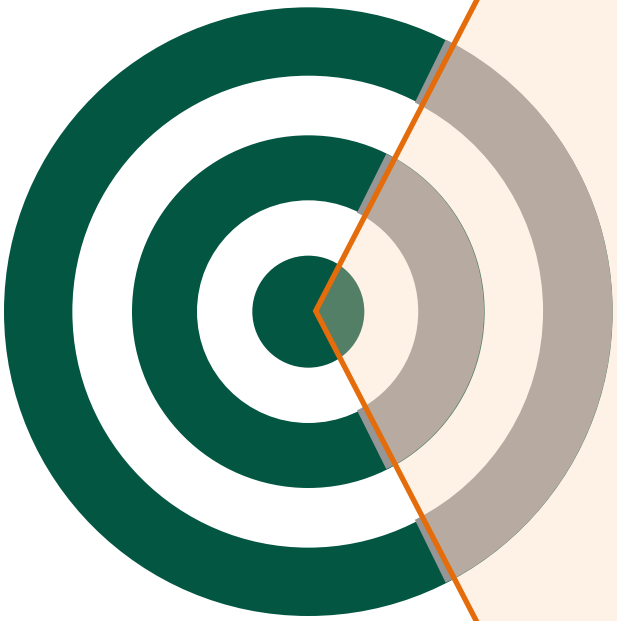
Decision Trees

Agenda

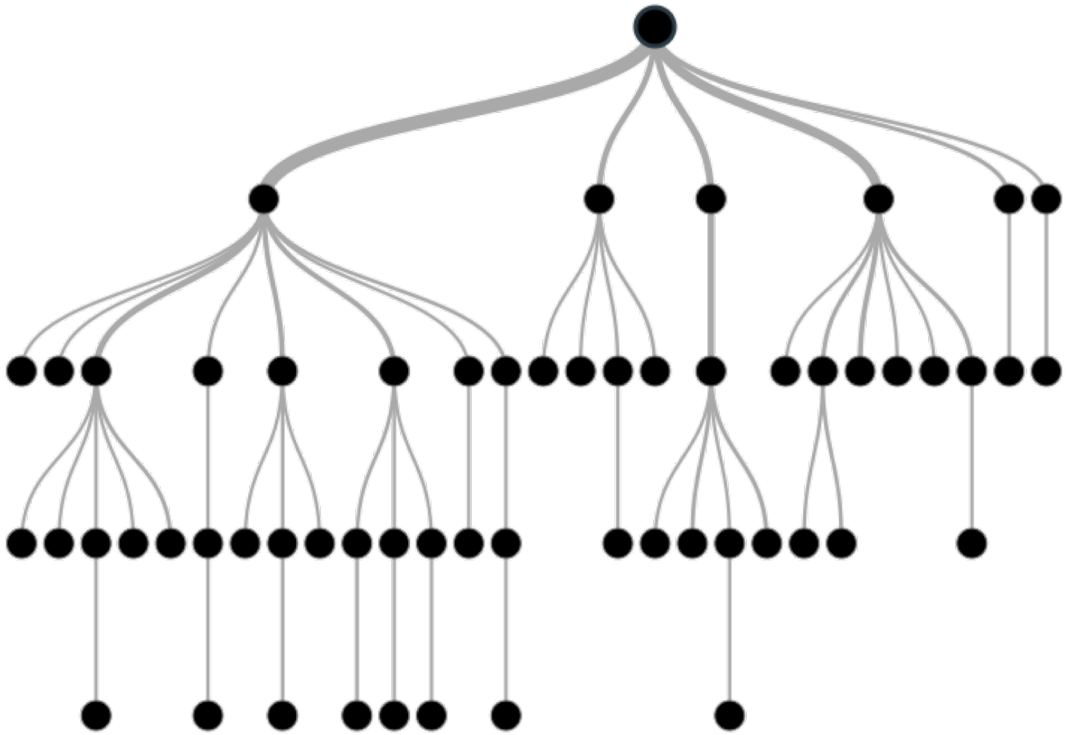


In this session, you will learn about:

- Introduction
- Requirements
- Strengths and Weakneses
- Random Forest
- Information Gain of the Attribute
- Entropy

Introduction

- Decision trees are powerful and popular tools for classification and prediction.
- Decision trees represent *rules*, which can be understood by humans and used in knowledge system such as database.

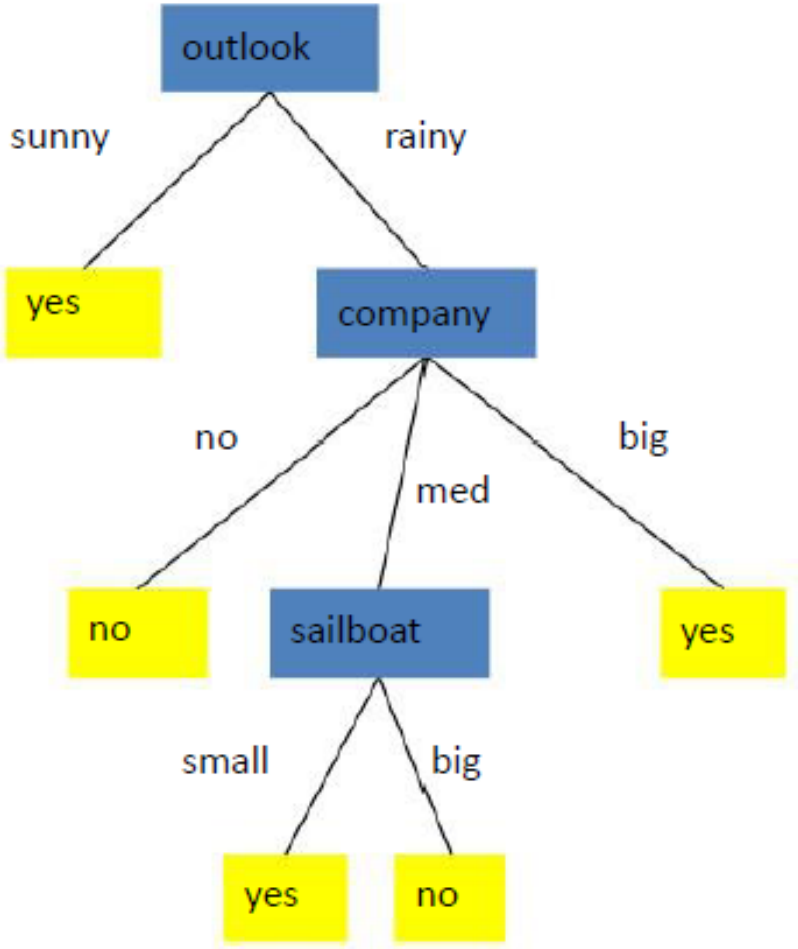


Requirements

Attribute-value Description	Predefined Classes	Sufficient Data
Object or case must be expressible in terms of a fixed collection of properties or attributes E.g.: Hot, Mild, Cold	The target function has discrete output values Bollean or multiclass	Enough training cases should be provided to learn the model.

Example

Sr.	Attribute			Class
No.	Outlook	Company	Sailboat	Sail
1	Sunny	Big	Small	Yes
2	Sunny	Med	Small	Yes
3	Sunny	Med	Big	Yes
4	Sunny	No	Small	Yes
5	Sunny	Big	Big	Yes
6	Rainy	No	Small	No
7	Rainy	Med	Small	Yes
8	Rainy	Big	Big	Yes
9	Rainy	No	Big	No
10	Rainy	Med	Big	No



Decision Tree

Decision tree is a classifier in the form of a tree structure

Decision Node

- Specifies a test on a single attribute

Leaf Node

- Indicates the value of the target attribute

Arc/edge

- Split of one attribute

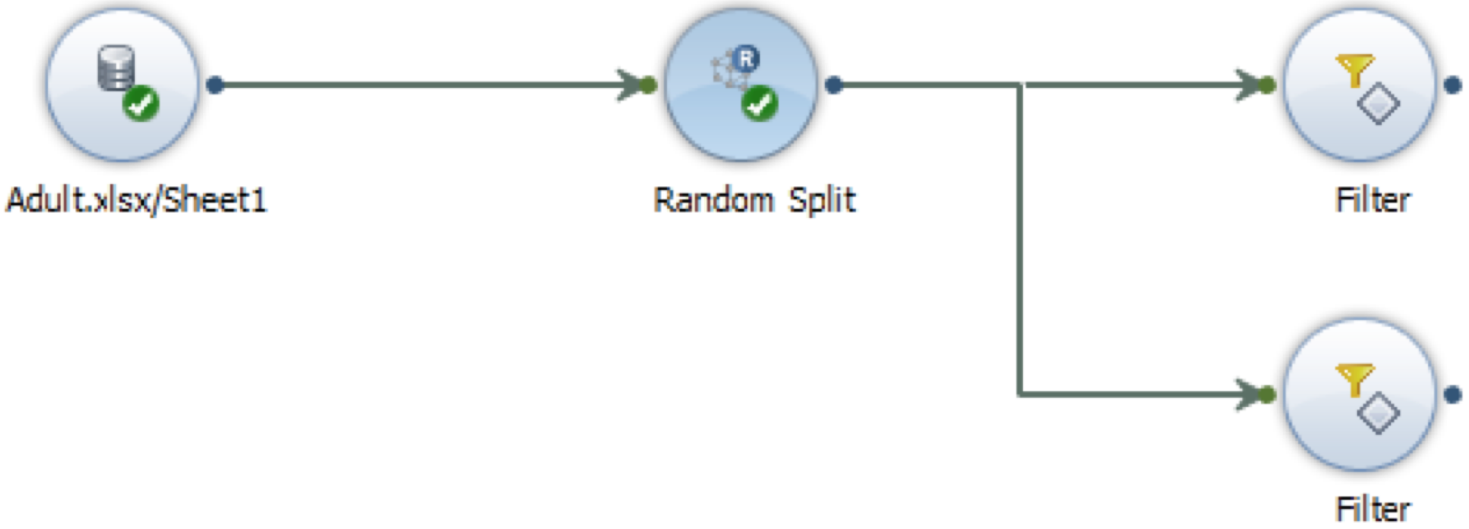
Path

- A Disjunction of test to make the final decision

Decision trees classify instances or examples by starting at the root of the tree and moving through it until a leaf node.

Random Split

- The tree can grow huge.
- These trees are hard to understand.
- Larger trees are typically less accurate than smaller trees.



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

Entropy

- A measure of homogeneity of the set of examples.
- Given a set S of positive and negative examples of some target concept (a 2-class problem), the entropy of set S relative to this binary classification is

$$E(S) = - p(P)log2\ p(P) - p(N)log2\ p(N)$$

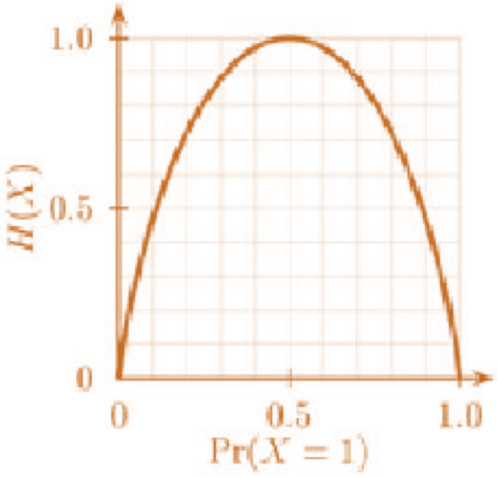
Entropy

- Suppose S has 25 examples, 15 positive and 10 negatives [15+, 10-].
- Then the entropy of S relative to this classification is

$$E(S)=-(15/25)\ log2(15/25) - (10/25)\ log2\ (10/25)$$

Some Intuitions

- 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)



Entropy of a 2-class problem with regard to the portion of one of the two groups

Information Gain

- To classify an object, a certain information is needed
  - I, information
- After we have learned the value of attribute A, we only need some remaining amount of information to classify the object
  - Ires, residual information
- Gain
  - Gain(A) = I - Ires(A)
- The most 'informative' attribute is the one that minimizes Ires, i.e., maximizes Gain

Residual Information

- After applying attribute A, S is partitioned into subsets according to values v of A
- Ires is equal to weighted sum of the amounts of information for the subsets

$$I_{res} = -\sum_{v} p(v) \sum_{c} p(c|v) \log_2 p(c|v)$$

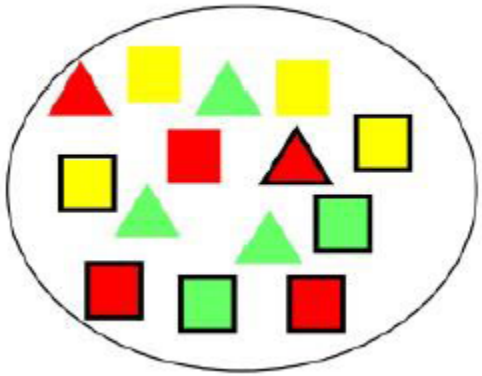
Triangles and Squares

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange

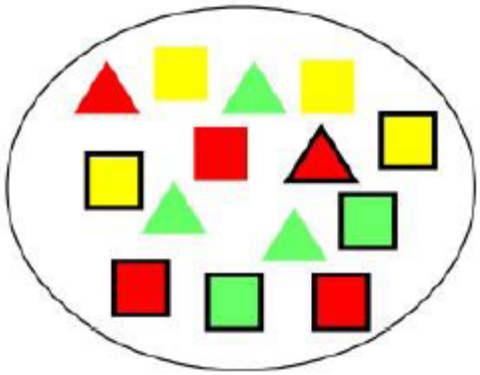
Triangles and Squares

#	Attribute			Shape
	Color	Outline	Dot	
1	green	dashed	no	triange
2	green	dashed	yes	triange
3	yellow	dashed	no	square
4	red	dashed	no	square
5	red	solid	no	square
6	red	solid	yes	triange
7	green	solid	no	square
8	green	dashed	no	triange
9	yellow	solid	yes	square
10	red	solid	no	square
11	green	solid	yes	square
12	yellow	dashed	yes	square
13	yellow	solid	no	square
14	red	dashed	yes	triange

Data Set
A Set of Classified Objects



Entropy (1)



- 5 Triangles
- 9 Squares
- Class Probabilities

$$p(\square) = \frac{9}{14}$$

$$p(\triangle) = \frac{5}{14}$$

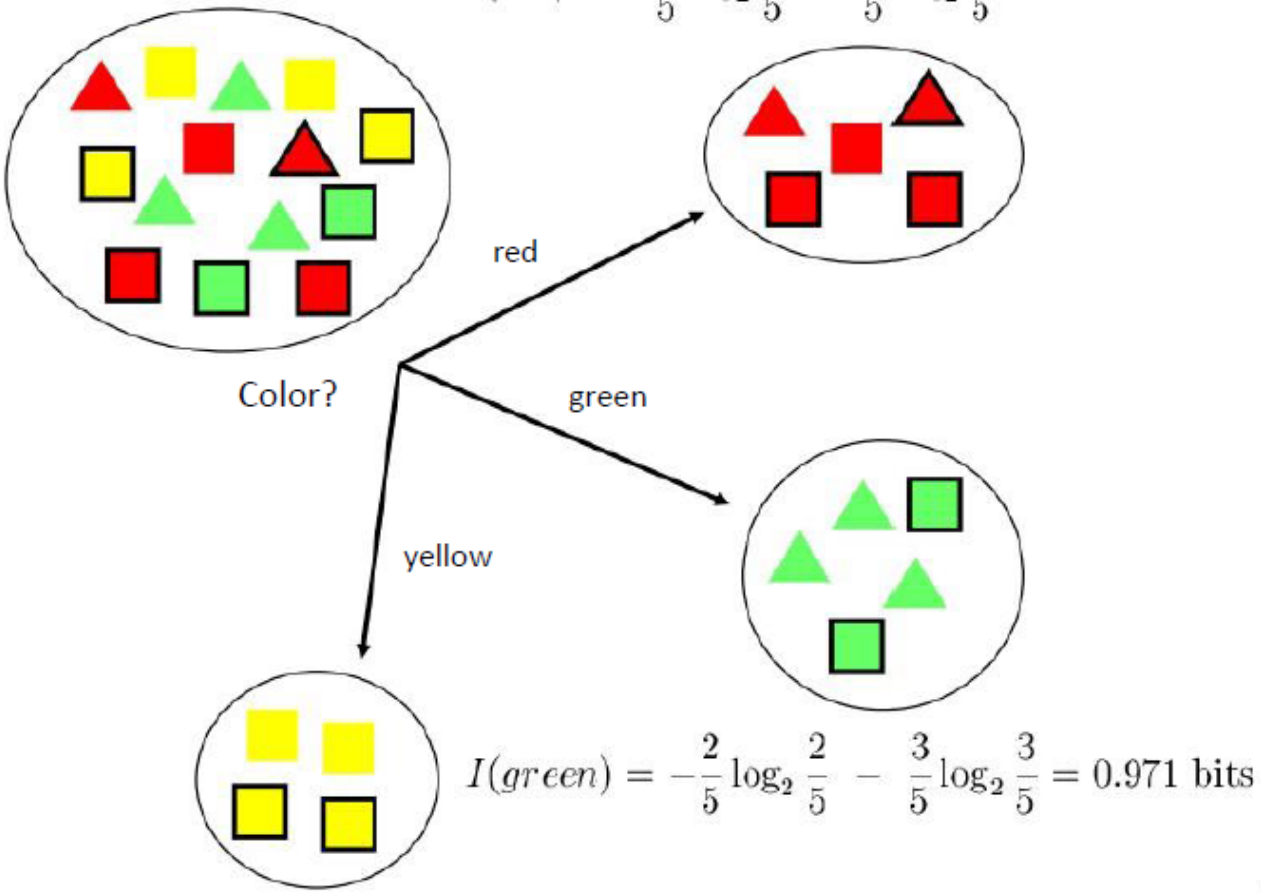
- Entropy

$$I = -\frac{9}{14}\log_2\frac{9}{14} - \frac{5}{14}\log_2\frac{5}{14} = 0.940 \text{ bits}$$

Entropy (2)

Entropy reduction by data set partitioning

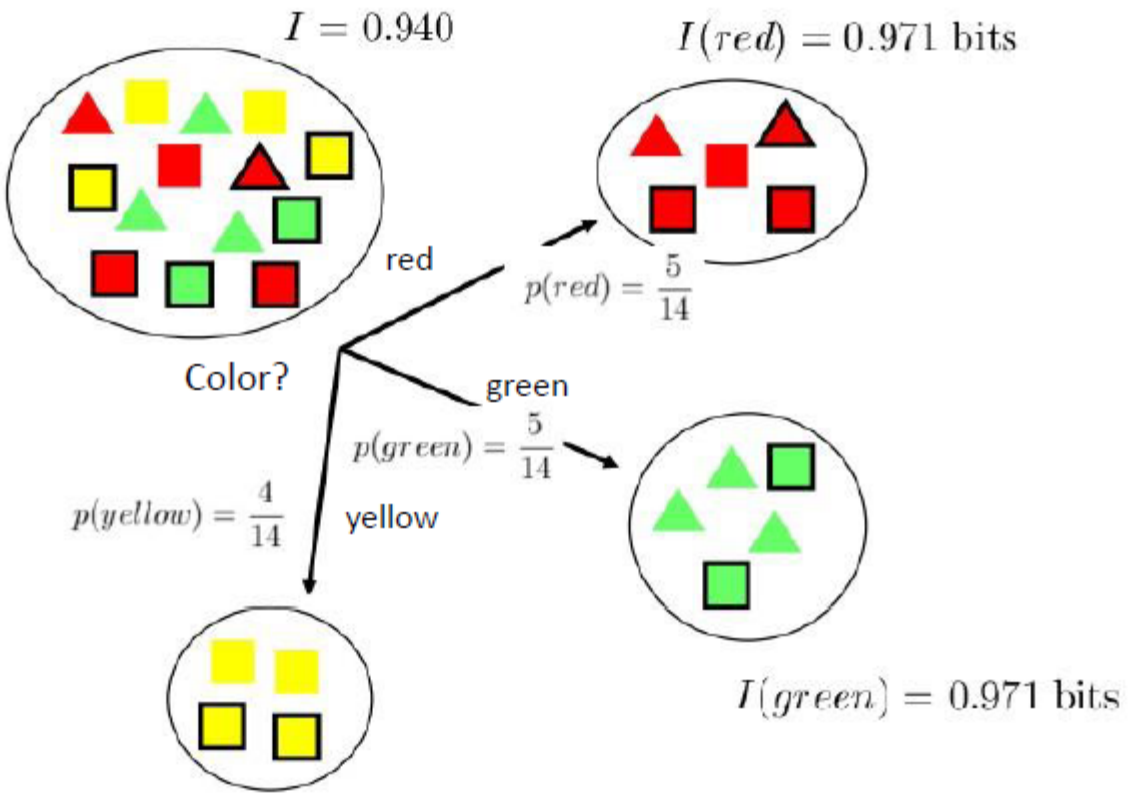
$$I(red) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} = 0.971 \text{ bits}$$



$$I(green) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} = 0.971 \text{ bits}$$

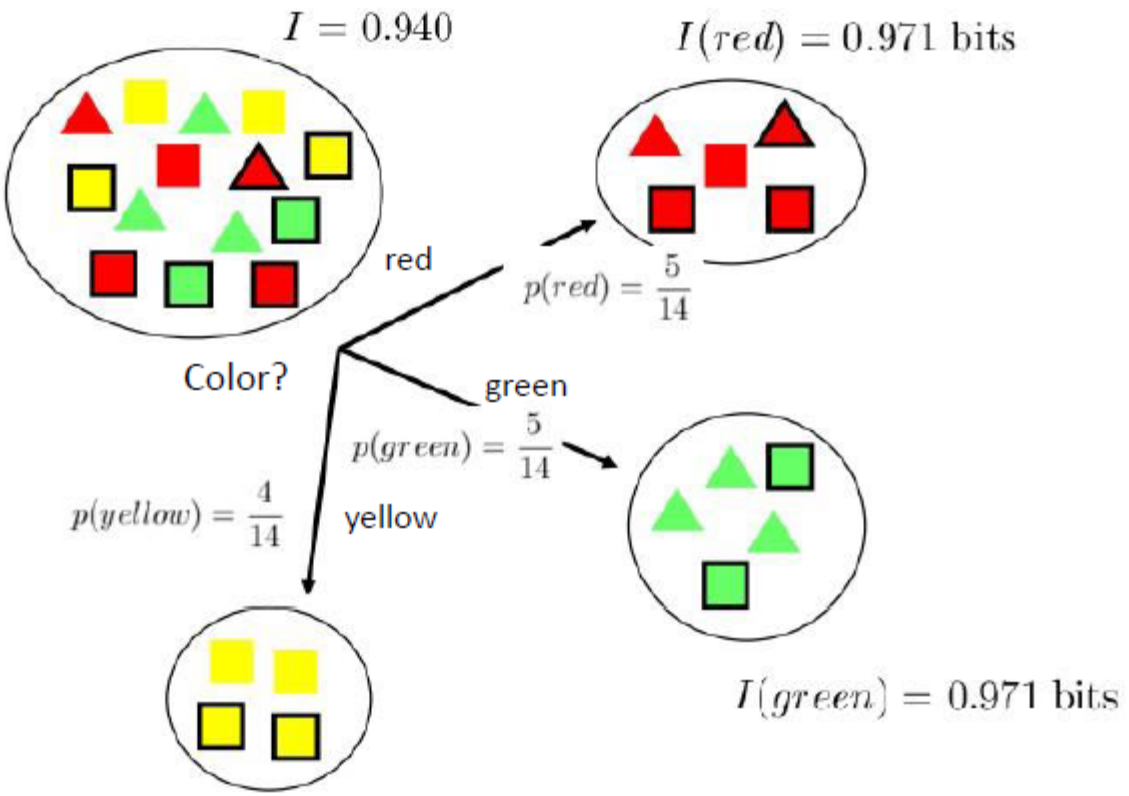
$$I(yellow) = -\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4} = 0.0 \text{ bits}$$

Entropy (3)



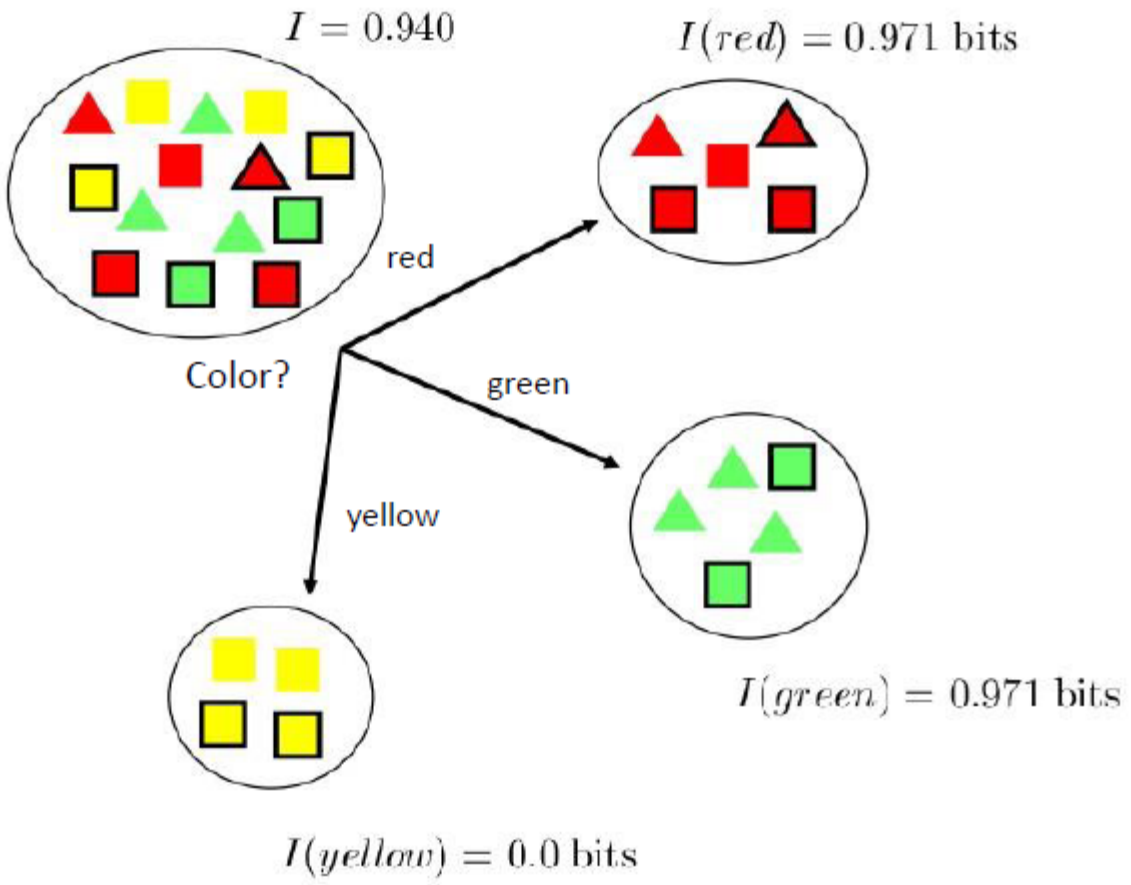
$$I_{res}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694 \ bits$$

Entropy (4)



$$I_{res}(\text{Color}) = \sum p(v)I(v) = \frac{5}{14}0.971 + \frac{5}{14}0.971 + \frac{4}{14}0.0 = 0.694 \ bits$$

Entropy (5)

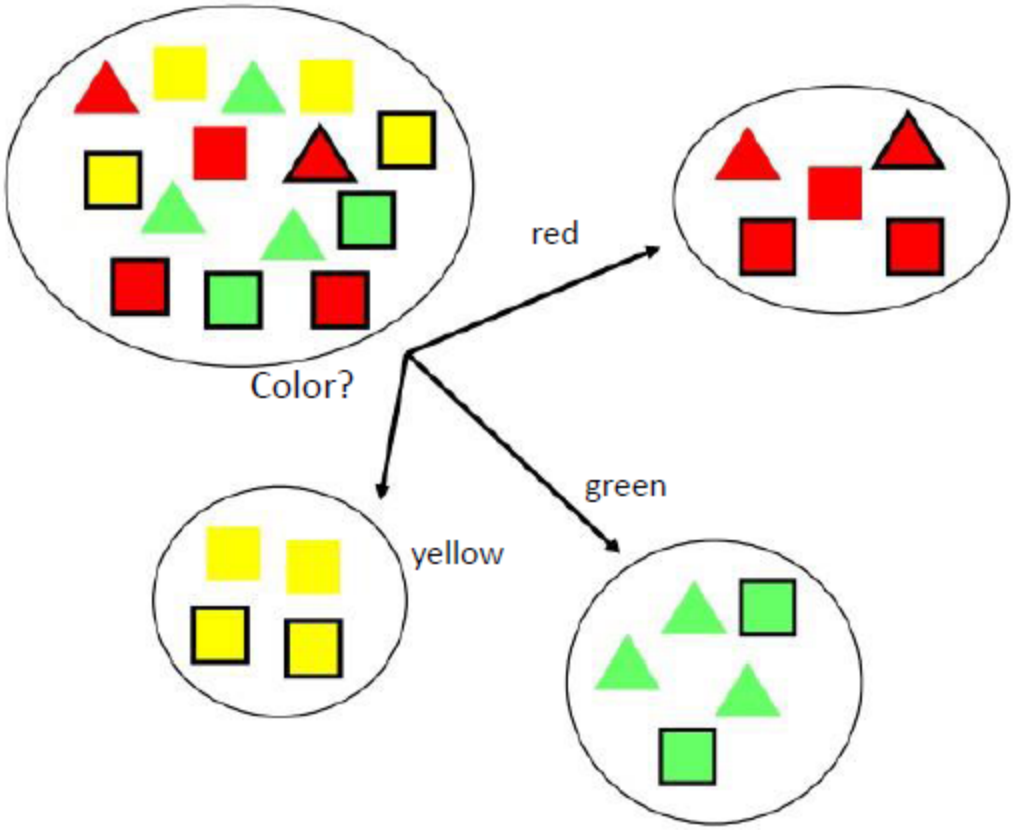


$$Gain(\text{Color}) = I - I_{res}(\text{Color}) = 0.940 - 0.694 = 0.246\ bits$$

Information Gain of The Attribute

- Attributes
  - Gain(Color) = 0.246
  - Gain(Outline) = 0.151
  - Gain(Dot) = 0.048
- Heuristics: attribute with the highest gain is chosen

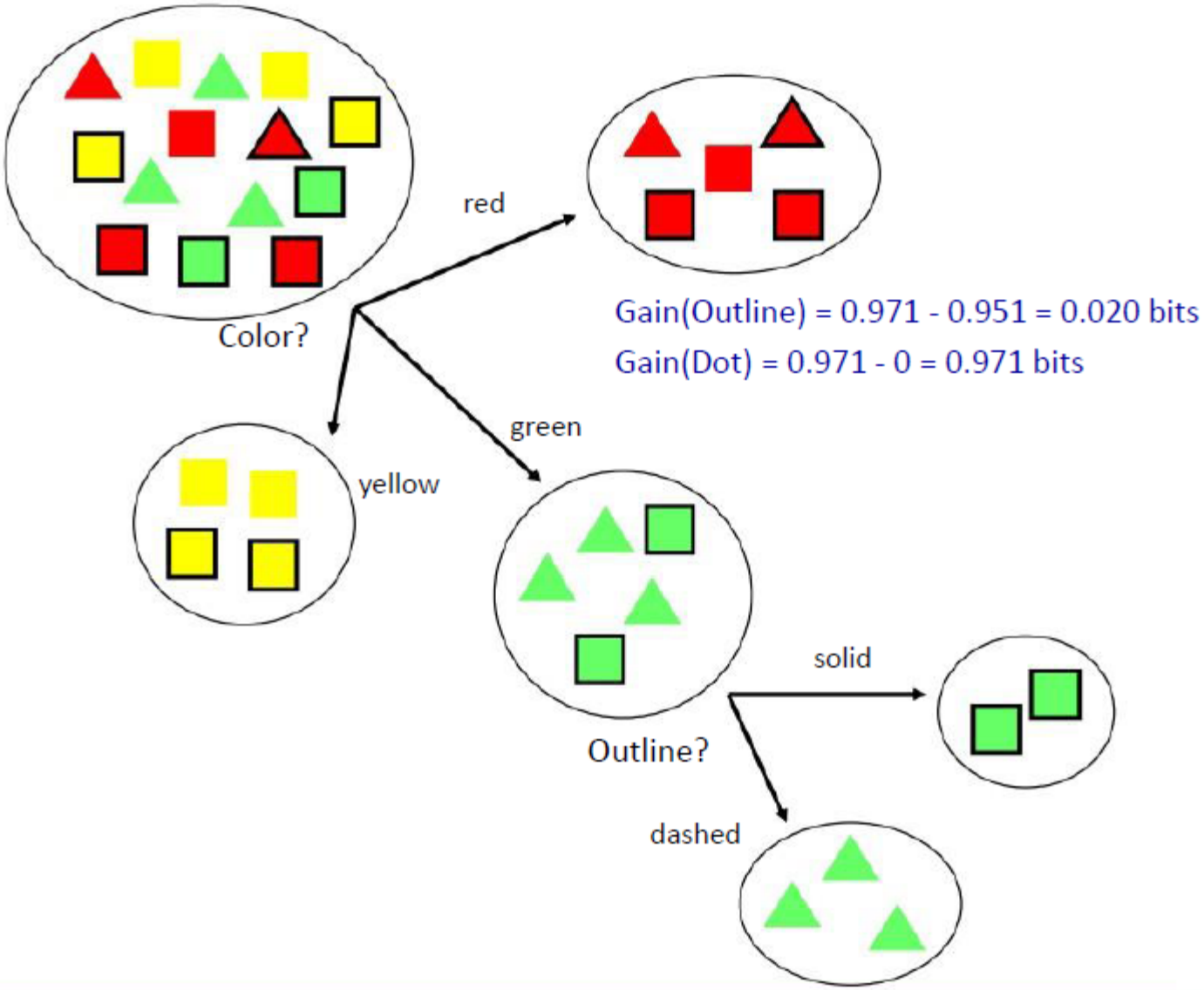
Information Gain of The Attribute



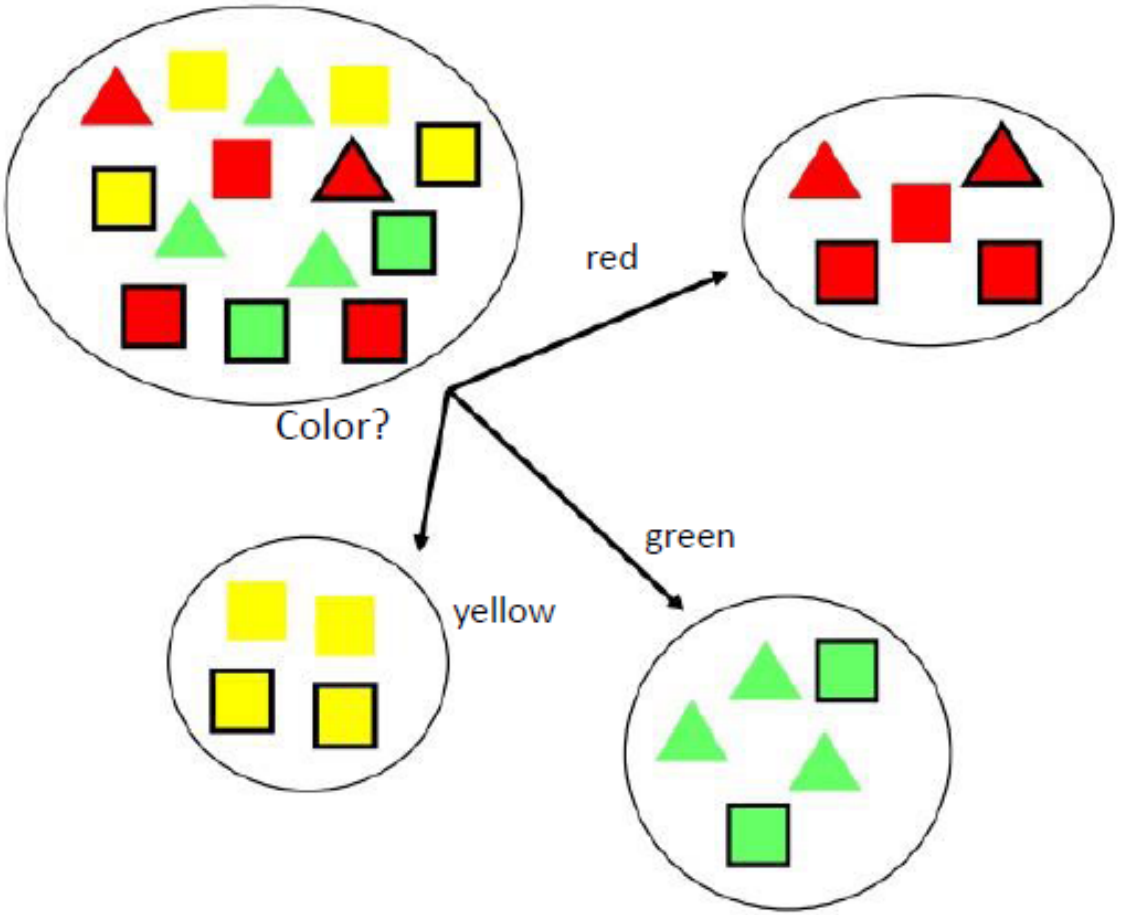
Gain(Outline) = 0.971 - 0 = 0.971 bits

Gain(Dot) = 0.971 - 0.951 = 0.020 bits

Information Gain of the Attribute



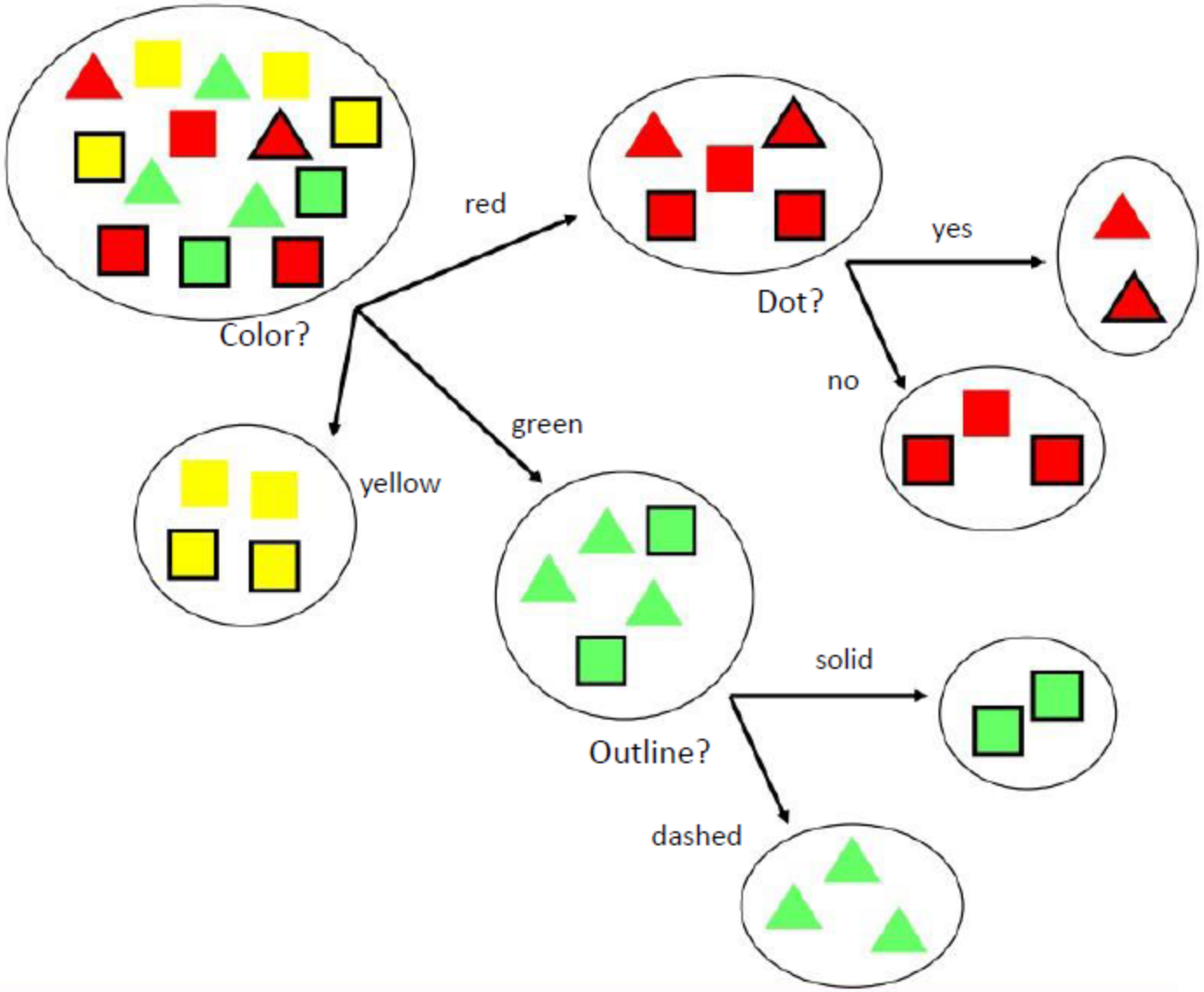
Information Gain of the Attribute



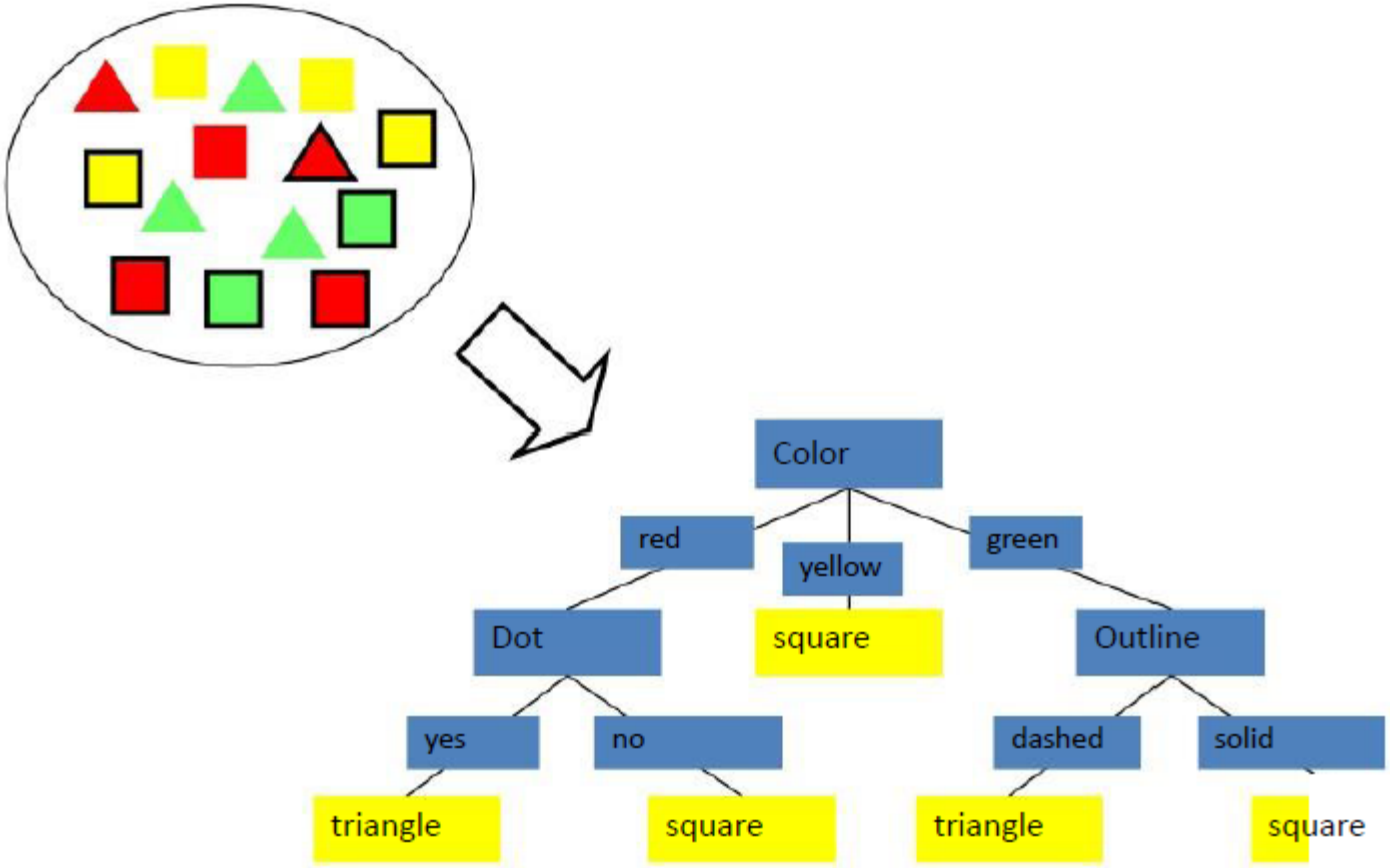
Gain(Outline) = 0.971 - 0 = 0.971 bits

Gain(Dot) = 0.971 - 0.951 = 0.020 bits

Information Gain of the Attribute



Decision Tree



Continuous Attribute?

- Each non-leaf node is a test, its edge partitioning the attribute into subsets (easy for discrete attribute).
- For continuous attribute
  - Partition the continuous value of attribute A into a discrete set of intervals
  - Create a new boolean attribute Ac , looking for a threshold c,

$$A_c = \begin{cases} true & \text{if } A < c \\ false & \text{otherwise} \end{cases}$$

How to choose c ?

Strength

- Can generate understandable rules
- Perform classification without much computation
- Can handle continuous and categorical variables
- Provide a clear indication of which fields are most important for prediction or classification

Weaknesses

- Not suitable for prediction of continuous attribute.
- Perform poorly with many class and small data.
- Computationally expensive to train.
  - At each node, each candidate splitting field must be sorted before its best split can be found.
  - In some algorithms, combinations of fields are used and a search must be made for optimal combining weights.
  - Pruning algorithms can also be expensive since many candidate sub- trees must be formed and compared.

Pruning

- Pruning is a technique in machine learning that reduces the size of decision trees by removing sections of the tree that provide little power to classify instances.
- The dual goal of pruning is
  - Reduced complexity of the final classifier
  - Better predictive accuracy by the reduction of over fitting and removal of sections of a classifier that may be based on noisy or erroneous data.

Random Forest

Ensemble Classifiers

Combining Classifiers

- Using different subset of training data with a single learning method
- Using different training parameters with a single training method (e.g. using different initial cluster centroids k-means clustering)
- Using different learning methods

Random Forest

- Random forest (or random forests) is an ensemble classifier that consists of many decision trees and outputs the class that is the mode of the class's output by individual trees.
- The term came from random decision forests that was first proposed by Tin Kam Ho of Bell Labs in 1995.
- The method combines Breiman's "bagging" idea and the random selection of features.

Decision Trees

- Decision trees are individual learners that are combined. They are one of the most popular learning methods commonly used for data exploration.
- One type of decision tree is called CRT… classification and regression tree.

Algorithm

Each Tree is Constructed Using this Algorithm

1. Let the number of training cases be *N*, and the number of variables in the classifier be *M*.
2. We are told the number *m* of input variables to be used to determine the decision at a node of the tree; *m* should be much less than *M*.
3. Choose a training set for this tree by choosing *n* times with replacement from all *N* available training cases (i.e. **take a bootstrap sample**). Use the rest of the cases to estimate the error of the tree, by predicting their classes.
4. For each node of the tree, randomly choose *m* variables on which to base the decision at that node. Calculate the best split based on these *m* variables in the training set.
5. Each tree is fully grown and not pruned (as may be done in constructing a normal tree classifier).

- For prediction a new sample is pushed down the tree.
- It is assigned the label of the training sample in the terminal node it ends up in.
- This procedure is iterated over all trees in the ensemble, and the average vote of all trees is reported as random forest prediction.

Tree Bagging

- The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners.
- Given a training set $X = x_1, \ldots, x_n$ with responses $Y = y_1, \ldots, y_n$, bagging repeatedly selects a random sample with replacement of the training set and fits trees to these samples

Decision Trees

- For b = 1, …, B:
  - Sample, with replacement, n training examples from X, Y; call these Xb, Yb.
  - Train a decision or regression tree fb on Xb, Yb.
  - After training, predictions for unseen samples x' can be made by averaging the predictions from all the individual regression trees on x':
- The number of samples/trees, B, is a free parameter; can be found using cross-validation for a dataset with p features, √p features are used in each split.

$$\hat{f} = \frac{1}{B}\sum_{b=1}^{B} \hat{fb}(x')$$

Features and Advantages

The Advantages of Random Forest

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.
- Generated forests can be saved for future use on other data.
- Prototypes are computed that give information about the relation between the variables and the classification.
- It computes proximities between pairs of cases that can be used in clustering, locating outliers, or (by scaling) give interesting views of the data.

IMARTICUS LEARNING

Thank you

Mumbai | Bangalore | Pune | Chennai | Jaipur

ACCREDITED TRAINING PARTNER:

CISI CHARTERED INSTITUTE FOR SECURITIES & INVESTMENT

IIBA International Institute of Business Analysis