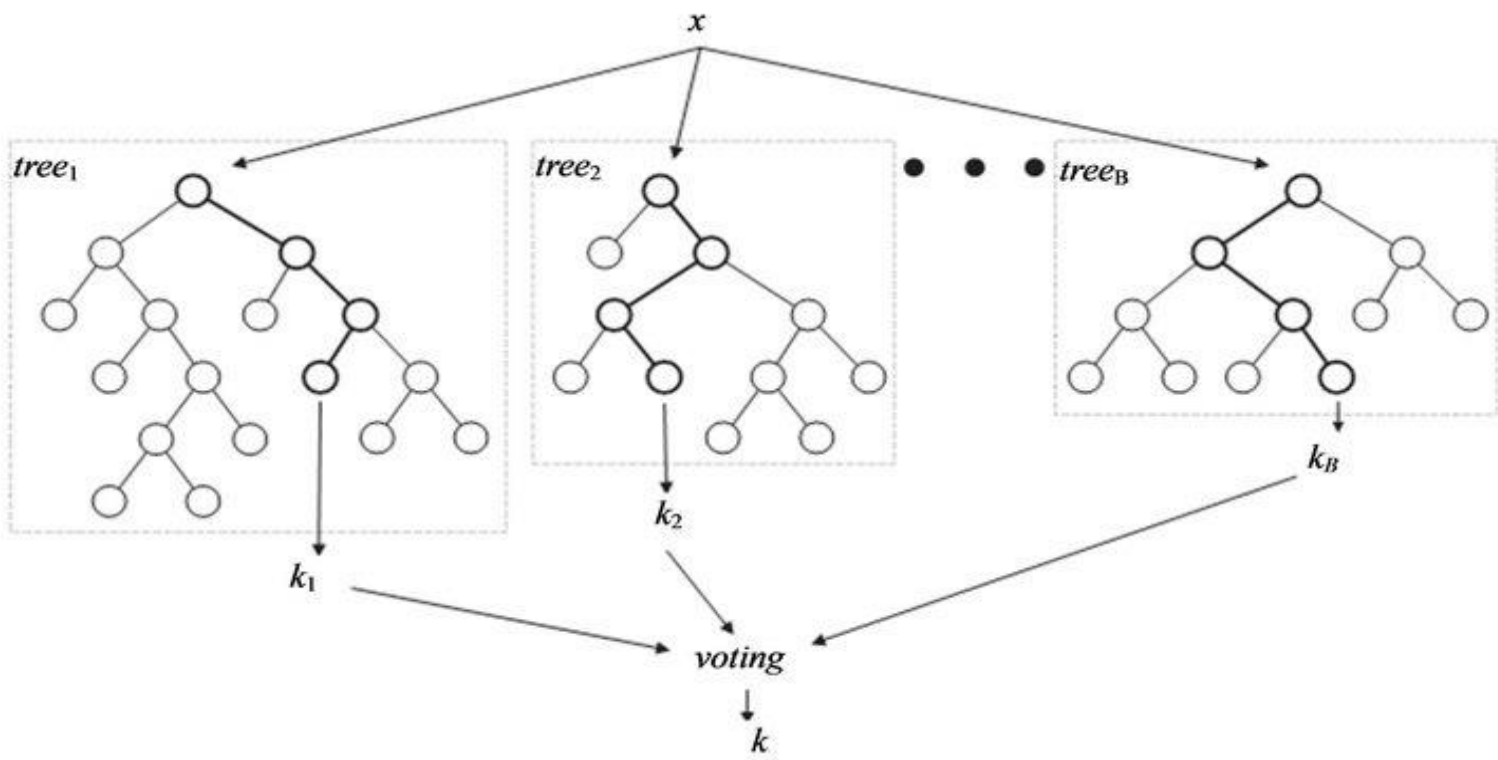
- Among the **“k”** features, calculate the node **“d”**.
- Split the node into **daughter nodes**.
- Repeat **1 to 3** steps
- Build forest by repeating steps **1 to 4** for “n” number times to create **“n” number of trees**.

# Key Points

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

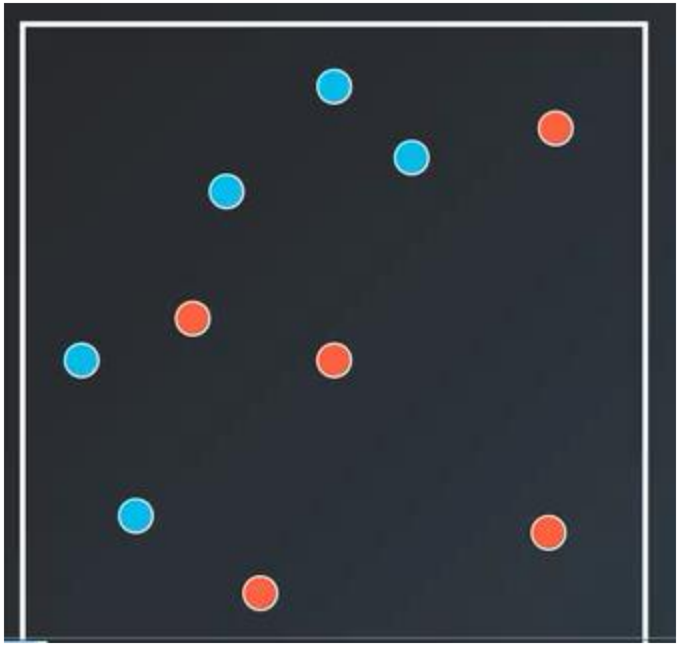
- When we have more trees in the forest, random forest classifier won’t **overfit** the model.
- For each bootstrap sample taken from the training data, there will be samples left behind that were not included. These samples are called **Out-Of-Bag samples** or OOB.
- The performance of each model on 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

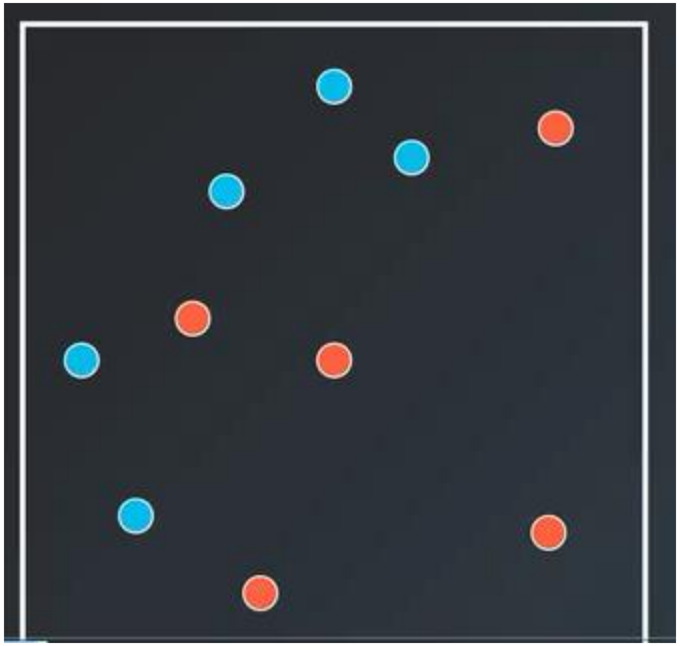


# Boosting

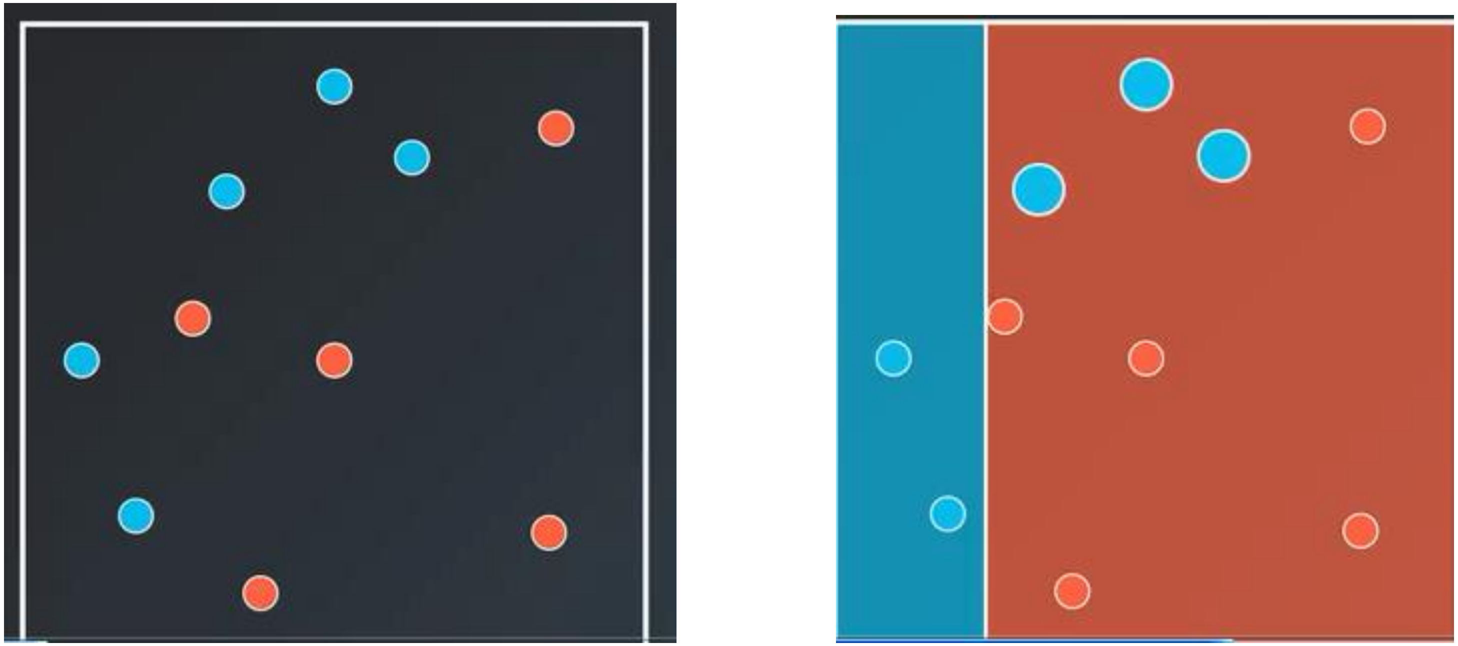
# AdaBoost (**Ada**ptive **Boost**ing)



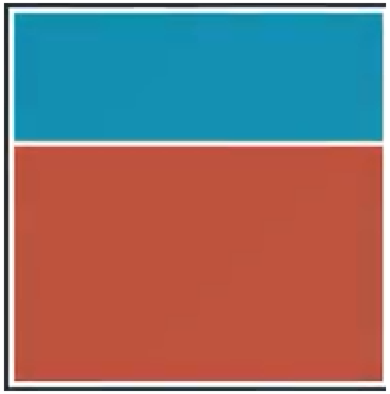
# AdaBoost – Pattern 1



# AdaBoost – Pattern 1



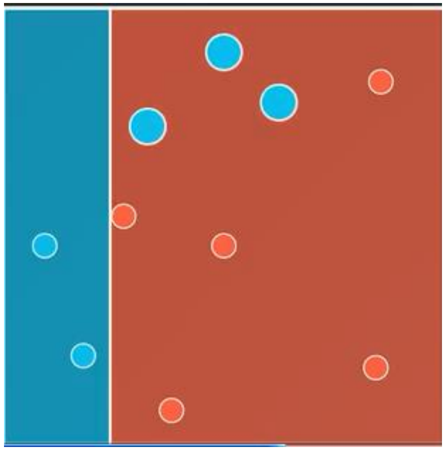
# AdaBoost – Pattern 2

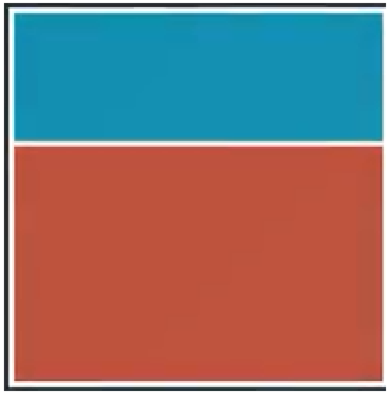


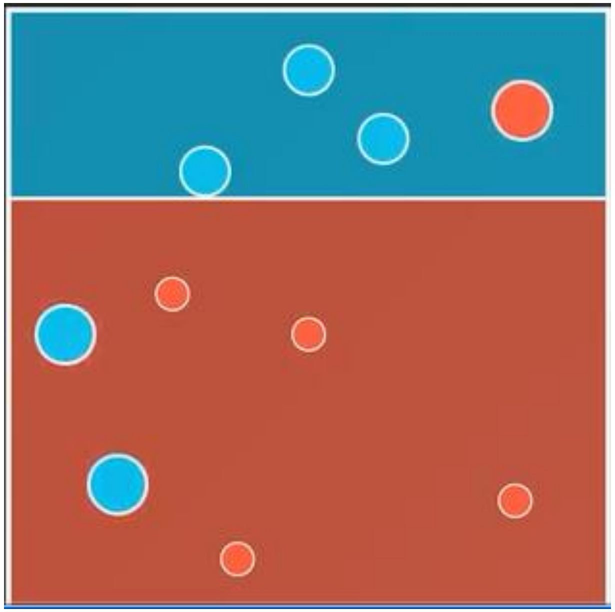
**Apply pattern 2  on the Input Data from pattern 1**

# AdaBoost – Pattern 2

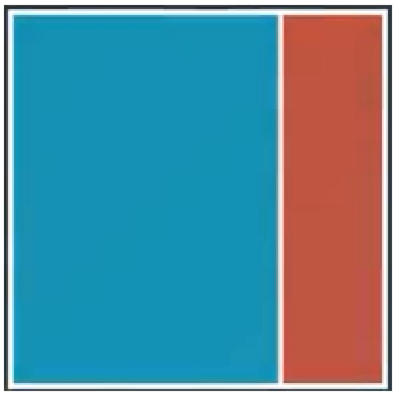
**Input Data**







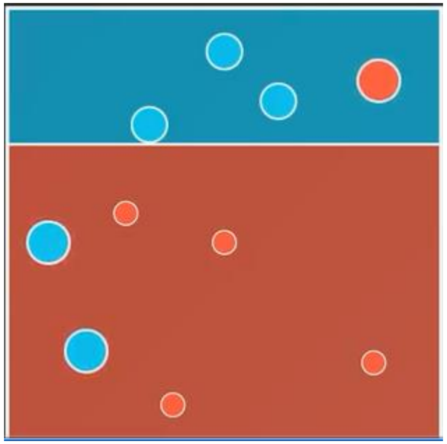
# AdaBoost – Pattern 3

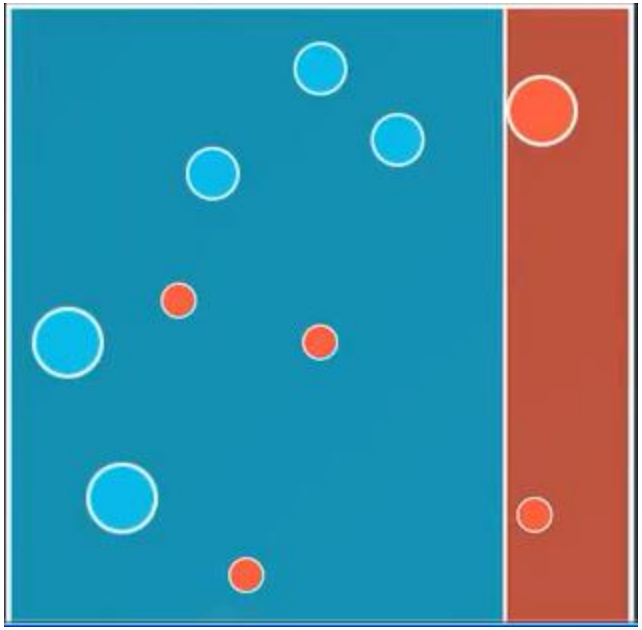


**Apply pattern 3  on the Input Data from pattern 2**

# AdaBoost – Pattern 3

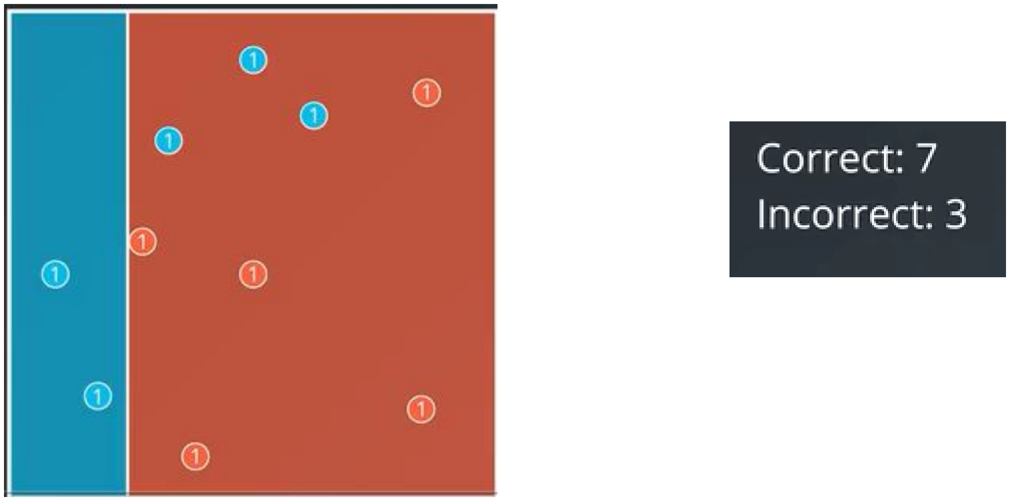
**Input Data**

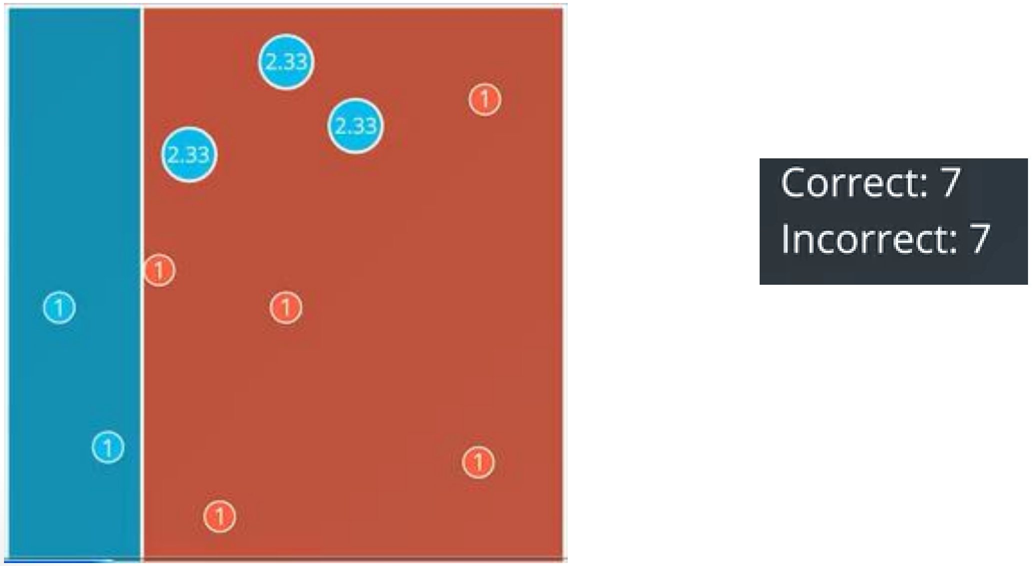




# AdaBoost – Pattern 1

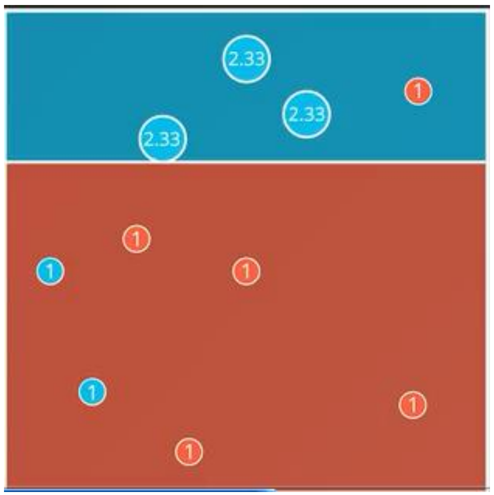
Weights after applying pattern 1



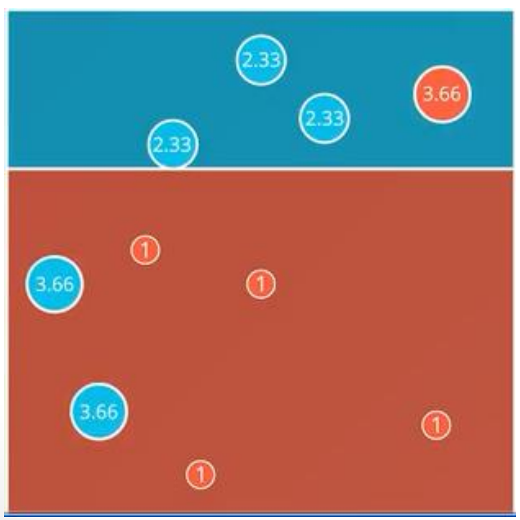



# AdaBoost – Pattern 2

## Weights after applying pattern 2




Correct: 11
Incorrect: 3




Correct: 11
Incorrect: 11

# AdaBoost – Pattern 3

## Weights after applying pattern 3

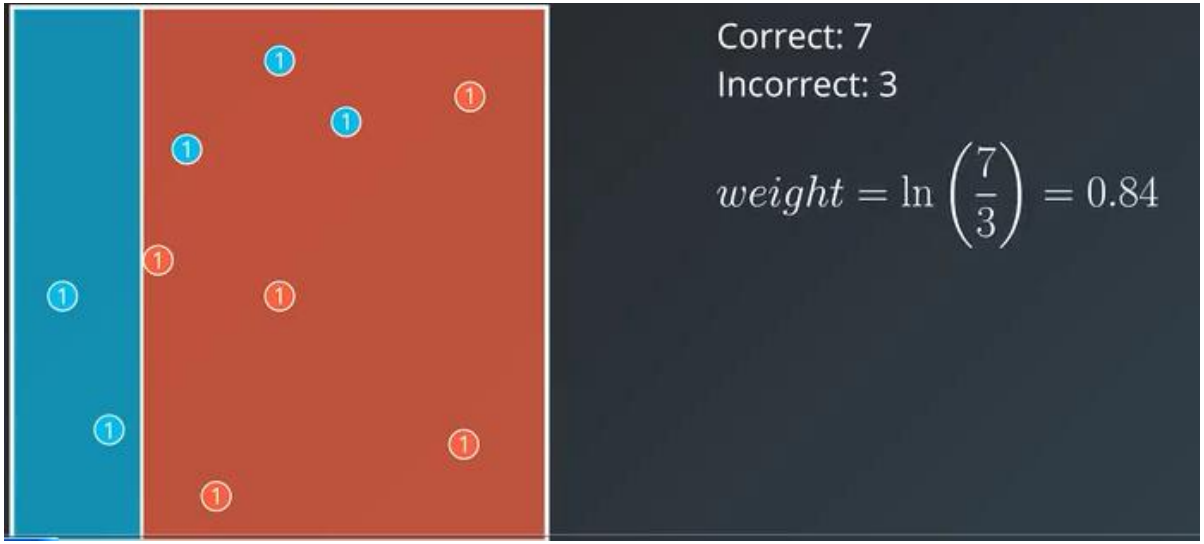

Correct: 19
Incorrect: 3

# AdaBoost – 3 Models

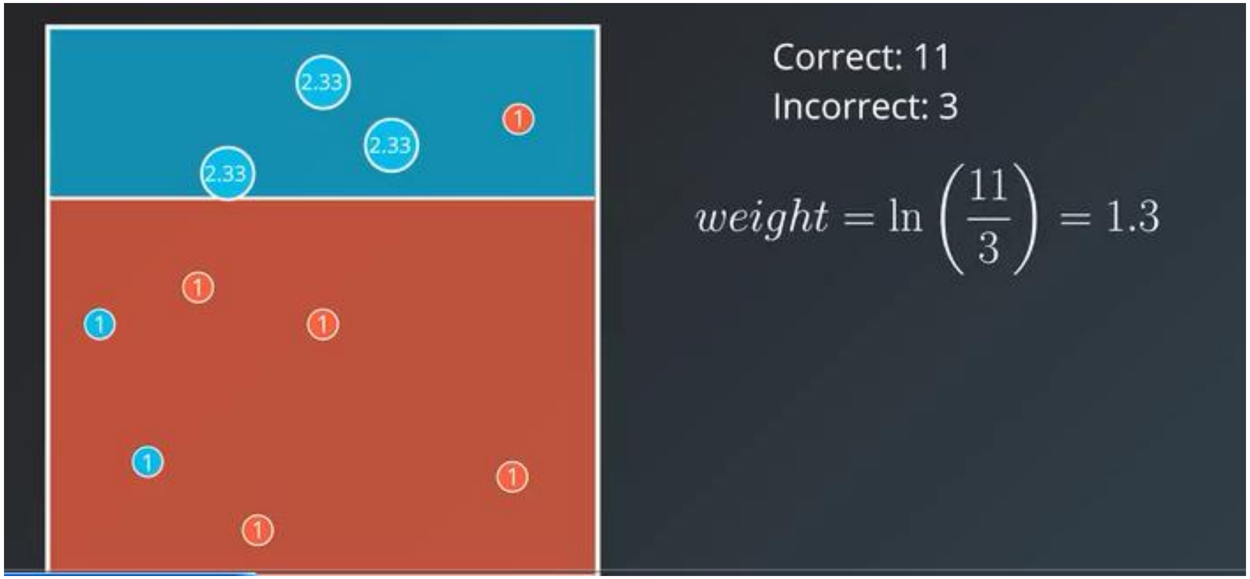
# Weightage of a Model

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

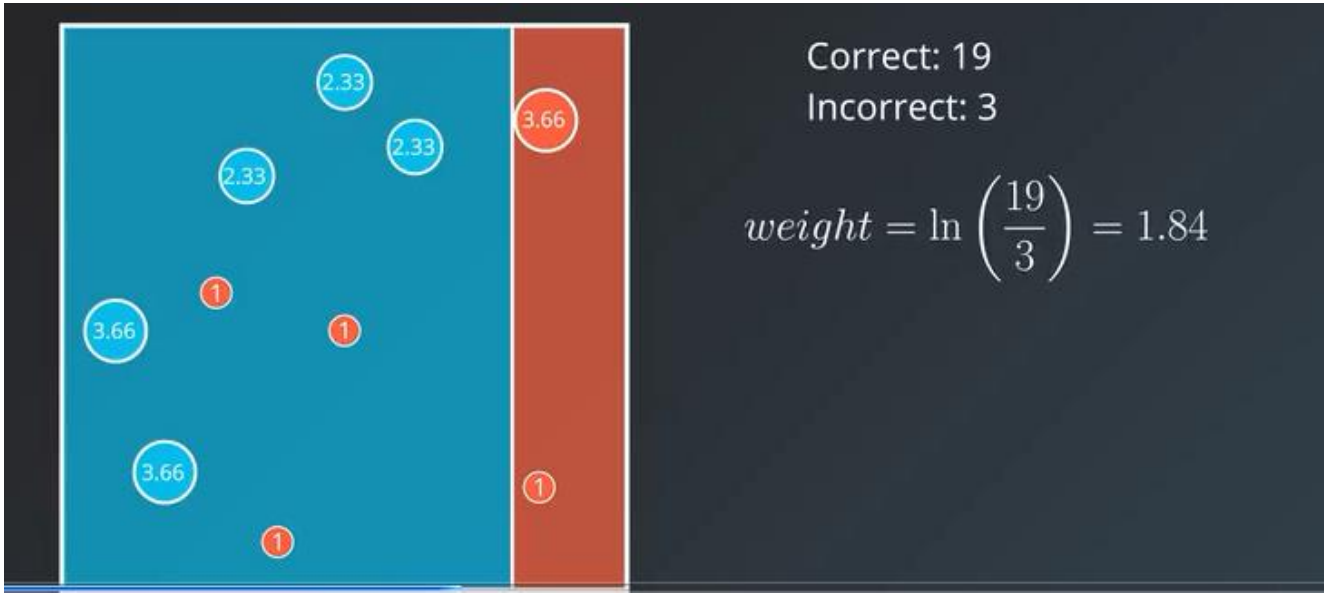
# Weight of Model 1



# Weight of Model 2



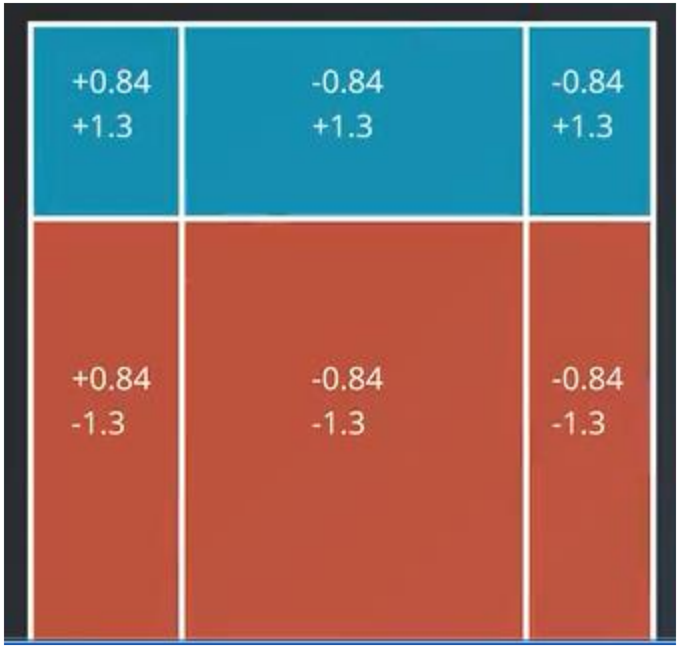
# Weight of Model 3



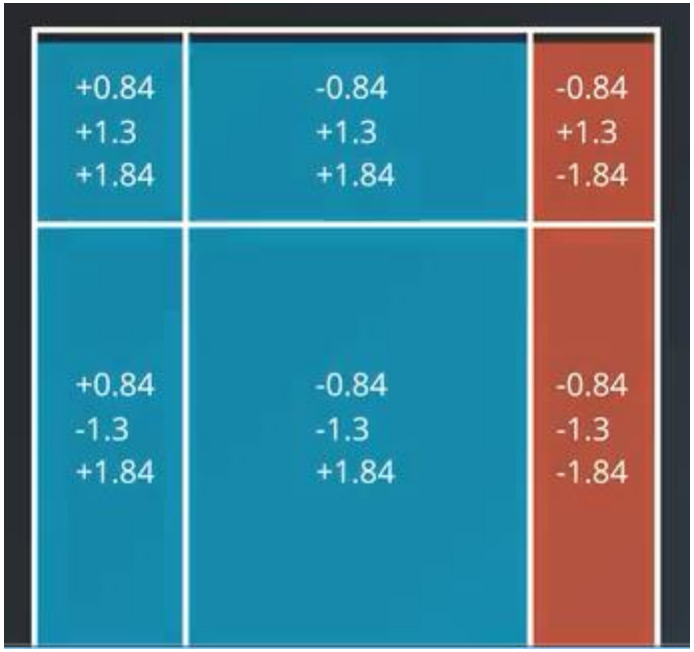
# Weight of 3 Models

# Assinging weights to 2 categories


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



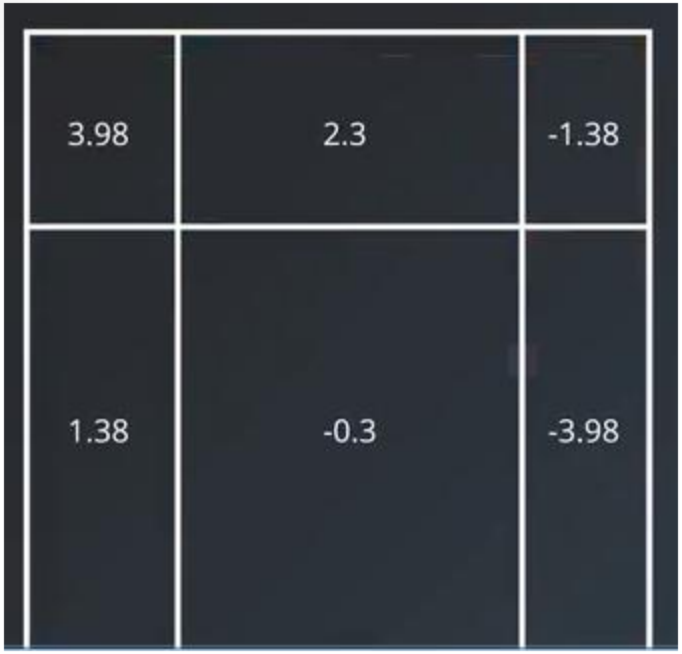
+0.84
+1.3
-0.84
+1.3
-0.84
+1.3
+0.84
-1.3
-0.84
-1.3
-0.84
-1.3

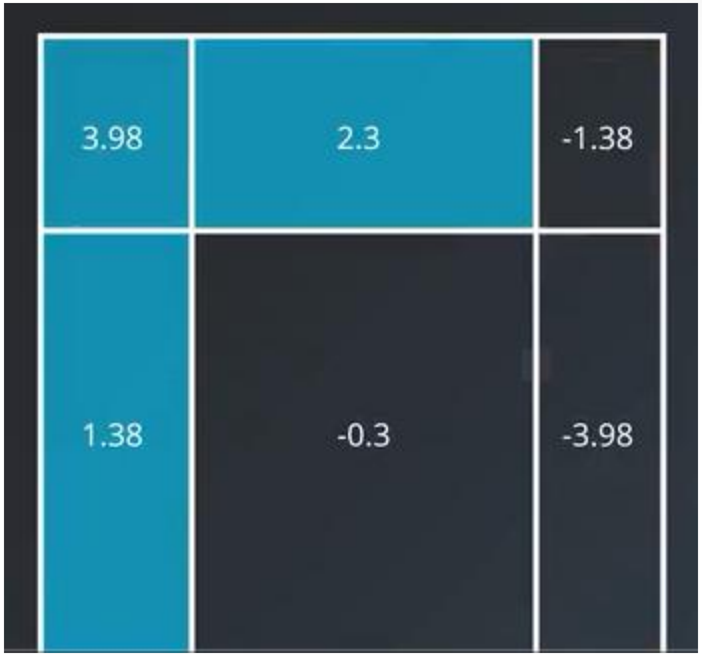


+0.84
+1.3
+1.84
-0.84
+1.3
+1.84
-0.84
+1.3
-1.84
+0.84
-1.3
+1.84
-0.84
-1.3
+1.84
-0.84
-1.3
-1.84

| | | |
|---|---|---|
| 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

# Final Model