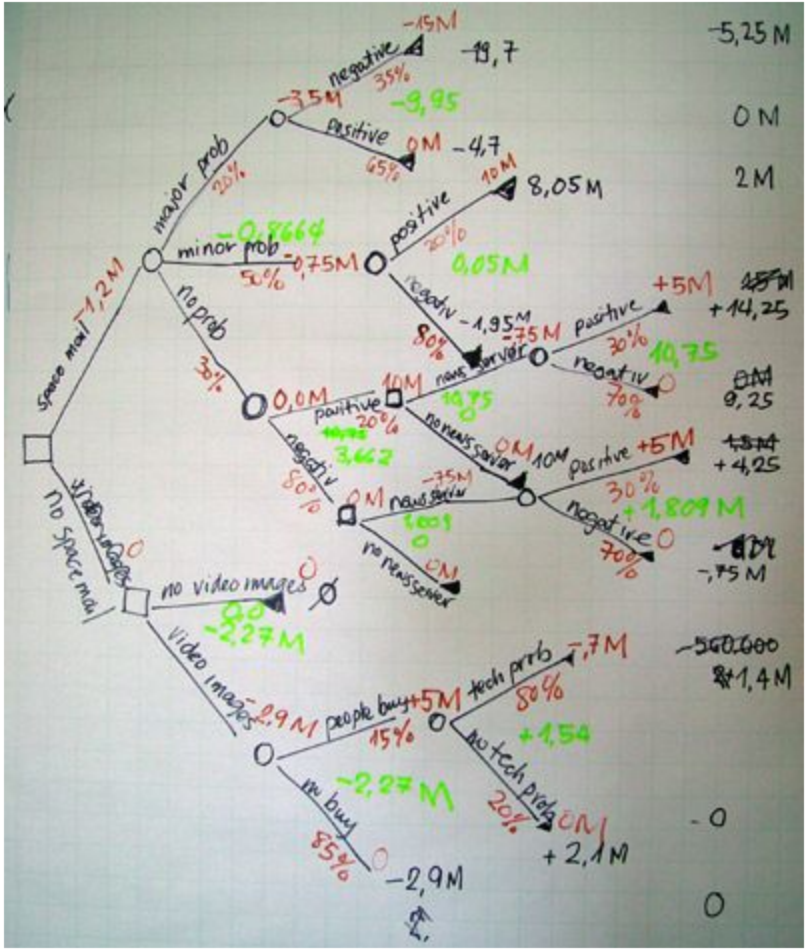
Decision Trees

Ayush Thada
16BCE1333

Topic for Discussion

- Introduction
- Working
- Mathematical Modelling
- Visualization
- Advantages
- Disadvantages
- Random Forest

Introduction



Introduction

- Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.
- The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.
- A decision tree is a decision support tool that uses a tree-like model of decisions and their possible consequences, including chance event outcomes, resource costs, and utility.
- It is one way to display an algorithm that only contains conditional control statements.

[Additional Info]

Machine Learning Algorithms

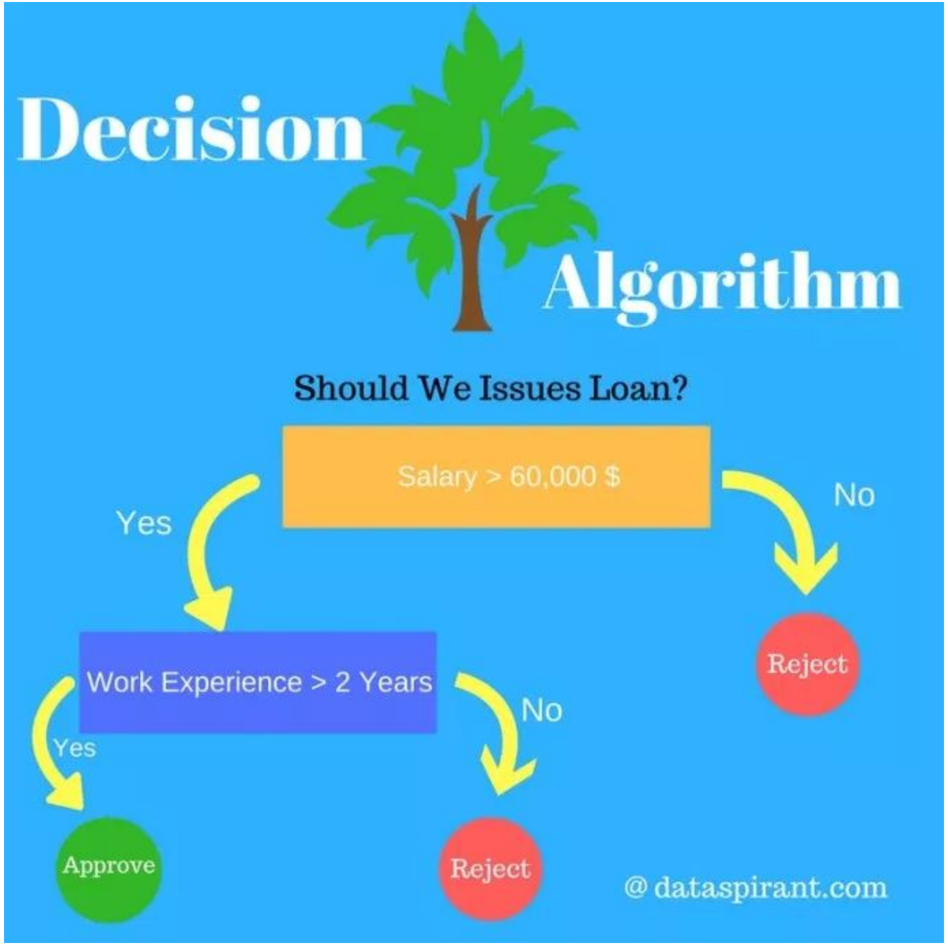
Parametric Algorithms

- A parametric algorithm has a fixed number of parameters.
- A parametric algorithm is computationally faster, but makes stronger assumptions about the data.
- The algorithm may work well if the assumptions turn out to be correct, but it may perform badly if the assumptions are wrong.
- Example: Linear regression, Support Vector Machines etc.

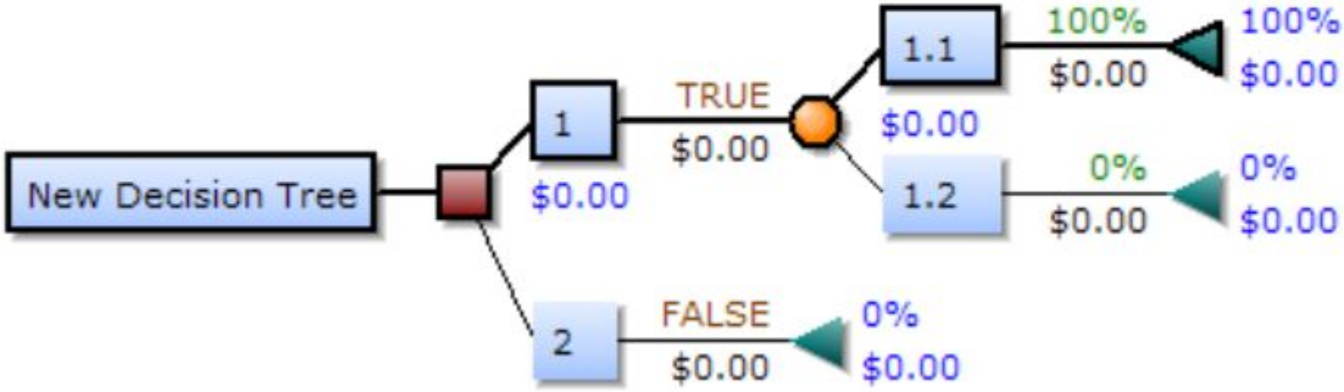
Non-Parametric Algorithms

- A non-parametric algorithm uses a flexible number of parameters, and the number of parameters often grows as it learns from more data.
- A non-parametric algorithm is computationally slower, but makes fewer assumptions about the data.
- Example: K-nearest neighbour, Decision Trees etc.

Working



Decision
Algorithm
Should We Issues Loan?
Salary > 60,000 $
Yes
No
Work Experience > 2 Years
Reject
Yes
No
Approve
Reject
@ dataspirant.com


New Decision Tree
1
$0.00
TRUE
$0.00
1.1
100%
$0.00
100%
$0.00
$0.00
1.2
0%
$0.00
0%
$0.00
2
FALSE
$0.00
0%
$0.00

Decision Tree Elements

- A decision tree consists of three types of nodes:

    - **Decision nodes** – typically represented by squares
    - **Chance nodes** – typically represented by circles
    - **End nodes** – typically represented by triangles

Decision Rules

- The decision tree can be linearized into decision rules, where the outcome is the contents of the leaf node, and the conditions along the path form a conjunction in the if clause.

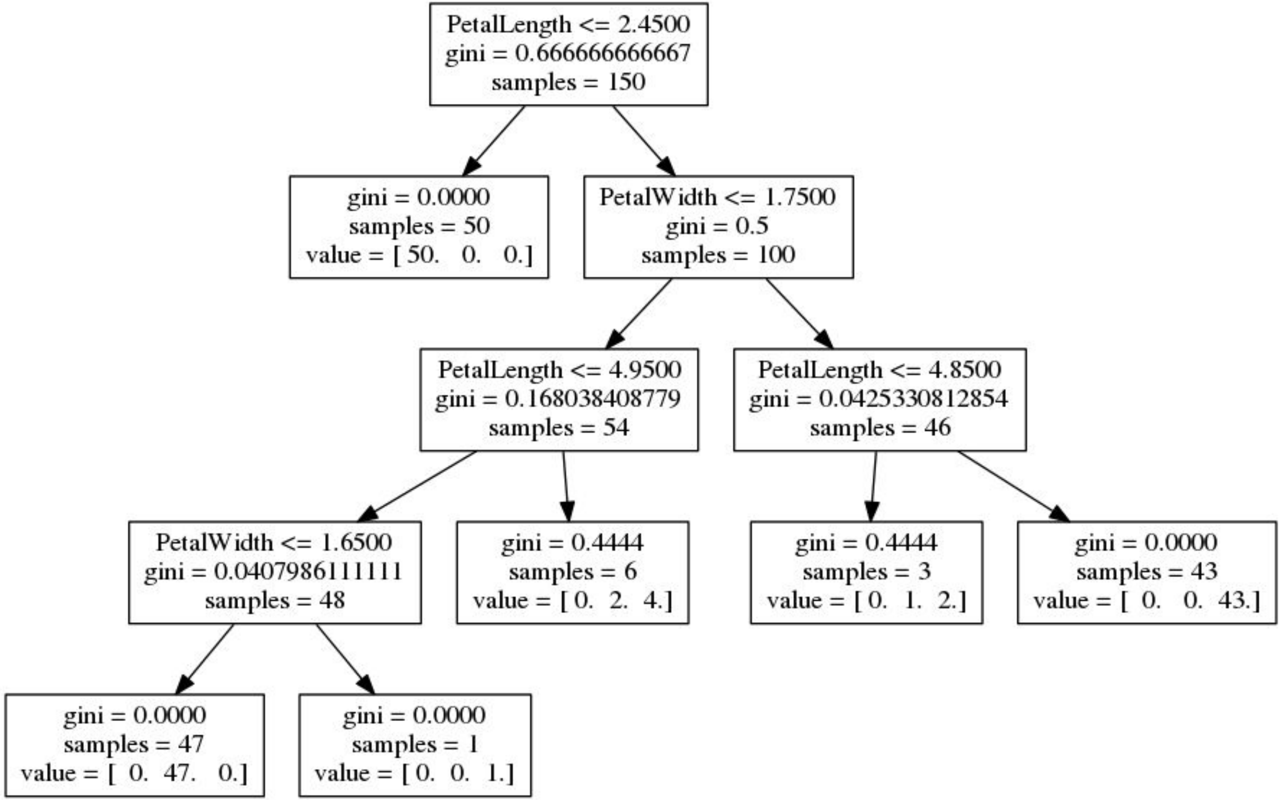
- In general, the rules have the form:

if condition1

and condition2

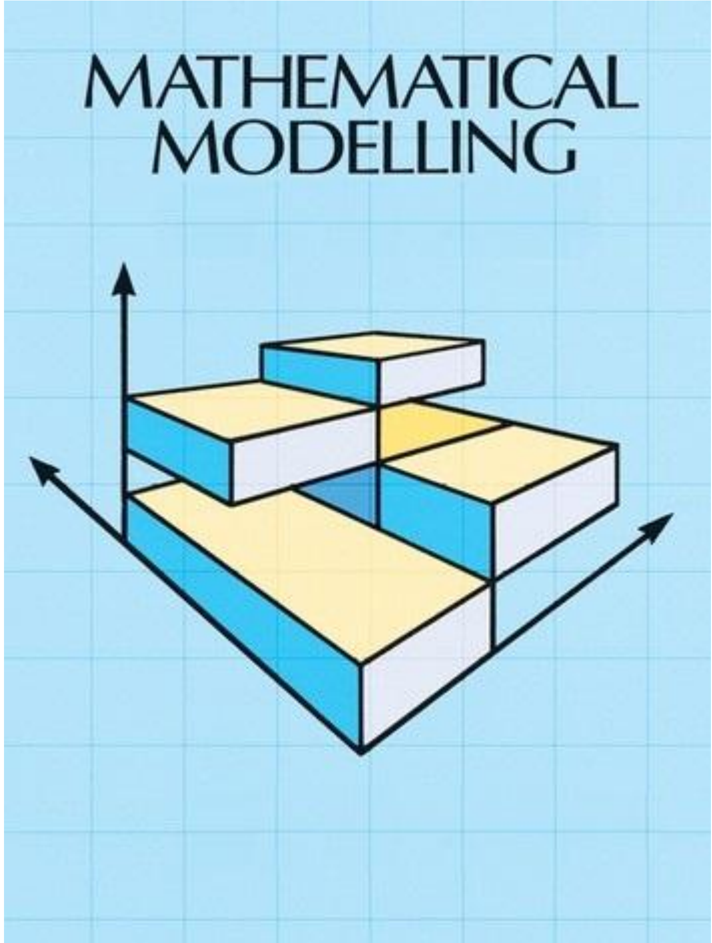
and condition 3

then outcome.



- Decision rules can be generated by constructing association rules with the target variable on the right. They can also denote temporal or causal relations.

Mathematical Modelling



There are couple of algorithms there to build a decision tree , out of them some of the most popular are as follow:

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

We are going to discuss ID3 algorithm for this session.

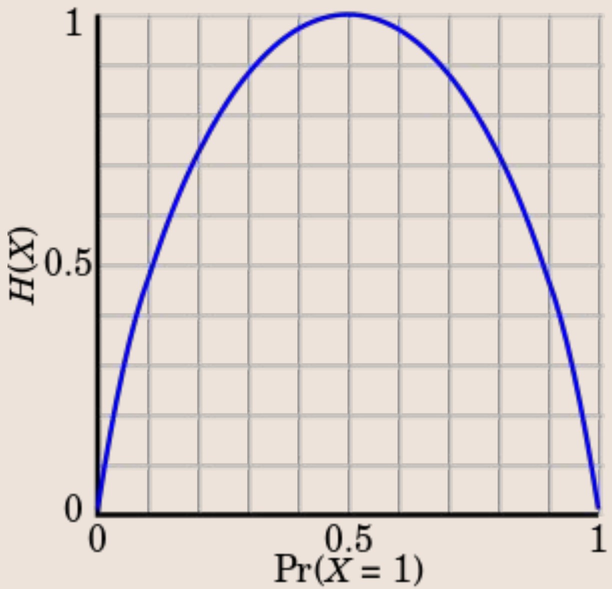
Entropy:

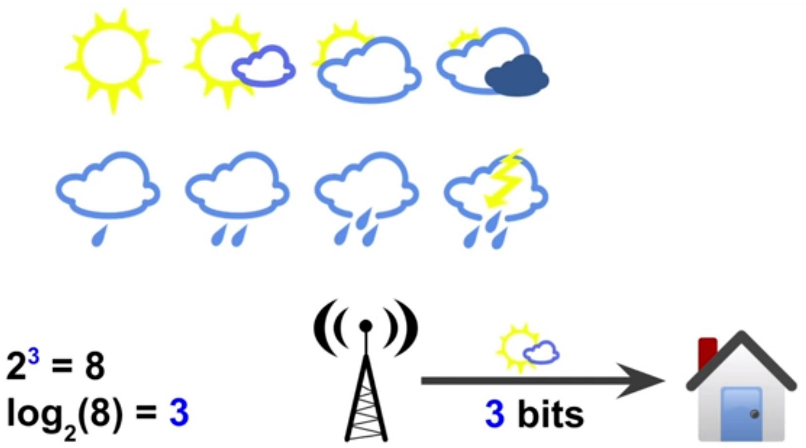
Expected number of bits need to encode class of randomly drawn sampled from a Probability distribution.

$$\textbf{Entropy} = -\sum p_i \log_2(p_i) = \mathbb{E}_{x\sim p(x)}[-\log(p(x))]$$

$$\textbf{Entropy} = H(S)$$

Here S is sample of training examples.





Probability of each event = ⅛

Entropy =

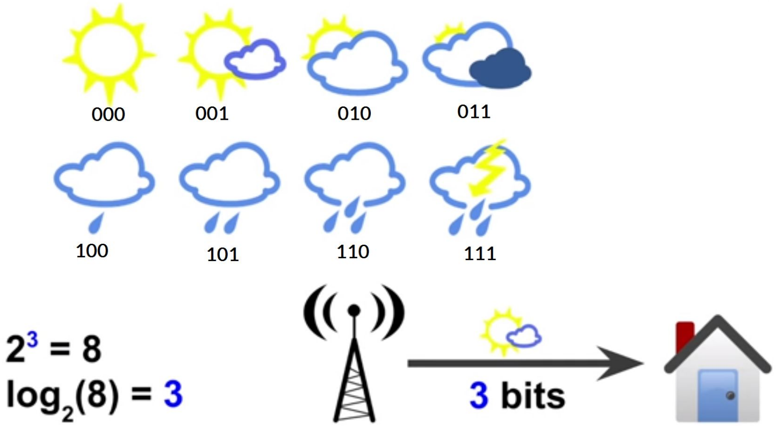
$-1/8\log(1/8)-1/8\log(1/8)-1/8\log(1/8)$
$-1/8\log(1/8)-1/8\log(1/8)-1/8\log(1/8)$
$-1/8\log(1/8)-1/8\log(1/8)$

Entropy = $8 \times (-1/8) \times \log(1/8)$

Entropy = $-\log(1/8)$

Entropy = $\log(8)$

Entopy = 3

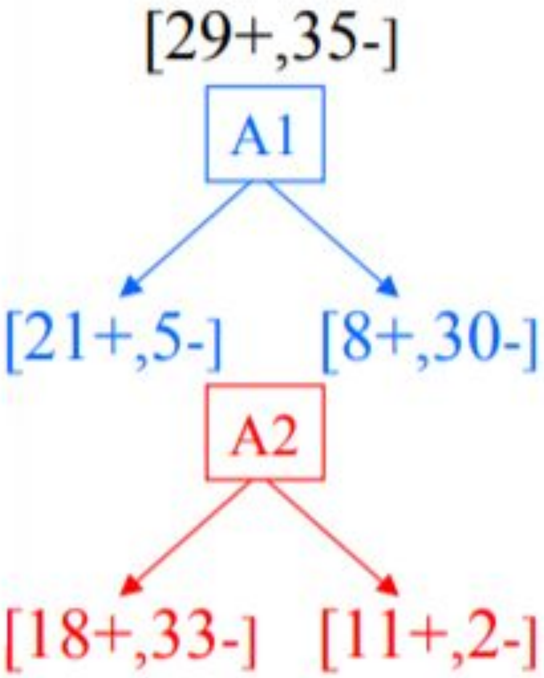


Information Gain:

Expected reduction in entropy due to sorting on an attribute. Also known as KL Divergence (in general terms).

IG(S,A) = Entropy(parent) -
[weighted average] x Entropy(children)

$$Gain(S,A) \equiv Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$



$$Entropy([29+,35-]) = -\frac{29}{64}\log_2(\frac{29}{64}) - \frac{35}{64}\log_2\frac{35}{64} = 0.994$$

$$Entropy([21+,5-]) = -\frac{21}{26}\log_2(\frac{21}{26}) - \frac{5}{26}\log_2\frac{5}{26} = 0.706$$

$$Entropy([8+,30-]) = 0.742$$

$$Gain(S,A1) = 0.994 - (\frac{26}{64}Entropy([21+,5-]) + \frac{38}{64}Entropy([8+,30-])) = 0.266$$

$$Entropy([18+,33-]) = 0.937$$

$$Entropy([11+,2-]) = 0.619$$

$$Gain(S,A2) = 0.121$$

Gini Index:

It is a measure of statistical dispersion intended to represent the income or wealth distribution of a nation's residents. Here dispersion simply means an extent upto which some distribution can be stretched.

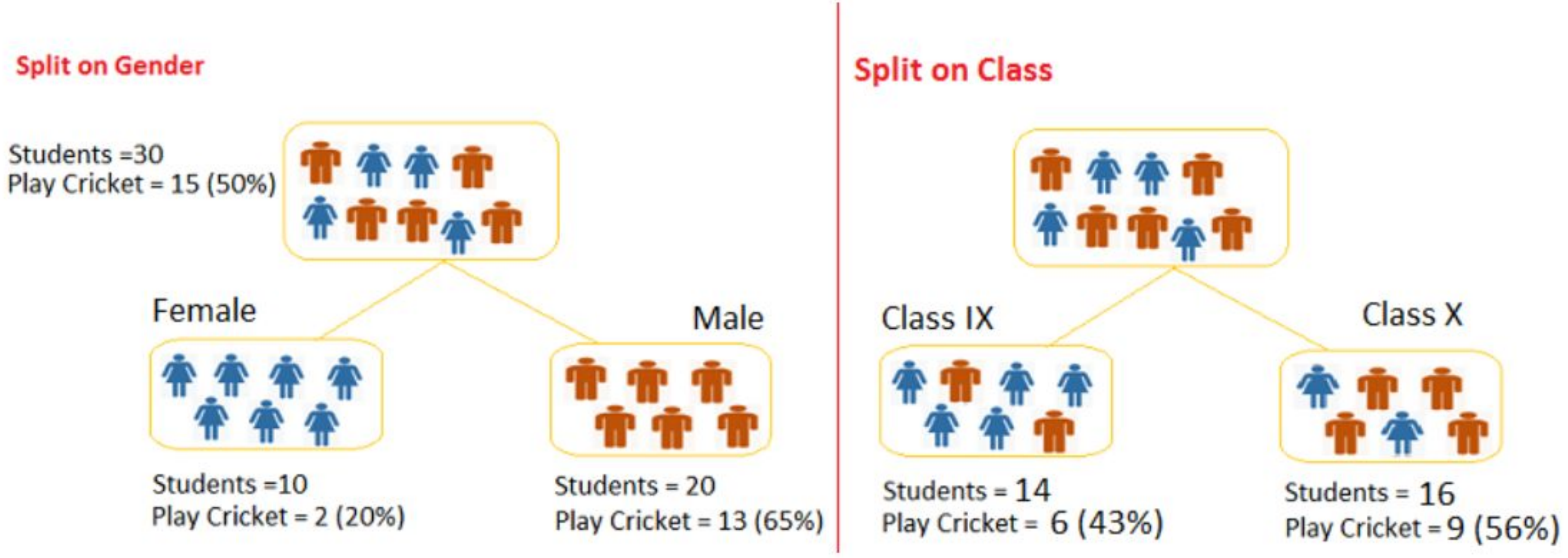
$$\mathbf{GI} = \sum_{i \neq j} p(i)p(j) = \mathbf{1} - \sum_{t=0 \rightarrow t=k} p^2(t)$$

, where no of classes are from 0 to k.
But it performs only binary split. Higher the value of Gini higher the homogeneity.

[Economics]

Gini Gain:

G(S,A) = Gini_Index(parent) -
[weighted average] x Gini_Index(children)



Gini for Parent Node = 1 - (0.5)*(0.5)-(0.5)*(0.5)= 0.5

<u>Split on Gender:</u>

Gini for sub-node Female = 1 - (0.2)*(0.2)+(0.8)*(0.8)= 0.32

Gini for sub-node Male = 1 - (0.65)*(0.65)+(0.35)*(0.35)= 0.45

Gini Gain for Split Gender = 0.5 - {(10/30)*0.68+(20/30)*0.55} = 0.0934

<u>Similar for Split on Class:</u>

Gini for sub-node Class IX = 1 - (0.43)*(0.43)+(0.57)*(0.57)=0.49

Gini for sub-node Class X = 1 - (0.56)*(0.56)+(0.44)*(0.44)=0.49

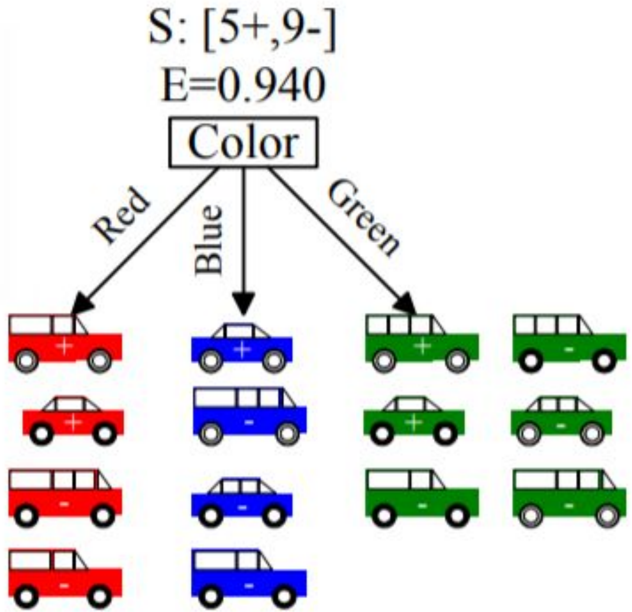
Gini Gain for Split Class = 05 - {(14/30)*0.51+(16/30)*0.51} = 0.01

Above, you can see that Gini Gain for Split on Gender is higher than Split on Class hence, the node split will take place on Gender.

Solved Example IG3 Method

Color	Type	Doors	Tires	Class
Red	SUV	2	Whitewall	+
Blue	Minivan	4	Whitewall	-
Green	Car	4	Whitewall	-
Red	Minivan	4	Blackwall	-
Green	Car	2	Blackwall	+
Green	SUV	4	Blackwall	-
Blue	SUV	2	Blackwall	-
Blue	Car	2	Whitewall	+
Red	SUV	2	Blackwall	-
Blue	Car	4	Blackwall	-
Green	SUV	4	Whitewall	+
Red	Car	2	Blackwall	+
Green	SUV	2	Blackwall	-
Green	Minivan	4	Whitewall	-

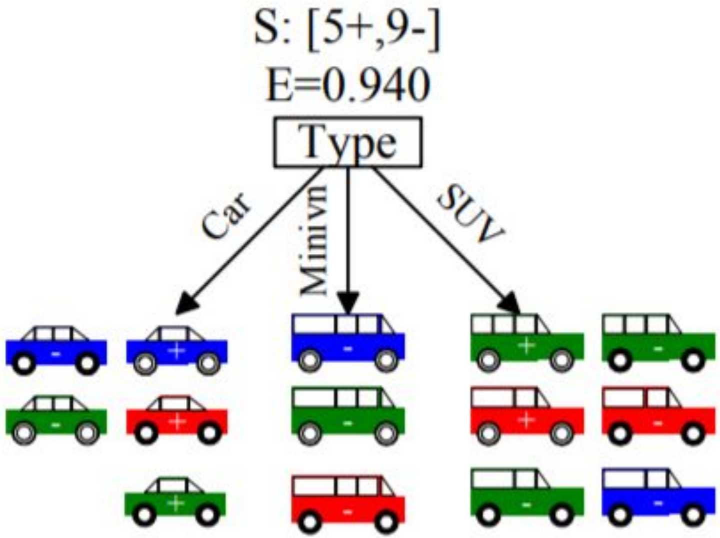
Selection of Root Attribute



Gain(S,Color)

$= .940 - (4/14)1.0 - (4/14).811 - (6/14).918$

$= 0.029$

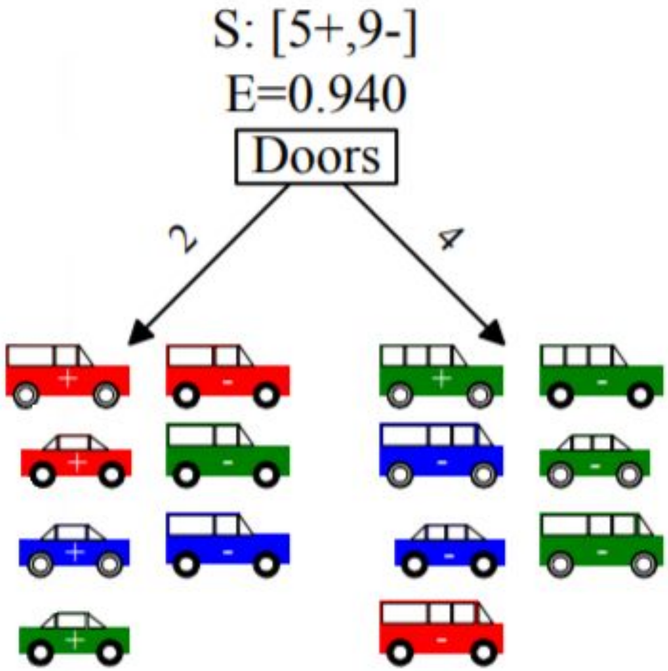


Gain(S,Type)

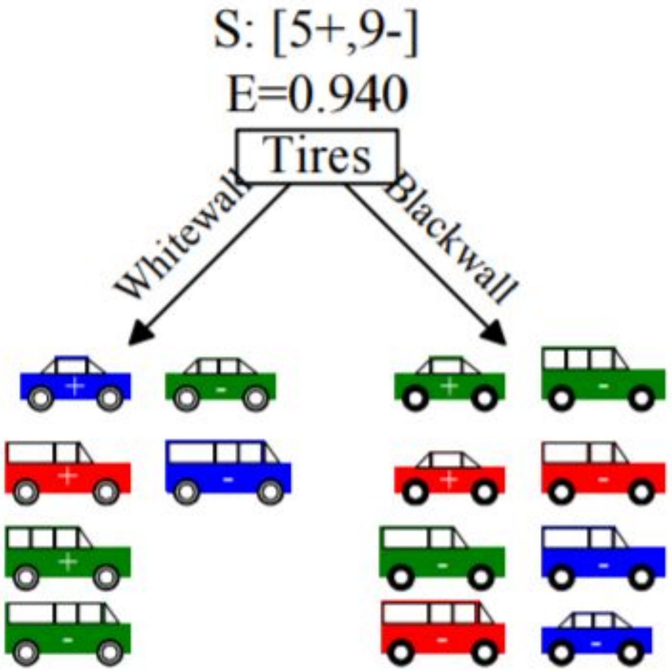
$= .940 - (5/14).971 - (3/14)0 - (6/14)0.918$

$= 0.200$

Selection of Root Attribute

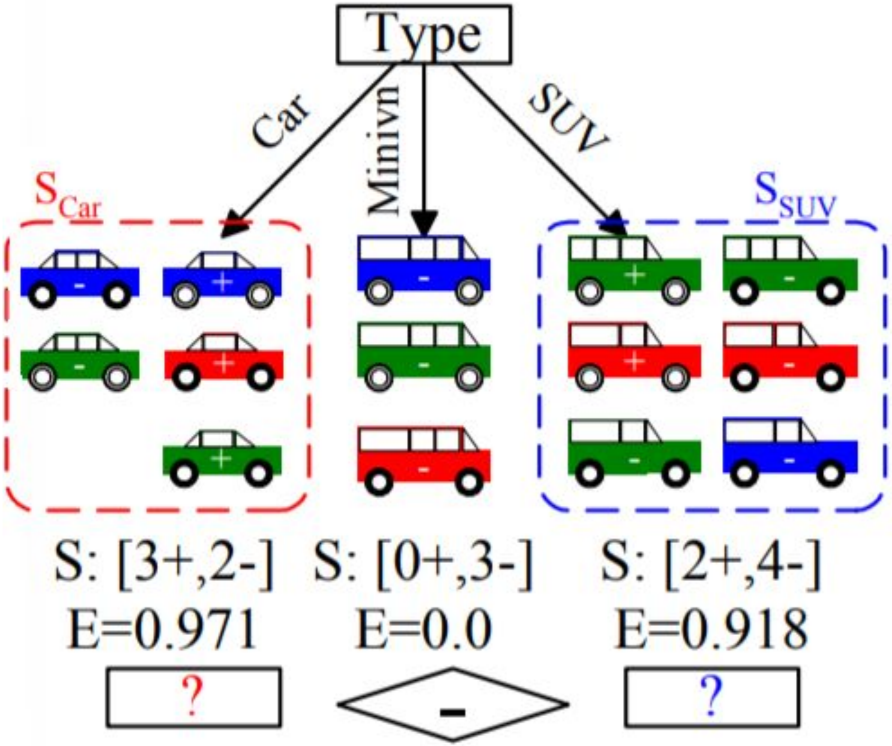


Gain(S,Doors)
= .940 - (7/14)0.985 - (7/14)0.592
= 0.152



Gain(S,Type)
= .940 - (6/14)1.0 - (8/14).811
= 0.048

Best Attribute is : TYPE



$Gain(S_{Car},Color) =$
$.971-(1/5)0.0-(2/5)1.0-(2/5)1.0=.171$

$Gain(S_{Car,}Doors)=$
$.971-(3/5)0.0-(2/5)0.0= .971$

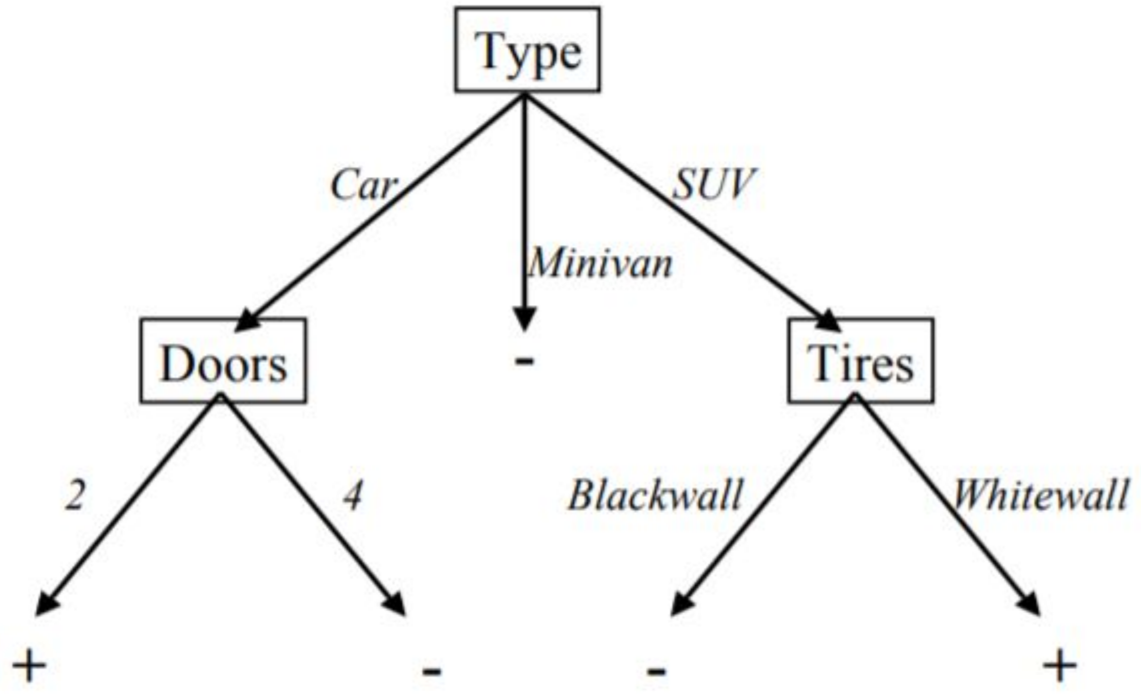
$Gain(S_{Car},Tires)=$
$.971-(2/5)1.0-(3/5).918= .020$

$Gain(S_{SUV},Color)=$
$.918-(2/6)1.0-(1/6)0.0-(3/6).918= .126$

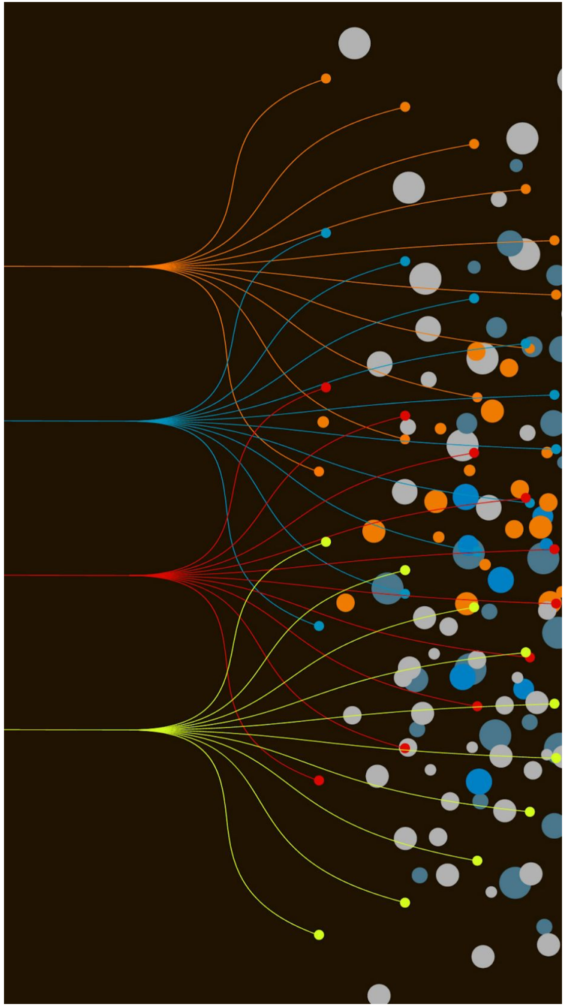
$Gain(S_{SUV},Doors)=$
$.918-(4/6).811-(2/6)1.0= .044$

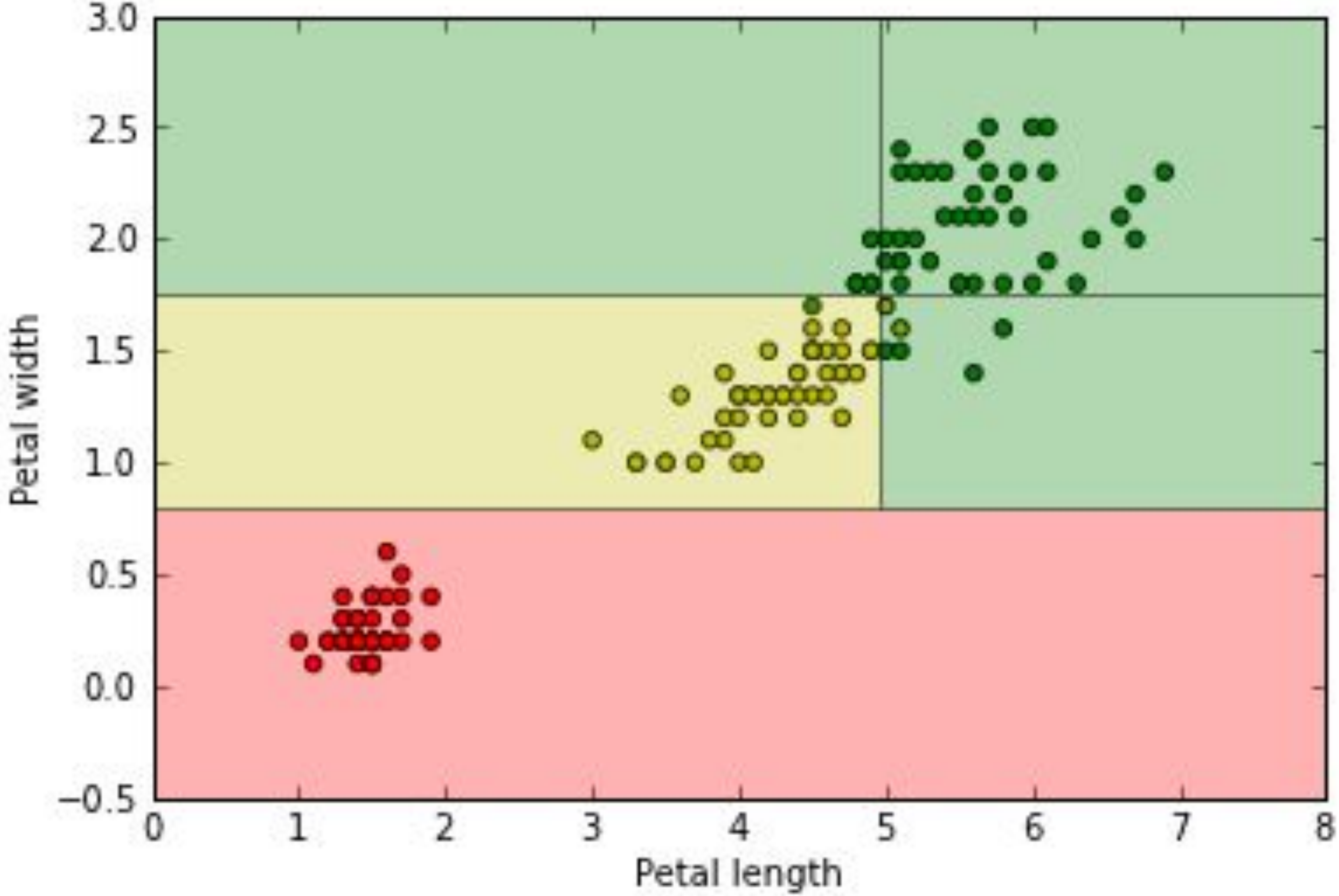
$Gain(S_{SUV},Tires)=$
$.918-(2/6)0.0-(4/6)0.0= .918$

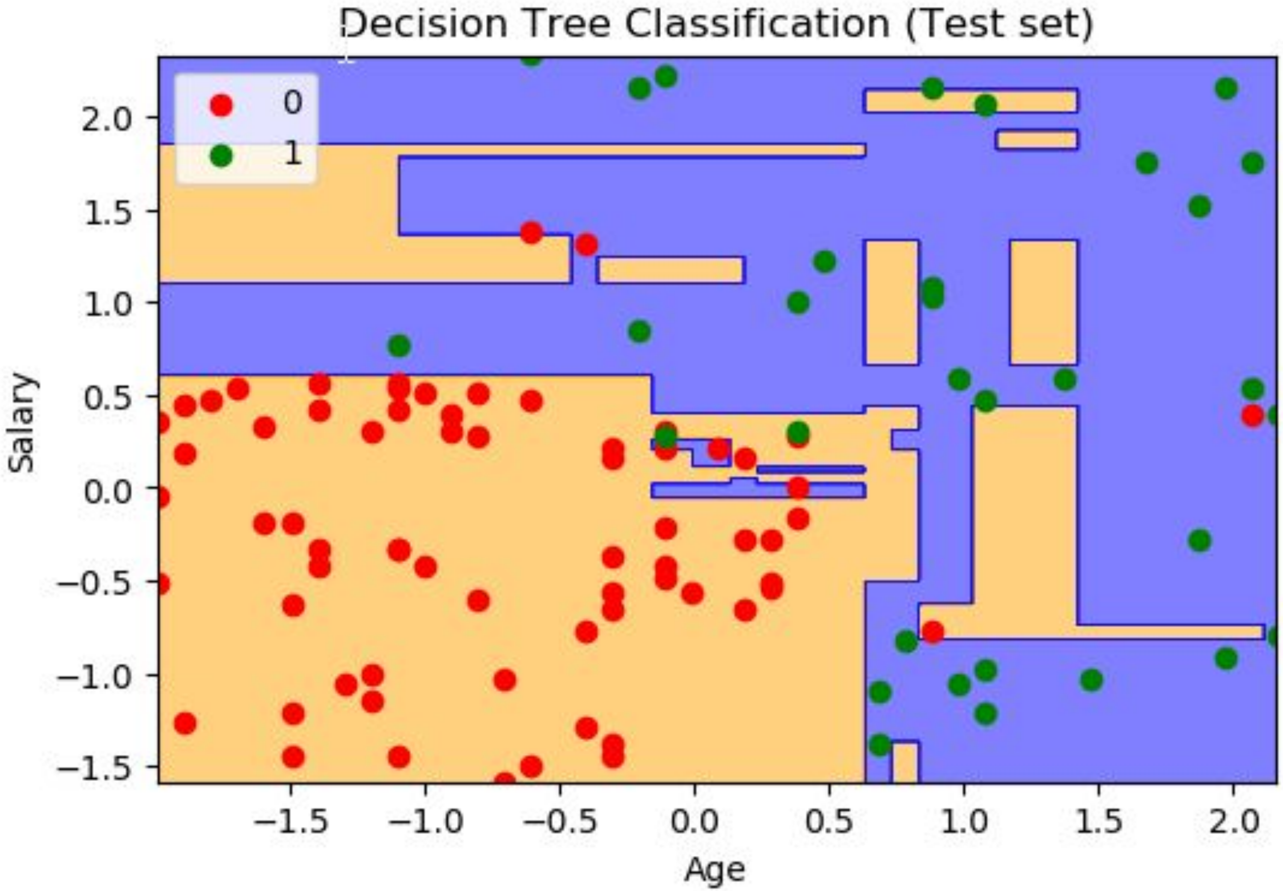
Final Decision Tree



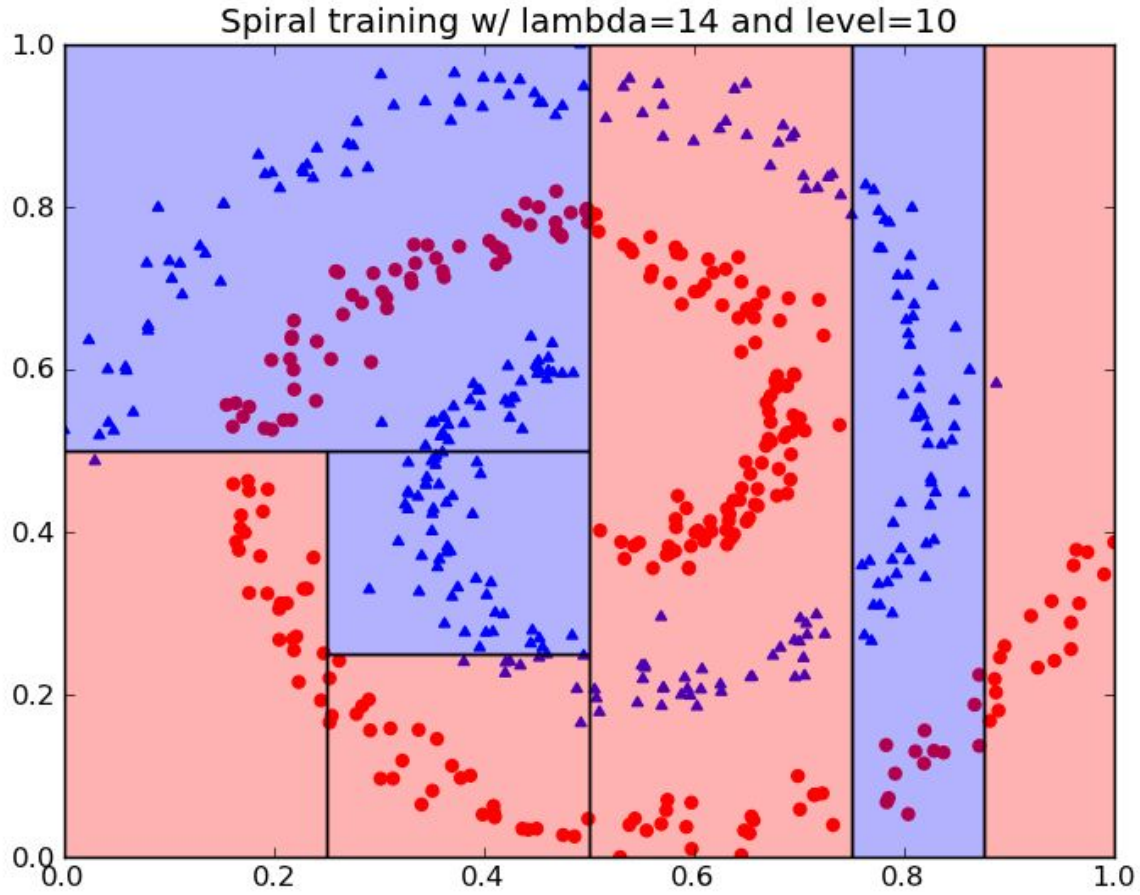
Visualization




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

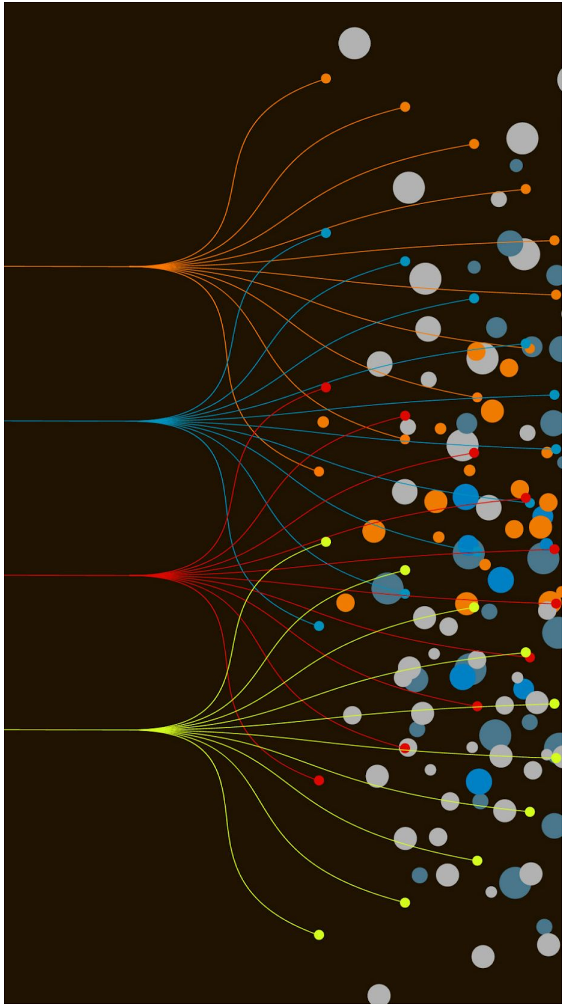


http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html



Spiral training w/ lambda=14 and level=10
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

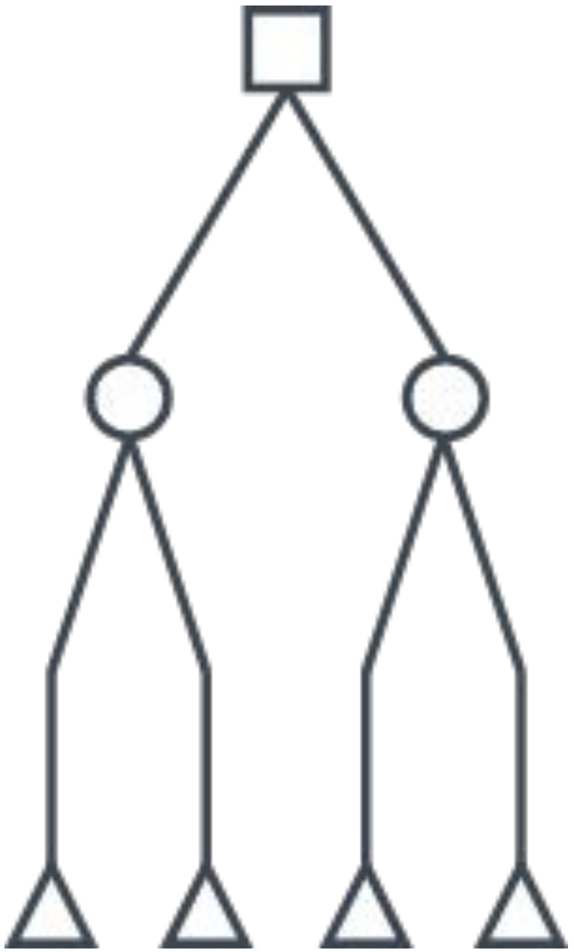
Advantages



- **Graphic**: You can represent decision alternatives, possible outcomes, and chance events schematically. The visual approach is particularly helpful in comprehending sequential decisions and outcome dependencies.
- **Efficient:** You can quickly express complex alternatives clearly. You can easily modify a decision tree as new information becomes available. Set up a decision tree to compare how changing input values affect various decision alternatives. Standard decision tree notation is easy to adopt.
- **Revealing:** You can compare competing alternatives-even without complete information-in terms of risk and probable value. The Expected Value (EV) term combines relative investment costs, anticipated payoffs, and uncertainties into a single numerical value. The EV reveals the overall merits of competing alternatives.
- **Complementary:** You can use decision trees in conjunction with other project management tools. For example, the decision tree method can help evaluate project schedules.

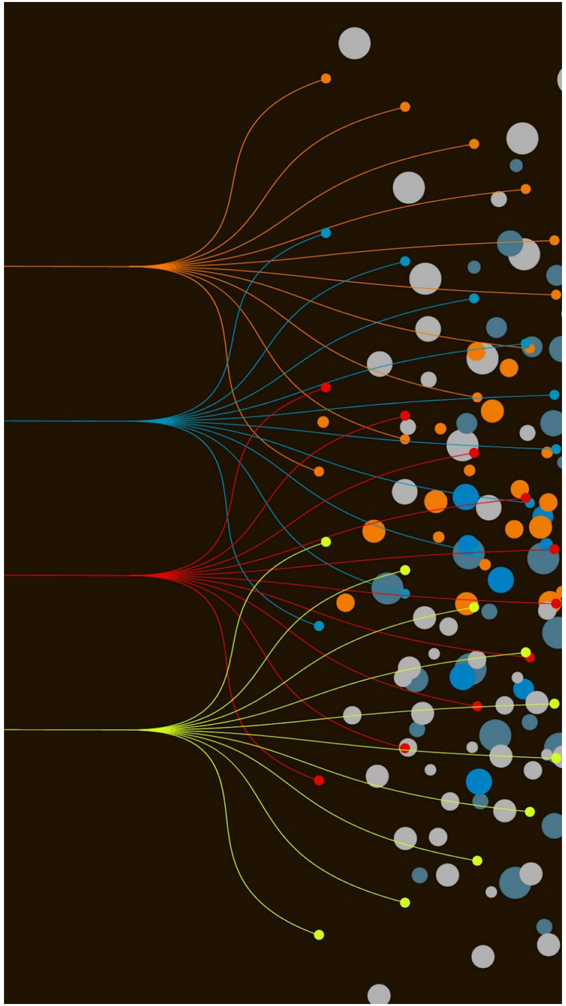
- Decision trees are **self-explanatory** and when compacted they are also easy to follow. In other words if the decision trees has a reasonable number of leaves, it can be grasped by non-professional users. Furthermore decision trees can be converted to a set of rules. Thus, this representation is considered as comprehensible.
- Decision trees can handle both **nominal and numerical attributes**.
- Decision trees representation is rich enough to represent any **discrete-value classifier**.
- Decision trees are capable of handling datasets that may have errors.
- Decision trees are capable of handling datasets that may have missing values.
- Decision trees are considered to be a **nonparametric method**. This means that decision trees have no assumptions about the space distribution and the classifier structure.

Disadvantages



- Most of the algorithms (like ID3 and C4.5) require that the target attribute will have only discrete values.
- As decision trees use the “divide and conquer” method, they tend to perform well if a few highly relevant attributes exist, but less so if many complex interactions are present. One of the reasons for this is that other classifiers can compactly describe a classifier that would be very challenging to represent using a decision tree.
- The greedy characteristic of decision trees leads to another disadvantage that should be pointed out. This is it's over-sensitivity to the training set, to irrelevant attributes and to noise.
- Decision tree learners create biased trees if some classes dominate. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Random Forest



- This methods falls under the category of Bootstrap aggregation ensemble method.
- Given a training set $X = [x_1, ..., x_n]$with responses $Y = [y_1, ..., y_n]$, bagging repeatedly (B times) selects a random sample with replacement of the training set and fits trees to these samples:

  For b = 1, ..., B:

  Sample, with replacement, n training examples from X, Y; call these X_b, Y_b.

  Train a classification or regression tree f_b on X_b, Y_b.
- Final result can be calculated using majority voting in case of classification and averaging in case of regression problem.

- http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html

References

- Cristina Petri: http://www.cs.ubbcluj.ro/~gabis/DocDiplome/DT/DecisionTrees.pdf
- https://en.wikipedia.org/wiki/Decision_tree
- https://www.datasciencecentral.com/profiles/blogs/random-forests-explained-intuitively
- http://arogozhnikov.github.io/2016/06/24/gradient_boosting_explained.html
- https://www.youtube.com/watch?v=NsUqRe-9tb4
- https://scikit-learn.org/stable/modules/tree.html
- https://www.youtube.com/watch?v=ErfnhcEV1O8