Decision Tree

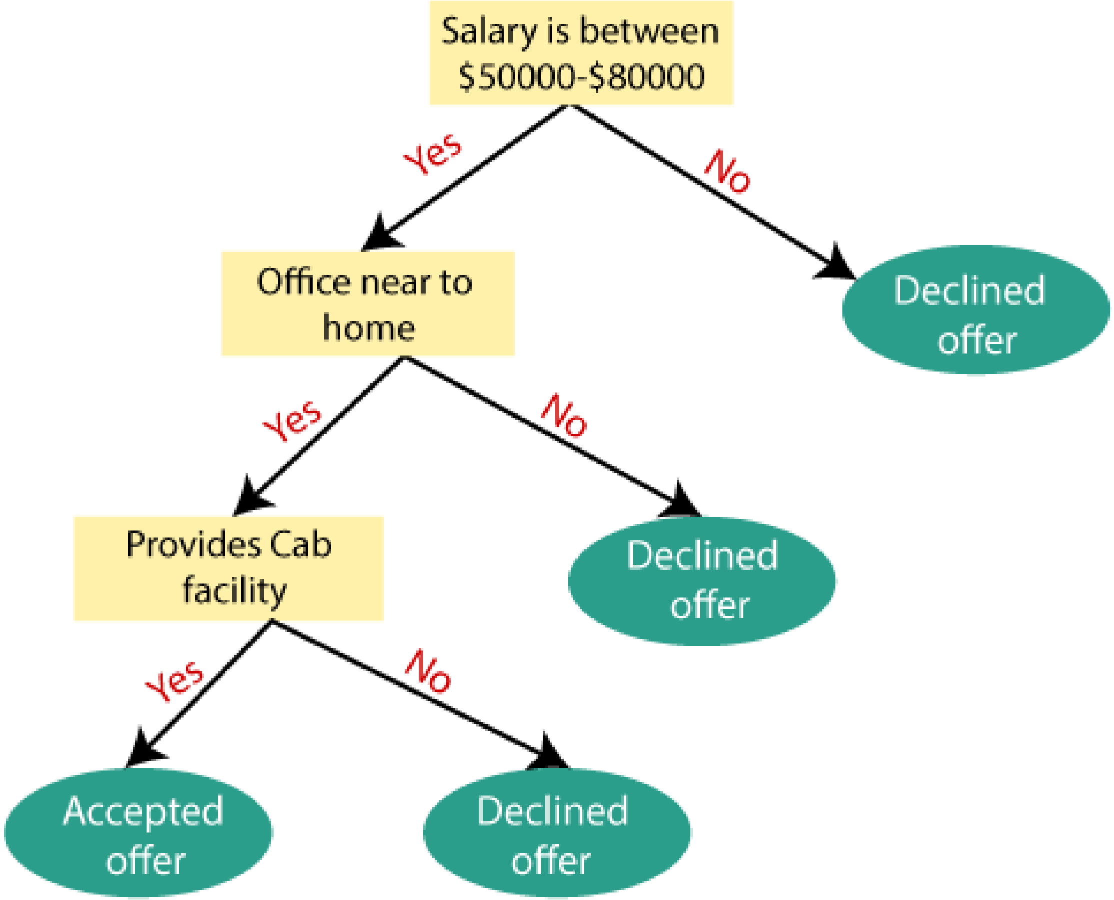
Tushar B. Kute,

http://tusharkute.com

Lets see the example...

- Suppose a job seeker was deciding between several offers, some closer or further from home, with various levels of pay and benefits.
- He or she might create a list with the features of each position. Based on these features, rules can be created to eliminate some options.
- For instance, "if I have a commute longer than an hour, then I will be unhappy", or "if I make less than $50k, I won't be able to support my family."
- The difficult decision of predicting future happiness can be reduced to a series of small, but increasingly specific choices.

Lets see the example...



Decision tree

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems.
- It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.

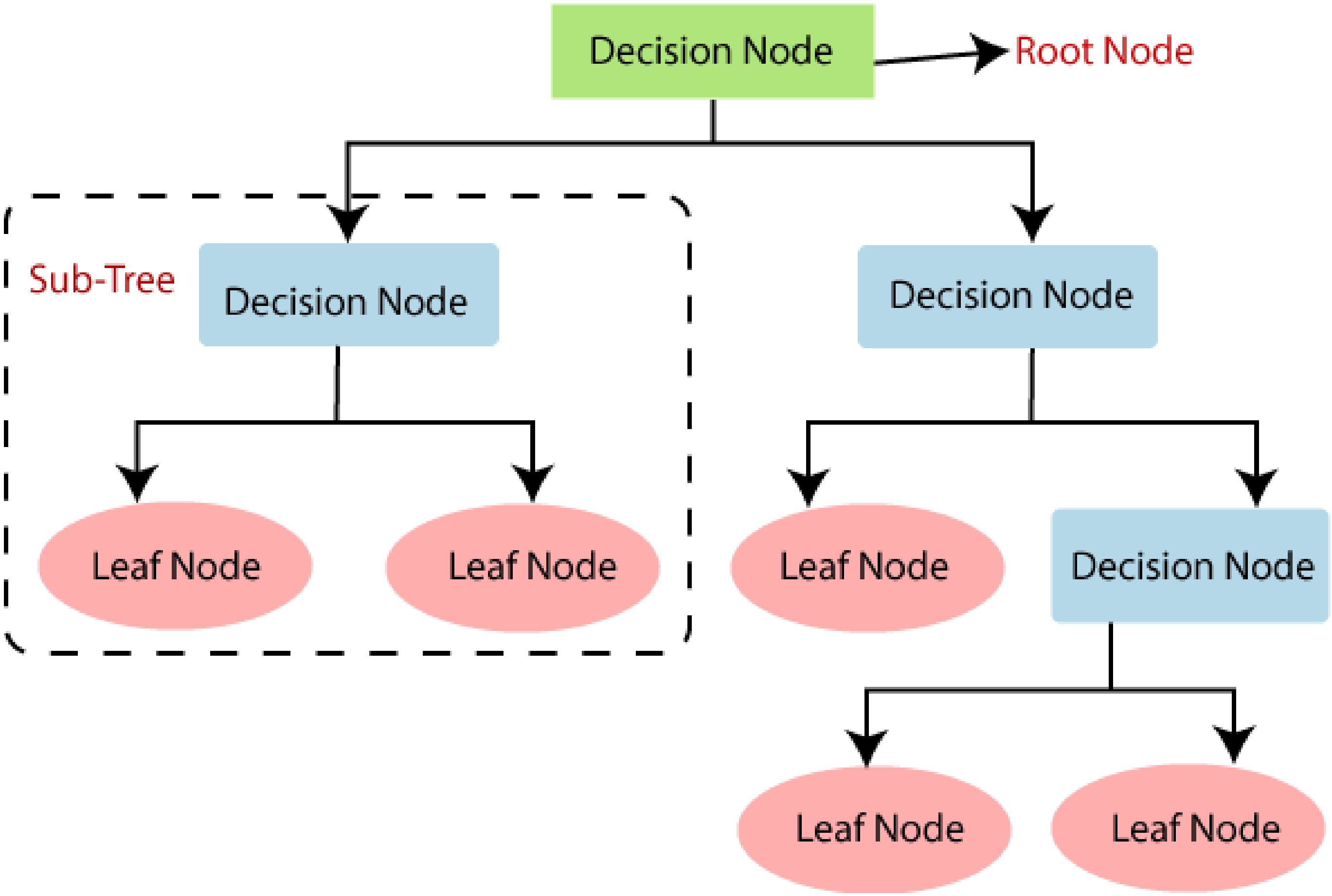
Understanding Decision tree

- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node.
- Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.

Decision tree

- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.
- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.

Decision tree



Divide and Conquer

- Decision trees are built using a heuristic called recursive partitioning.
- This approach is generally known as divide and conquer because it uses the feature values to split the data into smaller and smaller subsets of similar classes.
- Beginning at the root node, which represents the entire dataset, the algorithm chooses a feature that is the most predictive of the target class.
- The examples are then partitioned into groups of distinct values of this feature; this decision forms the first set of tree branches.

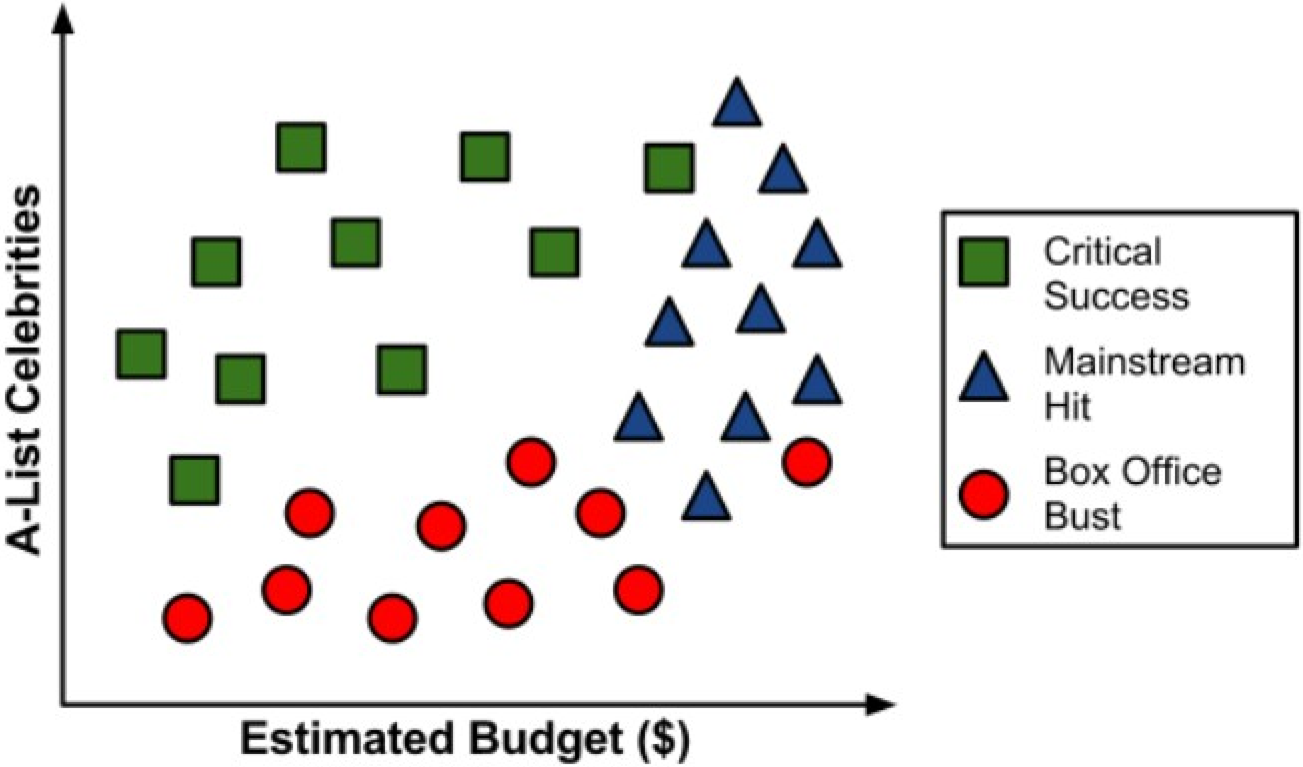
Divide and Conquer

- **To illustrate the tree building process, let's consider a simple example.**
- Imagine that you are working for a Hollywood film studio, and your desk is piled high with screenplays.
- Rather than read each one cover-to-cover, you decide to develop a decision tree algorithm to predict whether a potential movie would fall into one of three categories: mainstream hit, critic's choice, or box office bust.

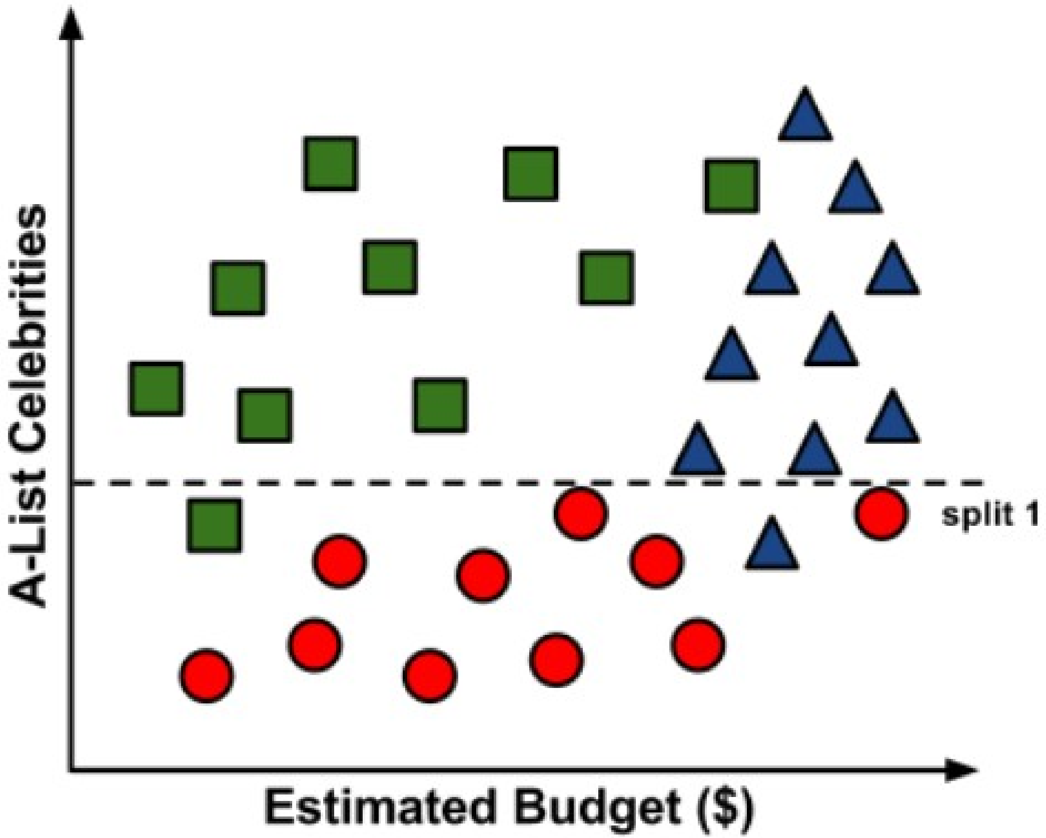
Continuing...

- To gather data for your model, you turn to the studio archives to examine the previous ten years of movie releases.
- After reviewing the data for 30 different movie scripts, a pattern emerges.
- There seems to be a relationship between the film's proposed shooting budget, the number of A-list celebrities lined up for starring roles, and the categories of success.
- A scatter plot of this data might look something like...

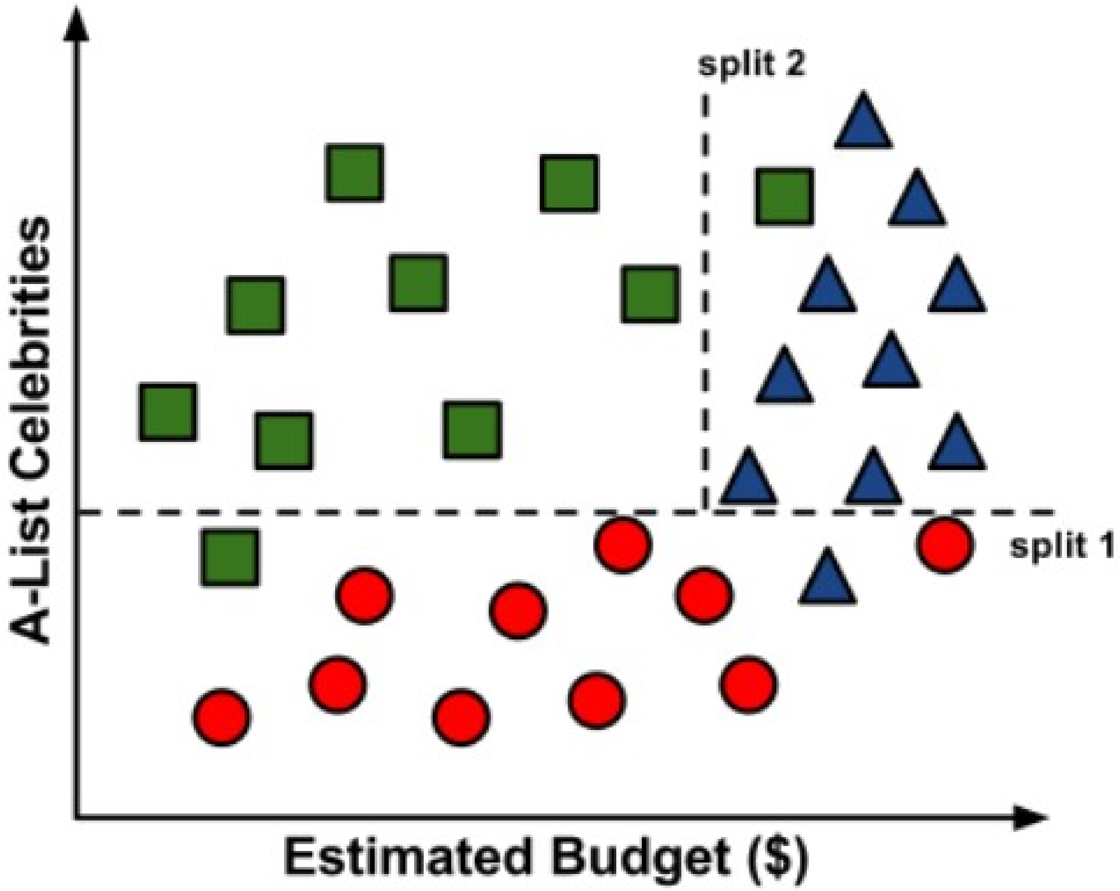
The scatterplot



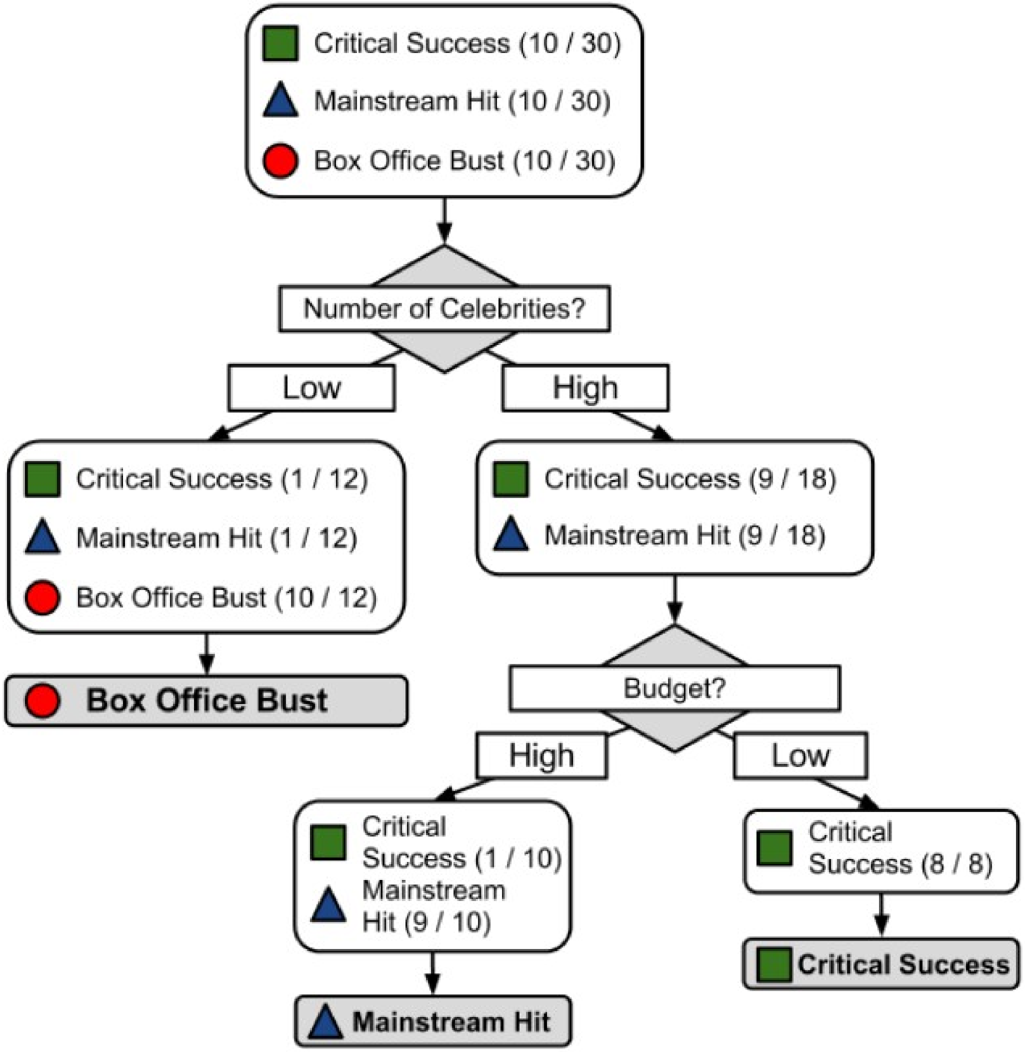
Scatterplot – Phase:1



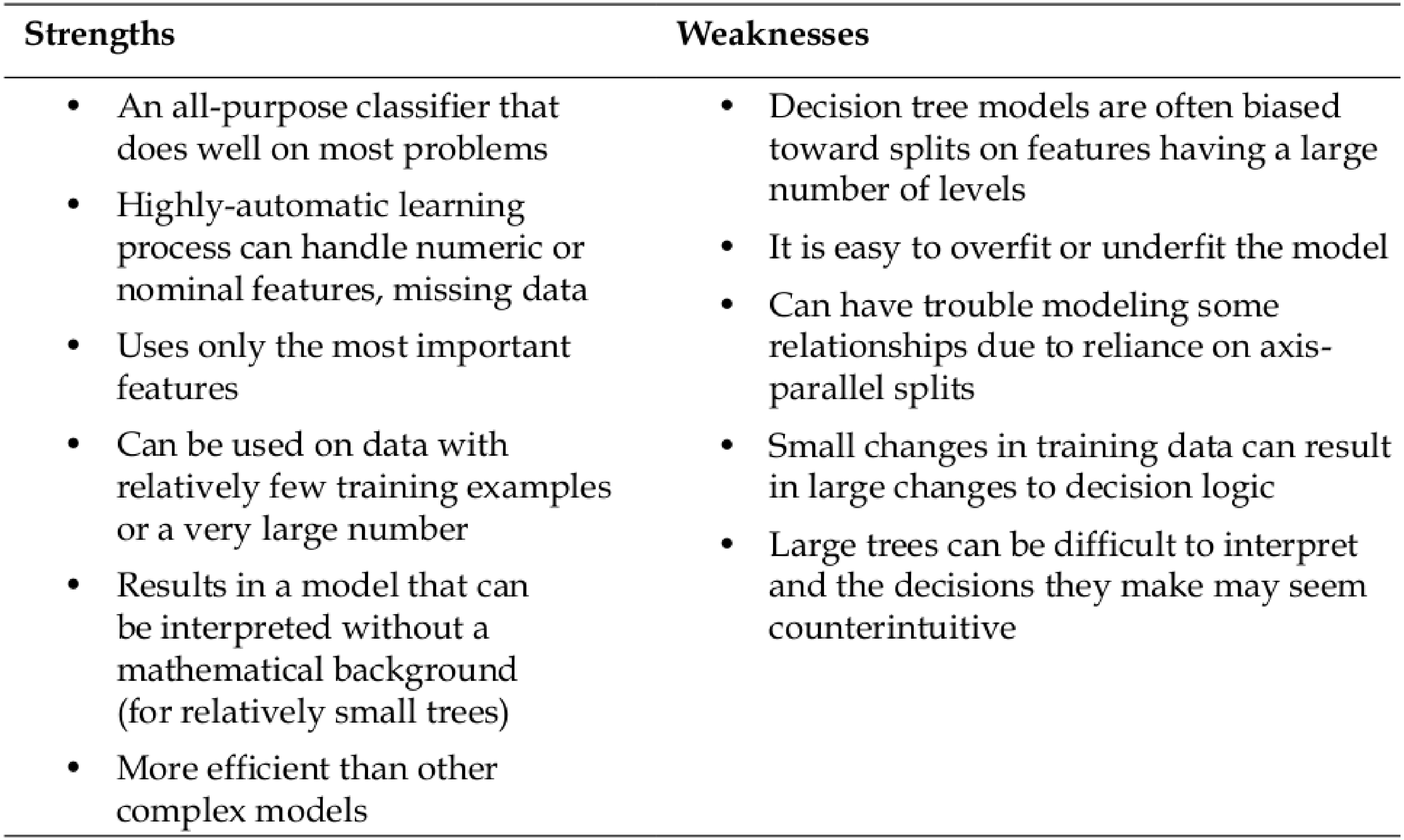
Scatterplot – Phase:2



The decision tree model

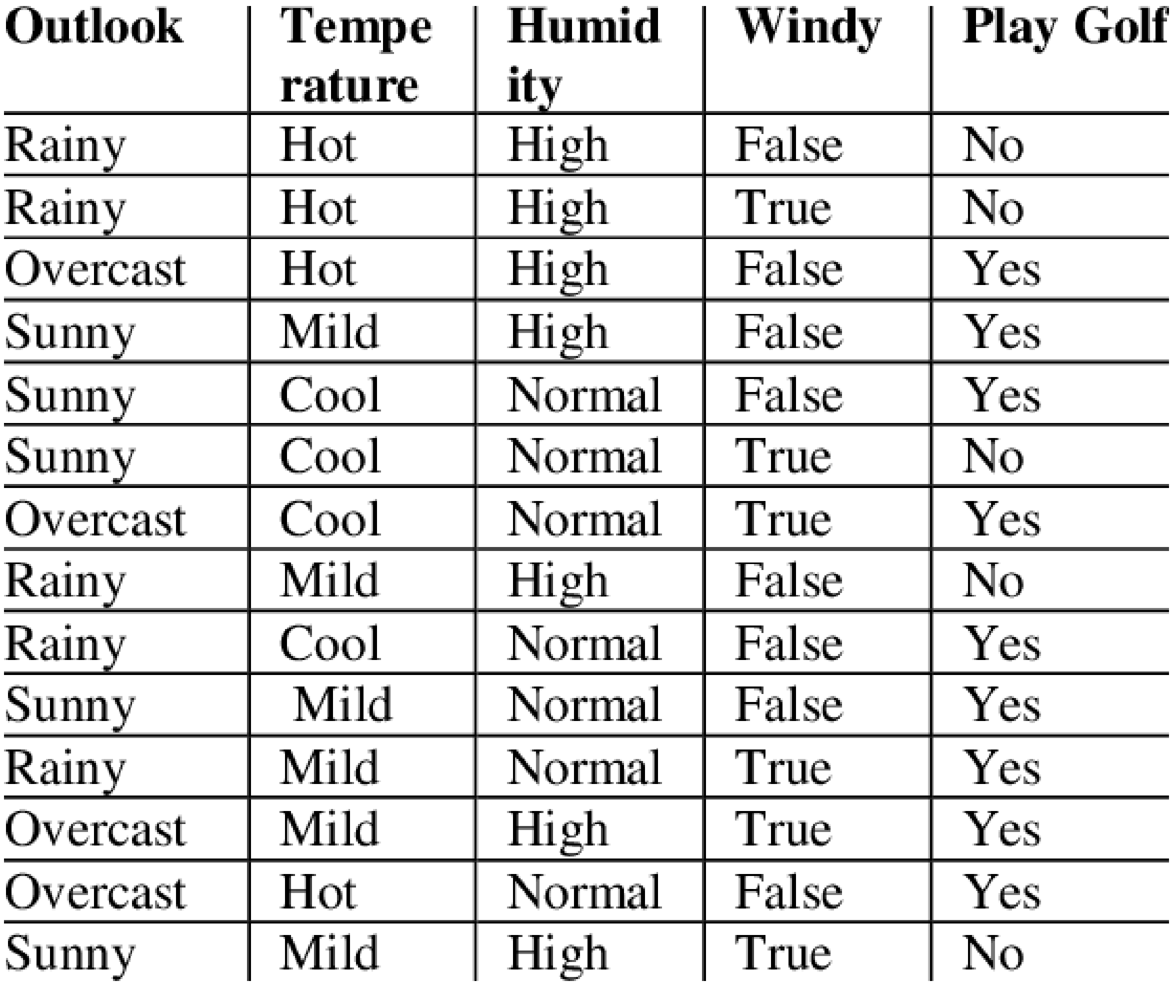


The Decision tree algorithm



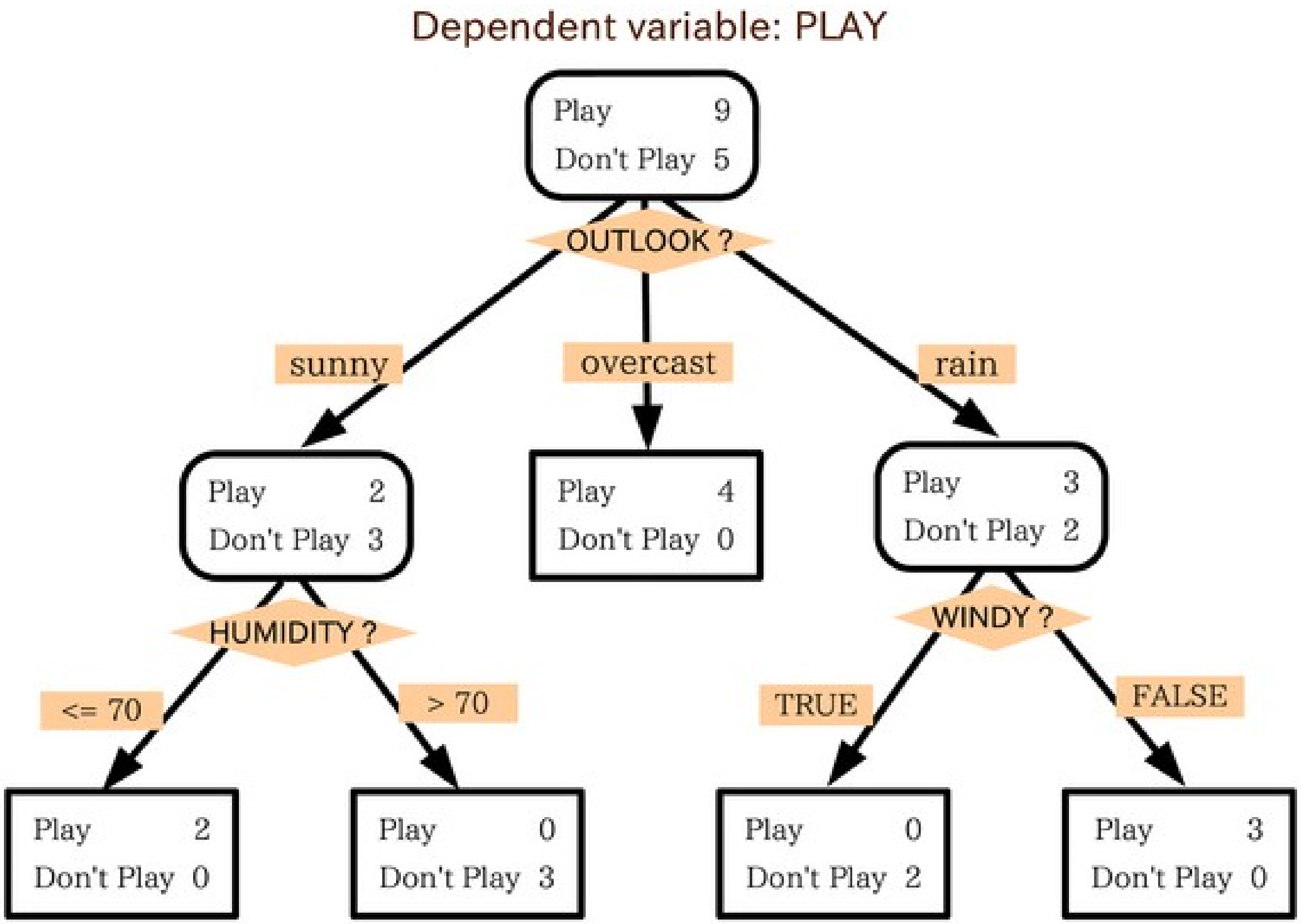
Strengths	Weaknesses
• An all-purpose classifier that does well on most problems • Highly-automatic learning process can handle numeric or nominal features, missing data • Uses only the most important features • Can be used on data with relatively few training examples or a very large number • Results in a model that can be interpreted without a mathematical background (for relatively small trees) • More efficient than other complex models	• Decision tree models are often biased toward splits on features having a large number of levels • It is easy to overfit or underfit the model • Can have trouble modeling some relationships due to reliance on axis-parallel splits • Small changes in training data can result in large changes to decision logic • Large trees can be difficult to interpret and the decisions they make may seem counterintuitive

Example:



Outlook	Temperature	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

Example:



Dependent variable: PLAY
Play 9
Don't Play 5
OUTLOOK ?
sunny
overcast
rain
Play 2
Don't Play 3
Play 4
Don't Play 0
Play 3
Don't Play 2
HUMIDITY ?
WINDY ?
<= 70
> 70
TRUE
FALSE
Play 2
Don't Play 0
Play 0
Don't Play 3
Play 0
Don't Play 2
Play 3
Don't Play 0

Terminologies Used

- Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.
- Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.
- Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.
- Branch/Sub Tree: A tree formed by splitting the tree.
- Pruning: Pruning is the process of removing the unwanted branches from the tree.
- Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

Search for a good tree

- How should you go about building a decision tree?
- The space of decision trees is too big for systematic search.
- Stop and
  - return the a value for the target feature or
  - a distribution over target feature values
- Choose a test (e.g. an input feature) to split on.
  - For each value of the test, build a subtree for those examples with this value for the test.

Top down induction

1. Which node to proceed with?

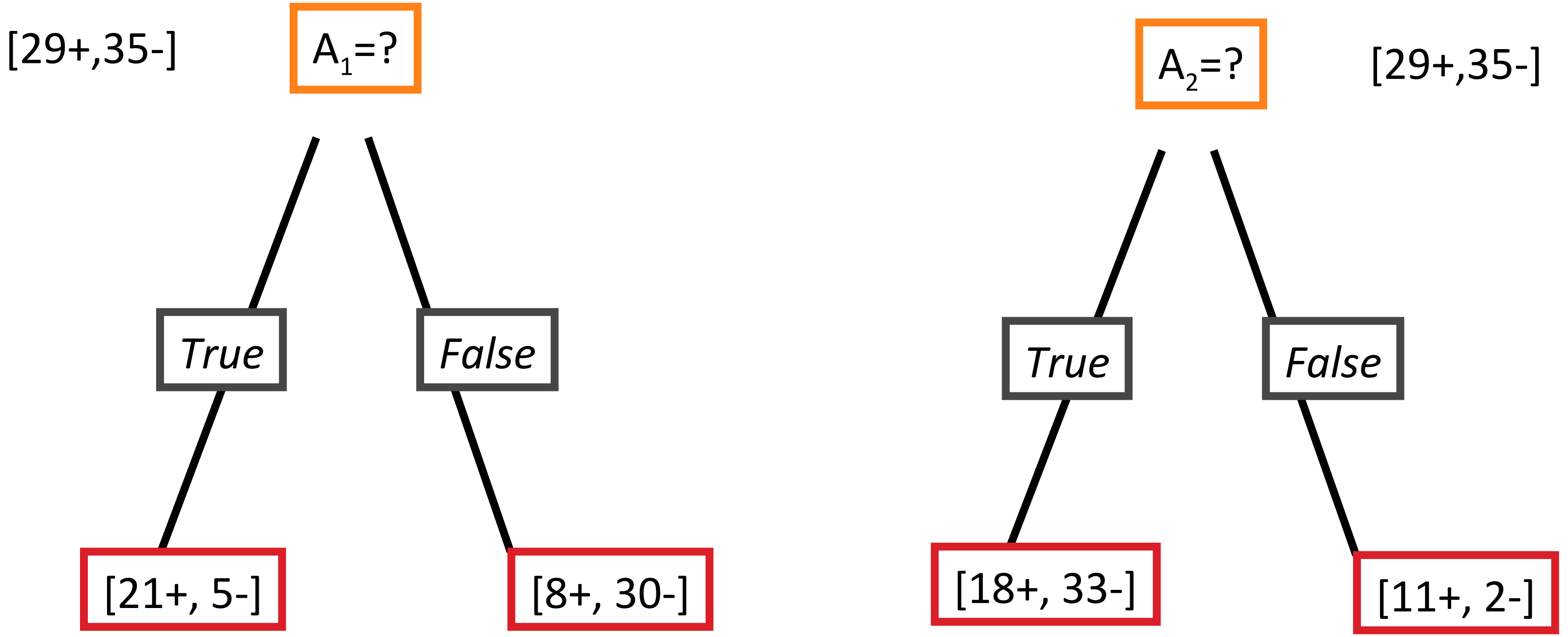
- A   the “best” decision attribute for next *node*
- Assign A as decision attribute for *node*
- For each value of A create new descendant
- Sort training examples to leaf node according to the attribute value of the branch
- If all training examples are perfectly classified (same value of target attribute) stop, else iterate over new leaf nodes.

2. When to stop?

Choices

- When to stop
  - no more input features
  - all examples are classified the same
  - too few examples to make an informative split
- Which test to split on
  - split gives smallest error.
  - With multi-valued features
  - split on all values or
  - split values into half.

Which attribute is best ?



Principle Criterion

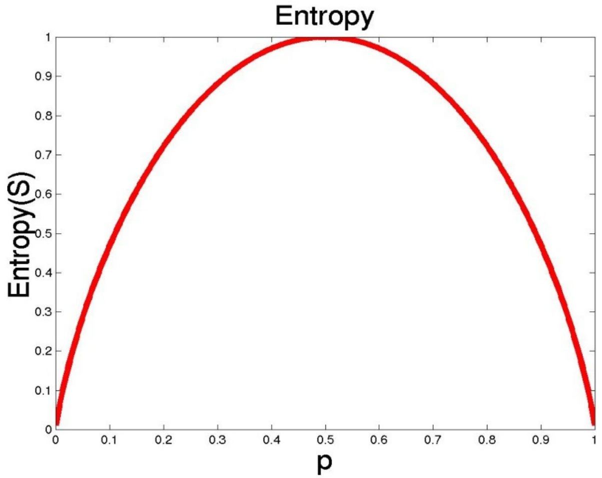
- Selection of an attribute to test at each node - choosing the most useful attribute for classifying examples.
- Information gain
  - measures how well a given attribute separates the training examples according to their target classification
  - This measure is used to select among the candidate attributes at each step while growing the tree
  - Gain is measure of how much we can reduce uncertainty (Value lies between 0,1)

Entropy

- A measure for
  - uncertainty
  - purity
  - information content
- Information theory: optimal length code assigns ($-\log_2 p$) bits to message having probability p
- S is a sample of training examples
  - p_+ is the proportion of positive examples in S
  - p_- is the proportion of negative examples in S
- Entropy of S: average optimal number of bits to encode information about certainty/uncertainty about S

  $$Entropy(S) = p_+(-\log_2 p_+) + p_-(-\log_2 p_-) = -p_+\log_2 p_+ - p_-\log_2 p_-$$

Entropy



- The entropy is 0 if the outcome is ``certain”.
- The entropy is maximum if we have no knowledge of the system (or any outcome is equally possible).

- S is a sample of training examples
- p_+ is the proportion of positive examples
- p_- is the proportion of negative examples
- Entropy measures the impurity of S

$$\text{Entropy}(S) = -p_+ \log_2 p_+ - p_- \log_2 p_-$$

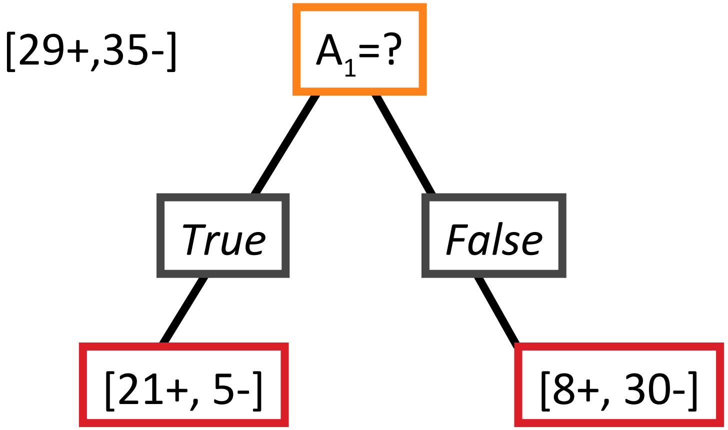
Information Gain

Gain(S,A): expected reduction in entropy due to partitioning S on attribute A

$$\text{Gain}(S,A)=\text{Entropy}(S) \quad {}_{v\ \text{values}(A)} \; |S_v|/|S|\ \text{Entropy}(S_v)$$

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

$$= 0.99$$



[29+,35-]

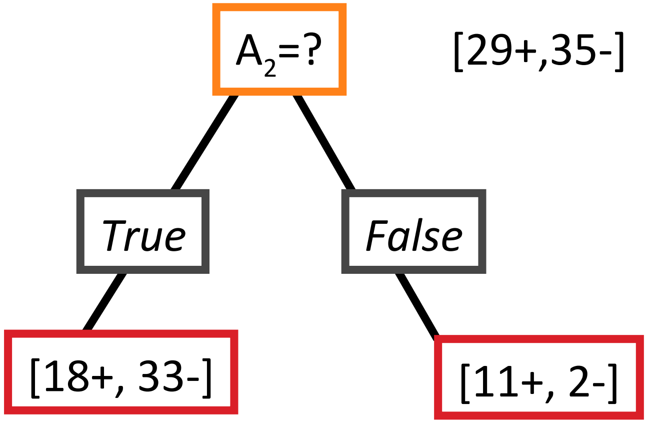
A_1=?

True

False

[21+, 5-]

[8+, 30-]



A_2=?

[29+,35-]

True

False

[18+, 33-]

[11+, 2-]

Information Gain

Entropy([21+,5-]) = 0.71

Entropy([8+,30-]) = 0.74

Gain(S,A_1)=Entropy(S)

-26/64*Entropy([21+,5-])

-38/64*Entropy([8+,30-])

=0.27

Entropy([18+,33-]) = 0.94

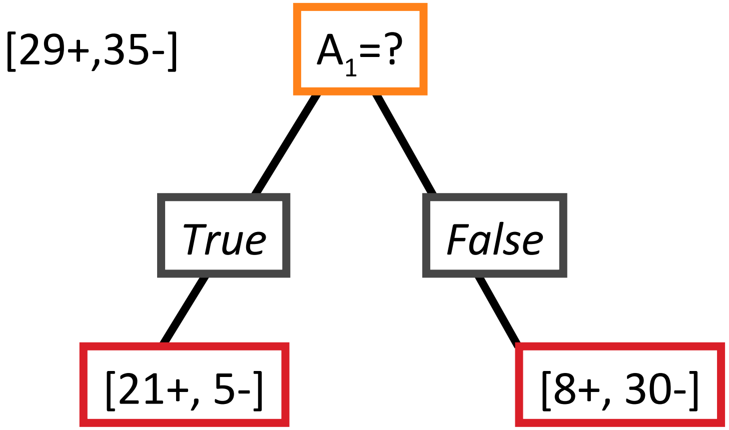
Entropy([8+,30-]) = 0.62

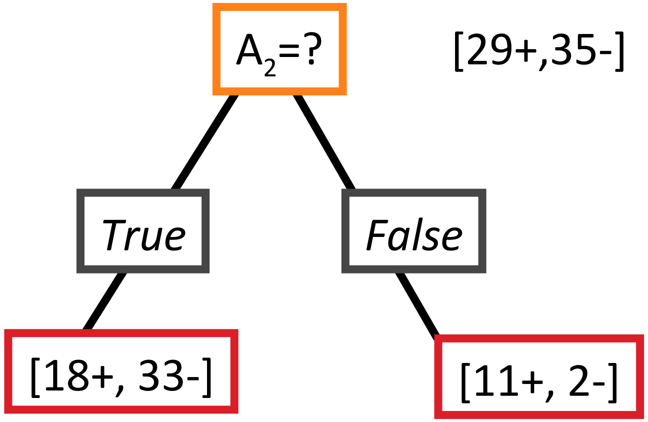
Gain(S,A_2)=Entropy(S)

-51/64*Entropy([18+,33-])

-13/64*Entropy([11+,2-])

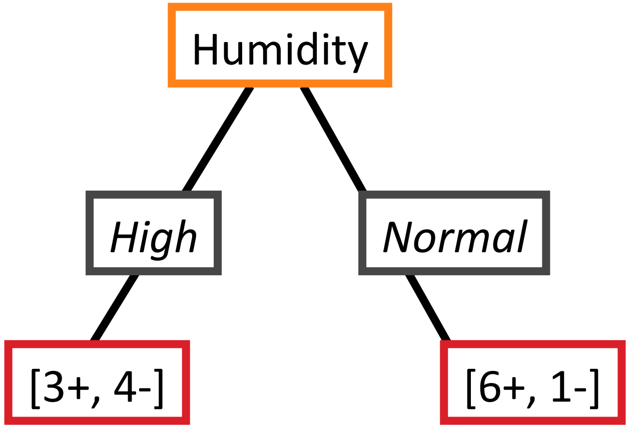
=0.12





Selecting next attribute

S=[9+,5-]
E=0.940



Humidity

High

Normal

[3+, 4-]

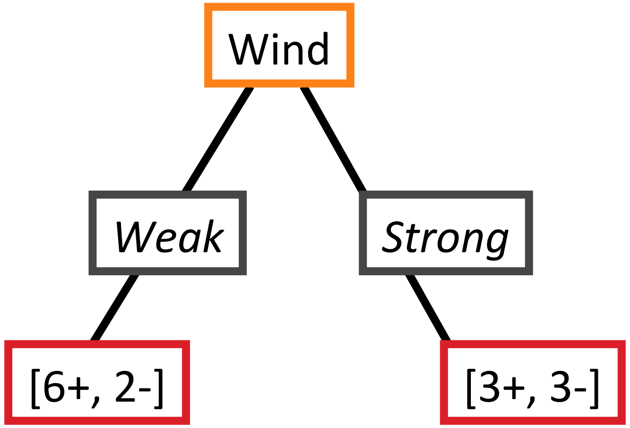
[6+, 1-]

E=0.985

E=0.592

Gain(S,Humidity)
=0.940-(7/14)*0.985
– (7/14)*0.592
=0.151

S=[9+,5-]
E=0.940



Wind

Weak

Strong

[6+, 2-]

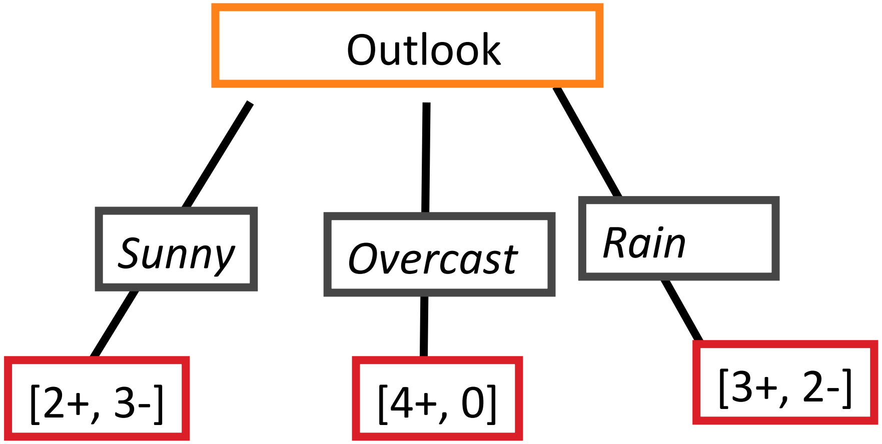
[3+, 3-]

Gain(S,Wind)
=0.940-(8/14)*0.811
– (6/14)*1.0
=0.048

Humidity provides greater info. gain than Wind, w.r.t target classification.

Selecting next attribute

S=[9+,5-]
E=0.940



E=0.971 E=0.0 E=0.971

Gain(S,Outlook)
=0.940-(5/14)*0.971
-(4/14)*0.0 – (5/14)*0.0971
=0.247

Selecting next attribute

The information gain values for the 4 attributes are:

- Gain(S,Outlook) =0.247
- Gain(S,Humidity) =0.151
- Gain(S,Wind) =0.048
- Gain(S,Temperature) =0.029

where S denotes the collection of training examples

Note: 0Log$_2$0 =0

Resources

- https://stackabuse.com/
- http://people.sc.fsu.edu
- https://www.geeksforgeeks.org
- http://scikit-learn.org/
- https://machinelearningmastery.com

Thank you

@mitu_skillologies

/mITuSkillologies

@mitu_group

/company/mitu-skillologies

MITUSkillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

contact@mitu.co.in

tushar@tusharkute.com