LEARNING

[*Through Classification*]

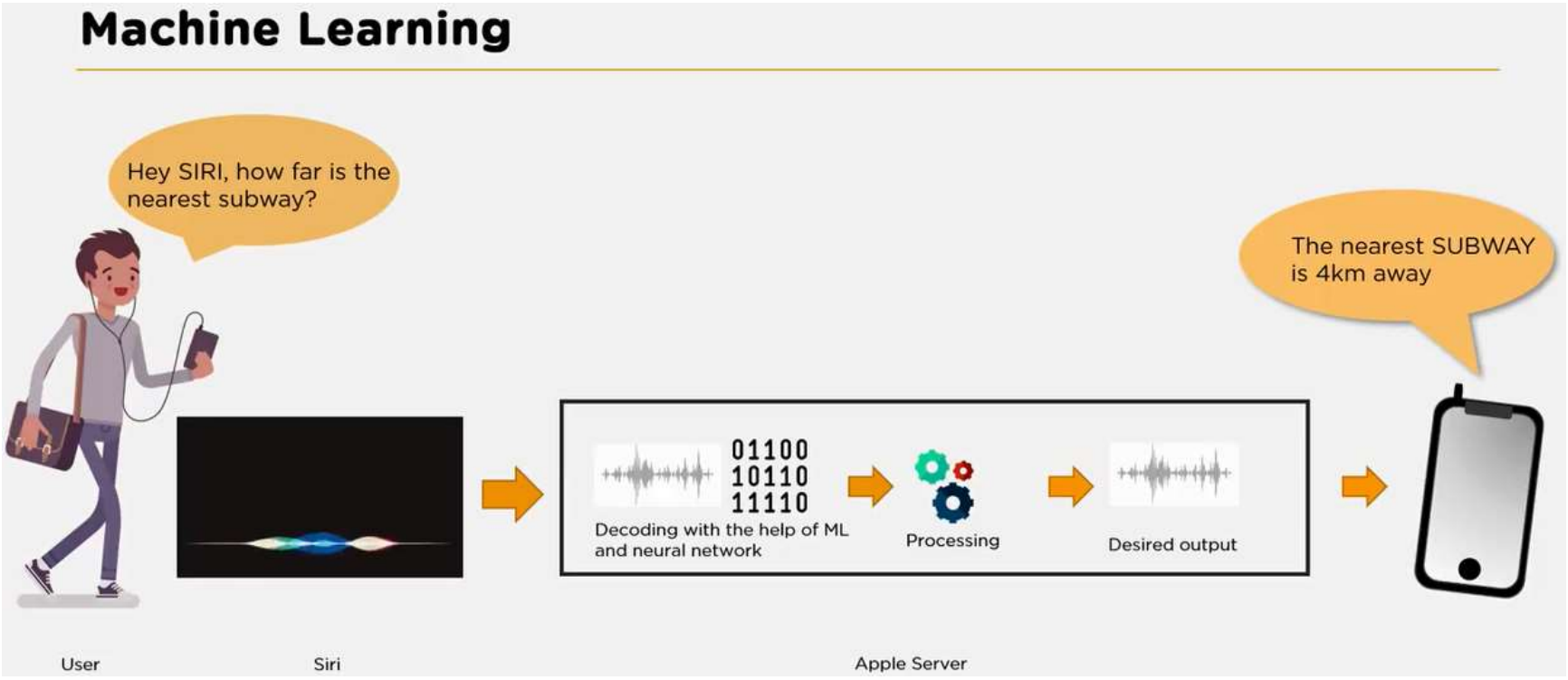
What is Learning?

- Knowledge acquired through study, experience, or being taught.
- **Machine learning** is an application of **artificial intelligence** (**AI**) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.
- **Machine learning** focuses on the development of computer programs that can access data and use it learn for themselves.

Why Learning is important

- Learning is **important** because it enables human capabilities – understanding, reasoning, planning, communication and perception – to be undertaken by software increasingly effectively, efficiently and at low cost.

Machine Learning (ML)



Machine Learning (ML)

What is Machine Learning?

Machine Learning is the science of making computers learn and act like humans by feeding data and information without being explicitly programmed!

Data is processed

System Learns

Machine Learning makes predictions and decisions based on past data

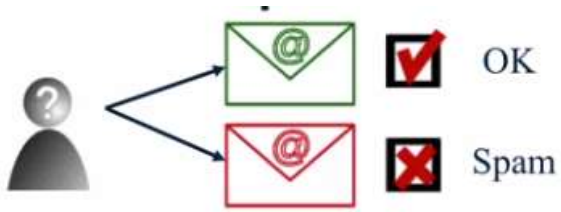
Why ML?

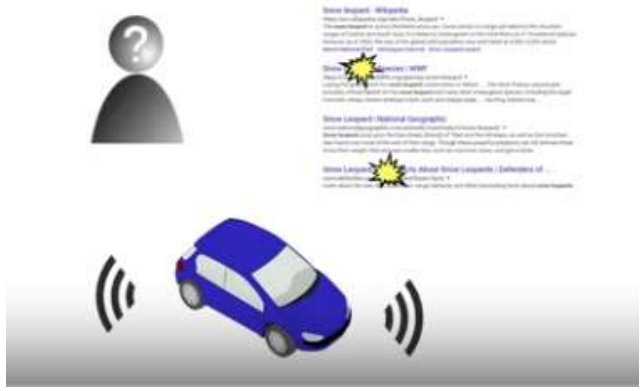
- When a computing problem needs to be solved – say, storing and retrieving data from a database – you try to write a program that manually specifies a series of programming steps that need to be run to solve that particular problem. Works great for the most of CS problems, but not all
- E.g. Speech recognition – A feature now commonly available on tech gadgets
  - How to write down a set of rules in a language for accurately converting human speech to text?
  - How to understand the complex human speech (huge variety of pronunciations, vocabulary, accents).
  - How to customize to recognize new words / features (write a whole new set of rules?).
- Machine Learning,
  - **Technology that allows us to automatically learn complex rules efficiently from labelled examples, called, training data,** (in more accurate and flexible way)
  - Meets the primary goal of being able to generalize, to correctly predict or recognize new objects that weren't seen during training

So the basic problem of machine learning is to explore how computers can program themselves to perform a task, and to improve their performance automatically as they gain more experience.

Machine Learns from data

- Experience for machine is Data!
- Data that takes various form and formats in different situations,
- Labelled examples that are used to train
  E.g. Email spam detection.
- Feedback from the user.
  E.g. A search engine tracks clicks on pages
- System getting data from the surrounding environment over time.
  E.g. Self-driving cars detecting nearby objects and events and learning to move reliably.





Machine Learning - Everywhere

Machine Learning algorithms are now involved in more and more aspects of everyday life.


Facial Recognition


Spam Detection


Recommendations


Medical Diagnosis


Smart Cars


Fraud Detection


Online Search


Speech

From what you read and watch,
to how you shop,
to who you meet
and how you travel.

Fraud detection

- Every time you buy something using a credit card, machine learning algorithms immediately check your purchase to verify whether or not this might be a fraudulent transaction.
- ML algorithms look at features transaction
  - Time
  - Location
  - Amount.
- Then make prediction of fraudulent or safe based on consistency with features of your previous purchases.
- The system also records and learns from any user feedback on whether the transaction was in fact fraudulent.

Search and recommendation systems

Search and recommendation systems are huge area of application for machine learning. Commercial search engines use ML starting with the moment you begin typing in a query.

- An ML algorithm might monitor your keystrokes to predict the best queries to auto complete while you're typing.
- An ML algorithm will determine the selection and ranking of the webpages you see for that query.
- An ML algorithm will determine which ads you see on the page or which related queries the system suggests, etc.
- Search engines also use data about how you interact with the search site, such as which pages you click, how long you read the pages to improve their future effectiveness.

Movie/Hotel recommendation

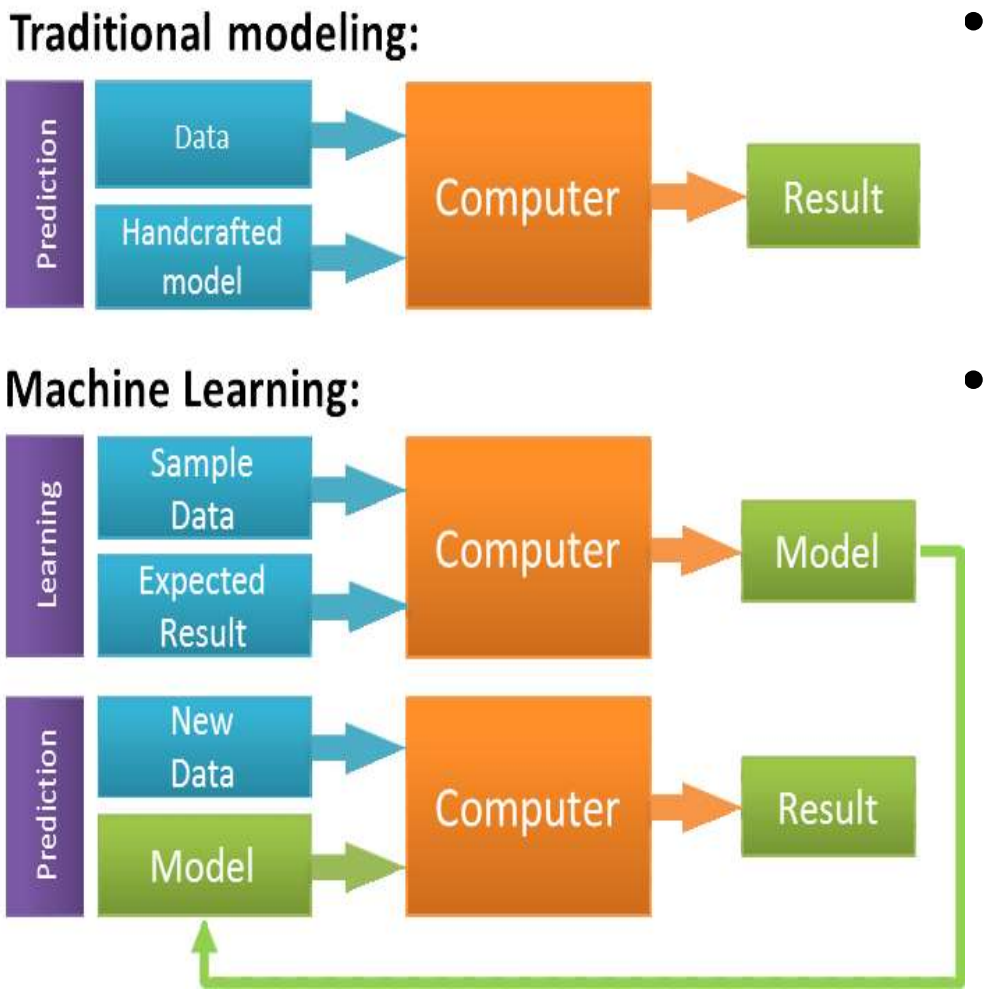
Over 80% of what people watch on Netflix comes on recommendations

Movie recommendation sites use machine learning algorithms to model what movie to suggest.

- What you liked in your past reviews
- Your interaction patterns with the site
- How your preferences relate to those of other users.

It uses all this data to learn a model for your personal taste to give you better choices and keep you engaged with the site/lead you to make more movie purchases over time.

How ML differs from Traditional(Rule) based Programming ?

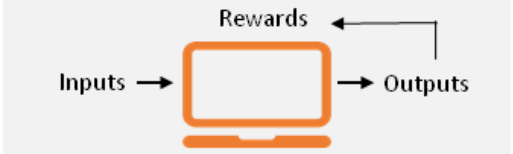


- Traditional approaches have a static handcrafted model.
- Traditional Models are generally a one step process but using ML will almost always a two step process involving “Learn Phase” followed by “Predict Phase”.

Machine Learning Approaches

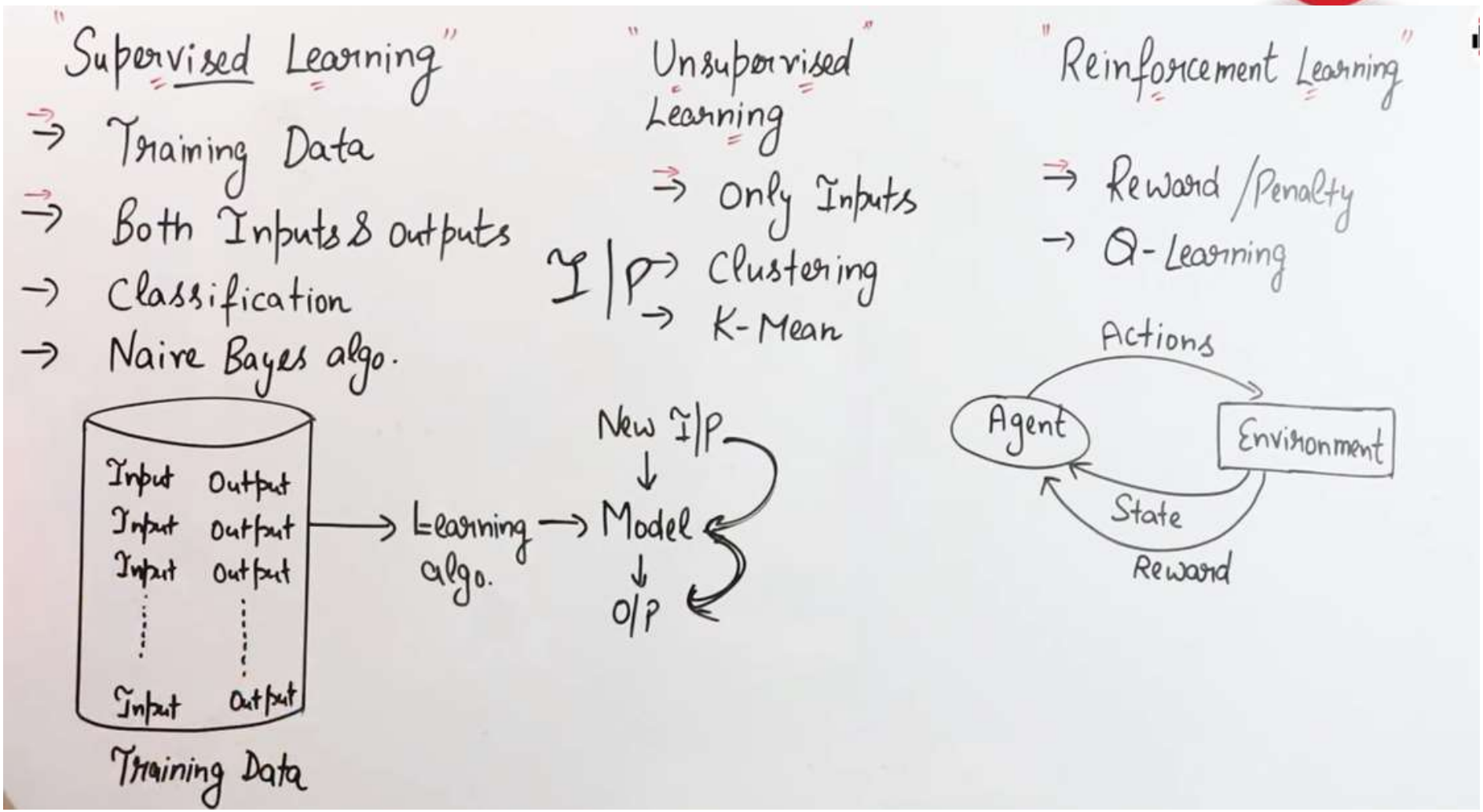
- *Supervised Learning*
- The machine learns explicitly with the help of labels
- Direct feedback is given
- Task driven process.
- Ex: Classification of mails into Spam/ham

- *Un Supervised Learning*
- Machine learns by trying to find patterns in the given data without labels
- No feedback is given
- Ex: Detecting unseen attacks in the system(clustering)



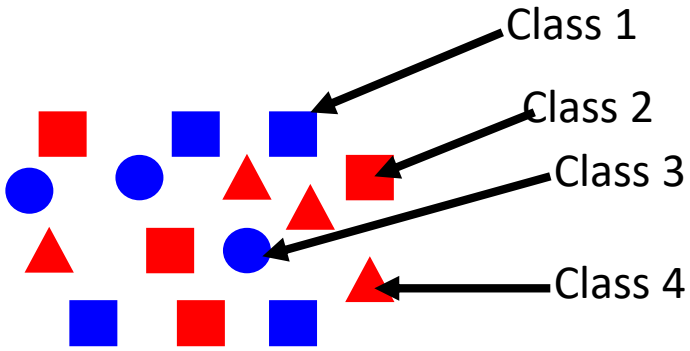
- *Reinforcement Learning*
- It works on positive and negative recommendations from the environment
- Reward Based learning analogous to human learning
- Example: Adaptive Cyber Defenses (ACD)

Machine Learning Approaches

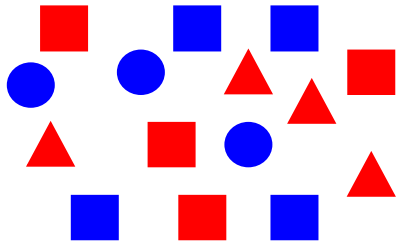


Learning Algorithms

- Supervised
  - Class information exists



- Unsupervised
  - NO Class information



Learning Agents

- Learns a classifier from examples: *Training stage*
- Classifies an unknown instance: *Testing or Classification stage*

Learning Algorithms

- Supervised
  - Nearest Neighbor / *k*-Nearest Neighbor
  - **Decision Tree**
  - **Neural Network / Artificial Neural Network**
  - Support Vector Machines
  - Bayesian Theory
    - Bayesian network / Belief Network
- Unsupervised
  - Clustering Algorithms

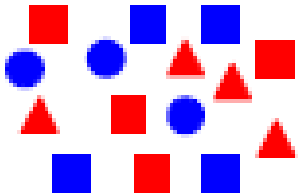
Decision Tree (DT)

- Easy to understand
- Can be expressed as *if then else* rules
- Builds a classification tree from examples
  - Top down methodology
  - Divide and conquer

- Uses the DT to classify unknown samples

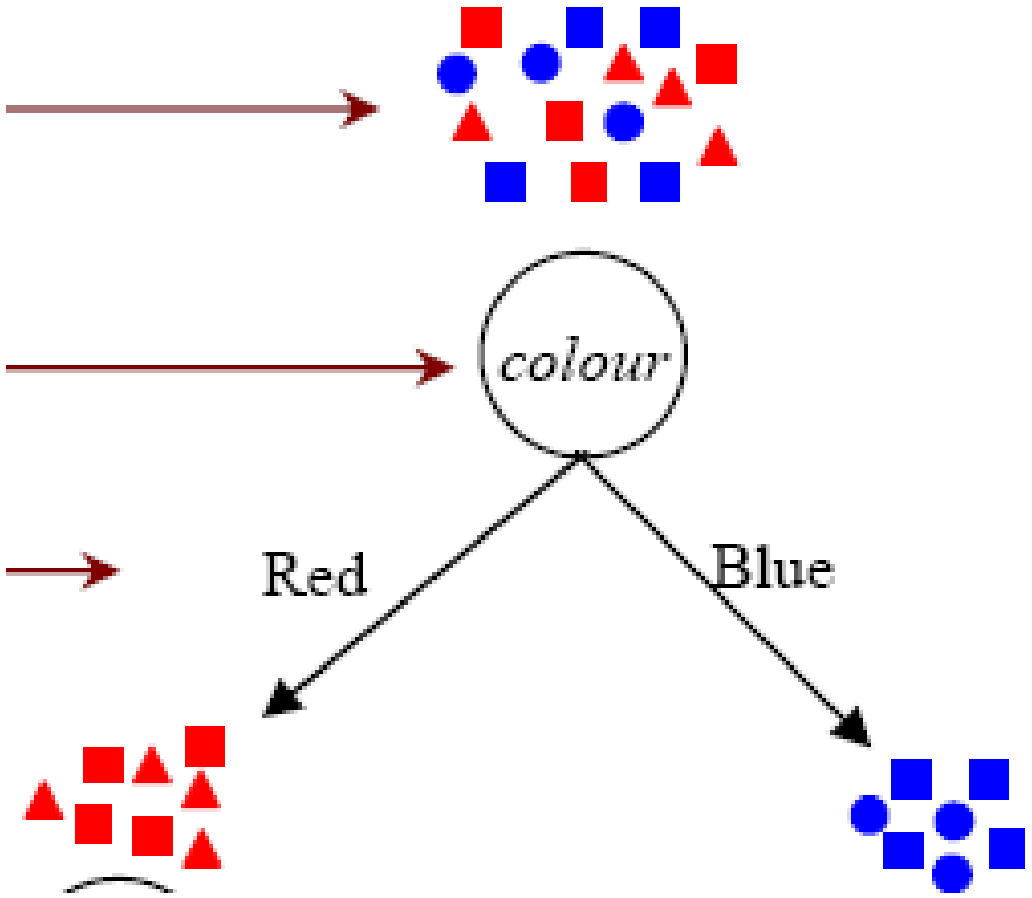
Decision Tree: An example

- Classify these objects

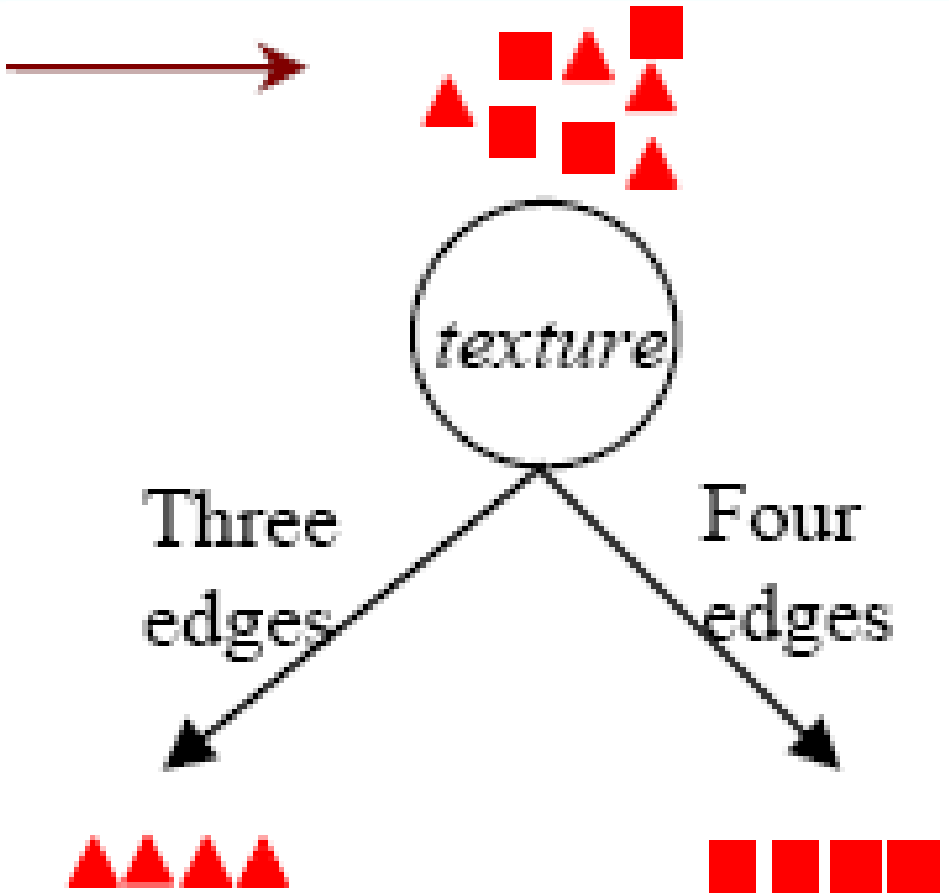


- Objects are represented by color, texture (*edge information*) and shape
- Class information is given

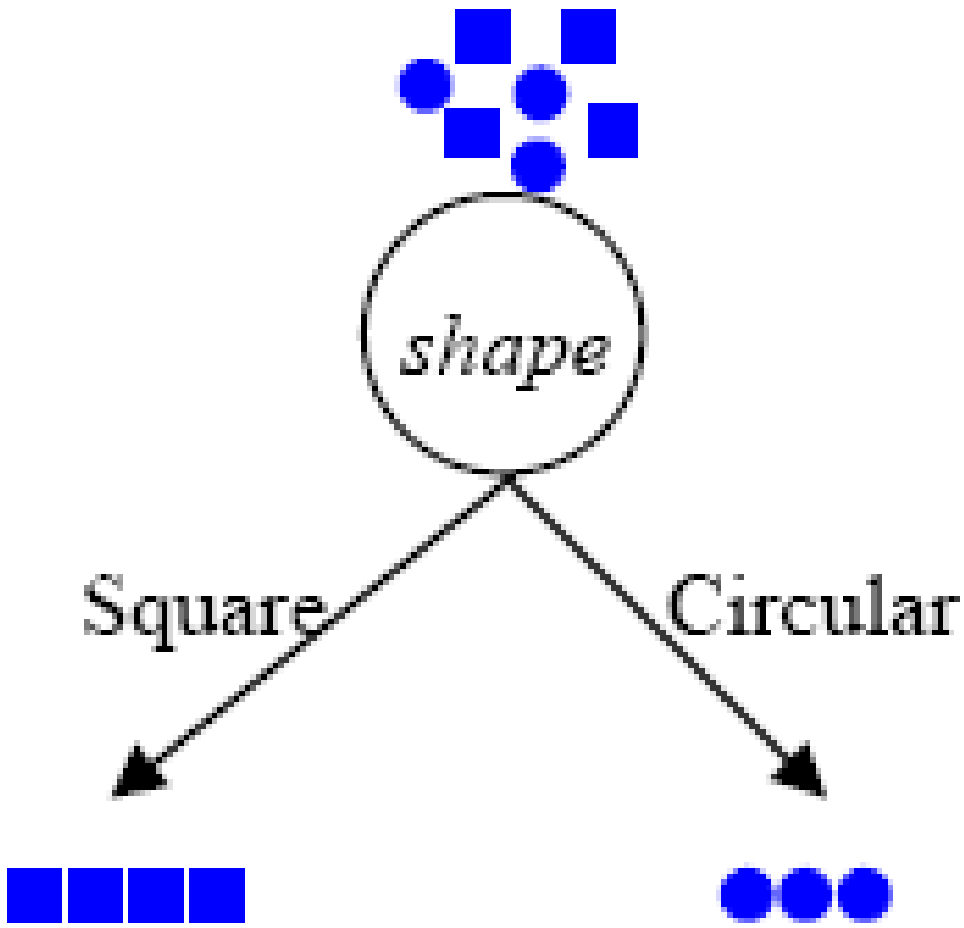
Decision Tree: An example

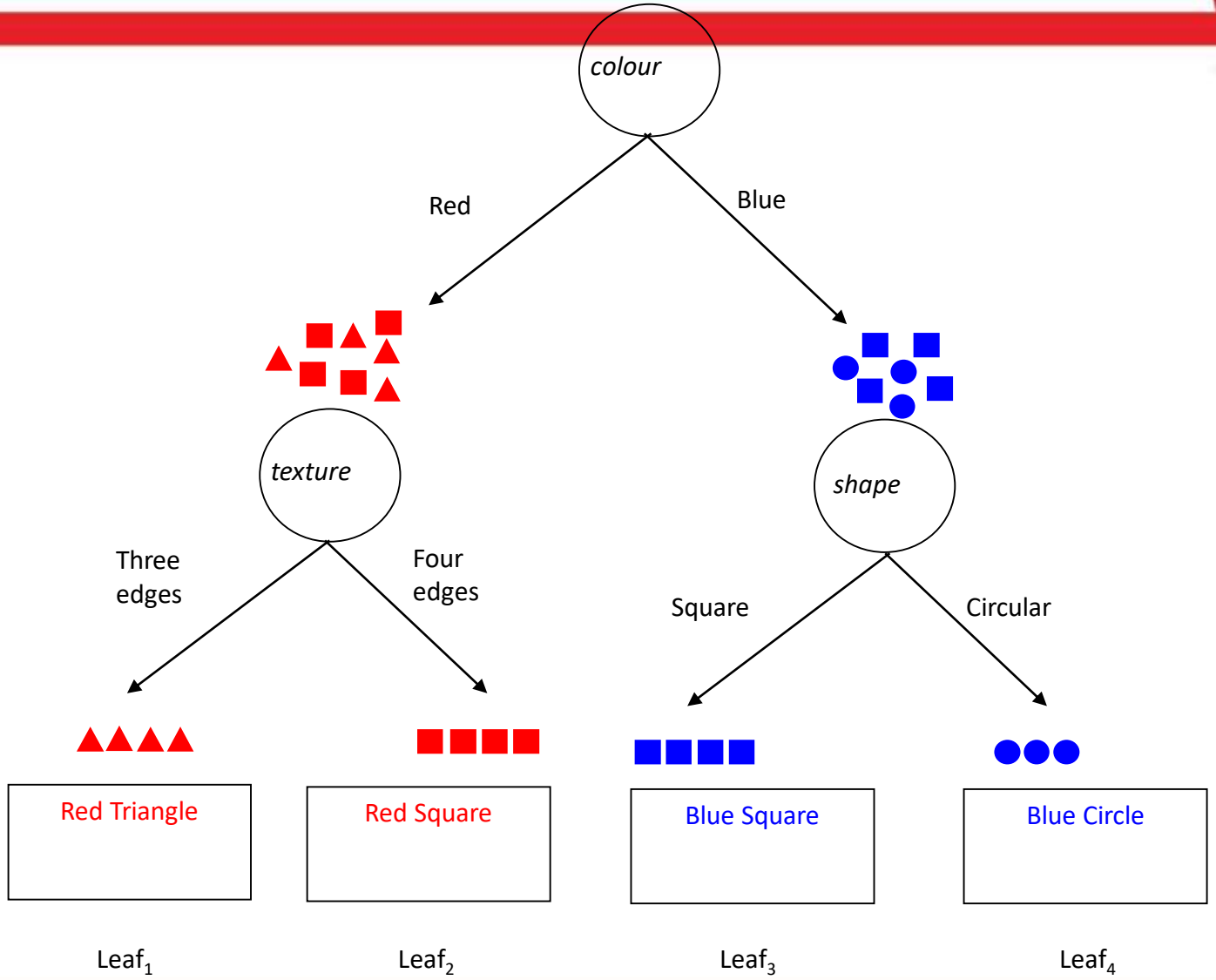


Decision Tree: An example



Decision Tree: An example




colour
Red
Blue
texture
shape
Three edges
Four edges
Square
Circular
Red Triangle
Red Square
Blue Square
Blue Circle
Leaf1
Leaf2
Leaf3
Leaf4

Rules from Decision Tree (DT)

if *colour* = Red and *texture* = Three edges
then *outcome* = Red Triangle — $Leaf_1$

if *colour* = Red and *texture* = Four edges
then *outcome* = Red Square — $Leaf_2$

if *colour* = Blue and *shape* = Square
then *outcome* = Blue Square — $Leaf_3$

if *colour* = Blue and *shape* = Circular
then *outcome* = Blue Circle — $Leaf_4$

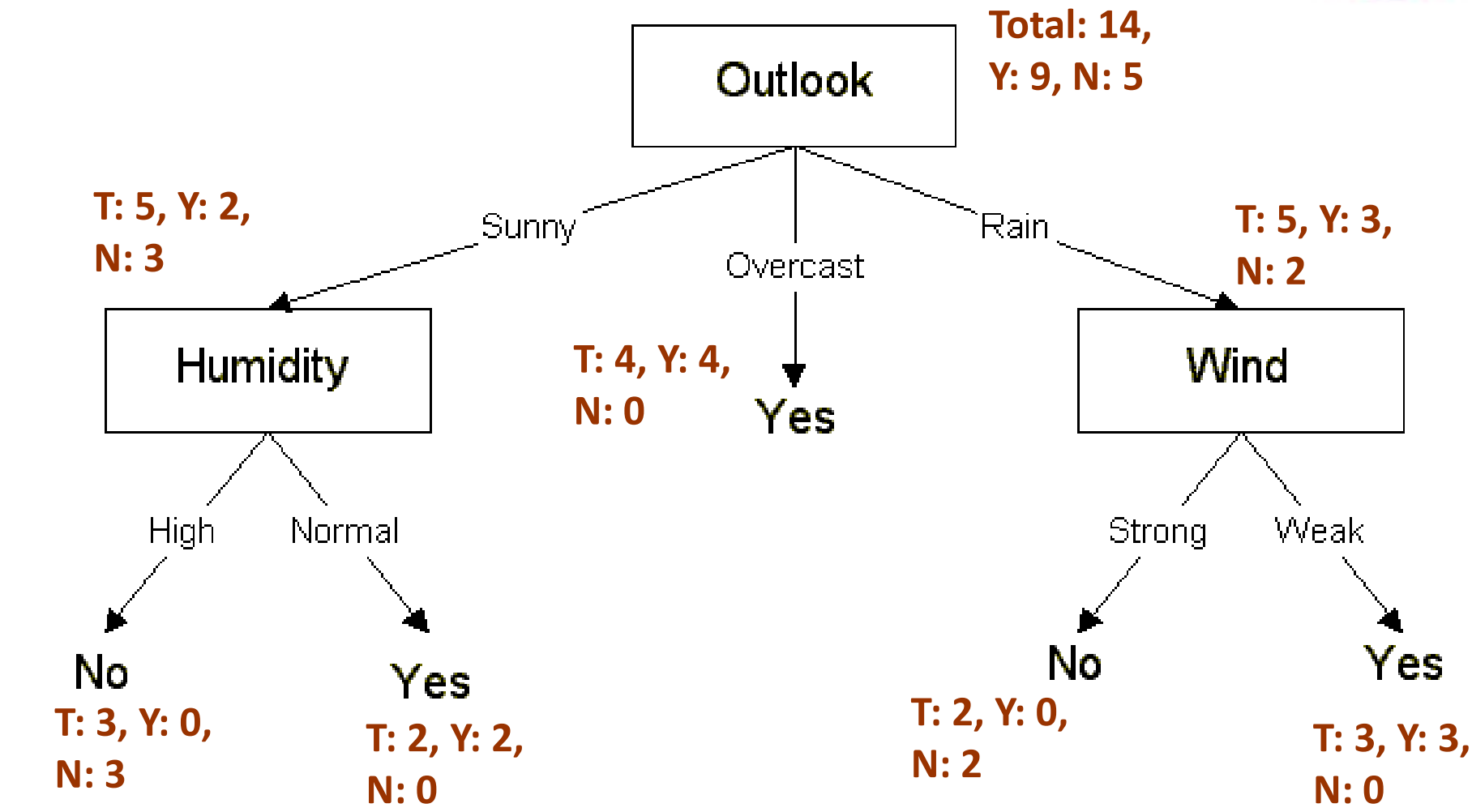
Issues in Decision Tree

- How to select the best characterizes or attributes?
- What happens when there are numerical (*continuous valued*) attributes?
- Outcome may also be numeric!
- When we should stop splitting?
- How to manage large DTs?

Another example of Decision Tree
The Weather Dataset

Case No.	Outlook	Temperature	Humidity	Wind	Play ball
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

Decision Tree



How to Select an Attribute

- Gain Criteria
  - Select an attribute which ***reduce the maximum uncertainty*** in data
- Gain measures
  - the reduction of uncertainty
  - How much info we can save after splitting the data

How to Select an Attribute

- Let *T* training examples
- Class labels: **{*C*1, *C*2, *C*3, …, *Cm*}**
- Attributes: **{*A*1, *A*2, *A*3, …, *An*}**
- Probability an instance belong to class C_j $P_j = \frac{\|T_j\|}{\|T\|}$
- Where, $T = \bigcup_{j=1}^{j=m} T_j$

How to Select an Attribute

- To classify an instance, we need info

$$info(T) = -\sum_{j=1}^{j=m} P_j \times \log P_j$$

How to Select an Attribute

Now consider the splitting:

- Let A_i has n_i nominal values such as, $A_i^1, A_i^2, A_i^3, \cdots, A_i^{n_i}$
- A_i splits T into $T_i^1, T_i^2, T_i^3, \cdots, T_i^{n_i}$

where, $$T = \bigcup_{k=1}^{k=n_i} T_i^k$$

How to Select an Attribute

Now consider the splitting:

- Expected info needed to know the partition of an instance is,

$$E(A_i) = \sum_{j=1}^{j=n_i} \frac{\left\| T_i^j \right\|}{\|T\|} \times info(T_i^j)$$

$$gain(A_i) = info(T) - E(A_i)$$

How to Select an Attribute

Now consider the splitting:

- We can save, that is, *gain* is,

$$gain(A_i) = info(T) - E(A_i)$$

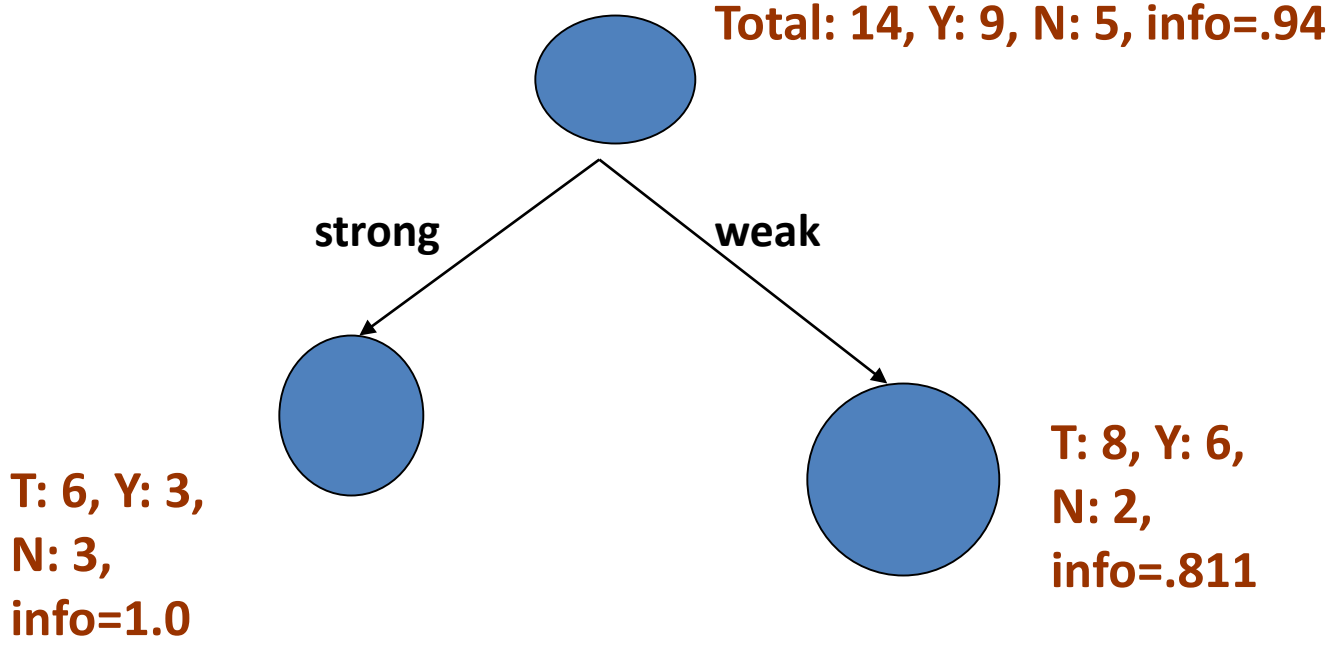
- So, select the attribute which as **max** *gain*

Illustration

Information at root:

info (T) = - (9/14) Log2 (9/14) - (5/14) Log2 (5/14)=.94

Suppose we split with attribute wind:



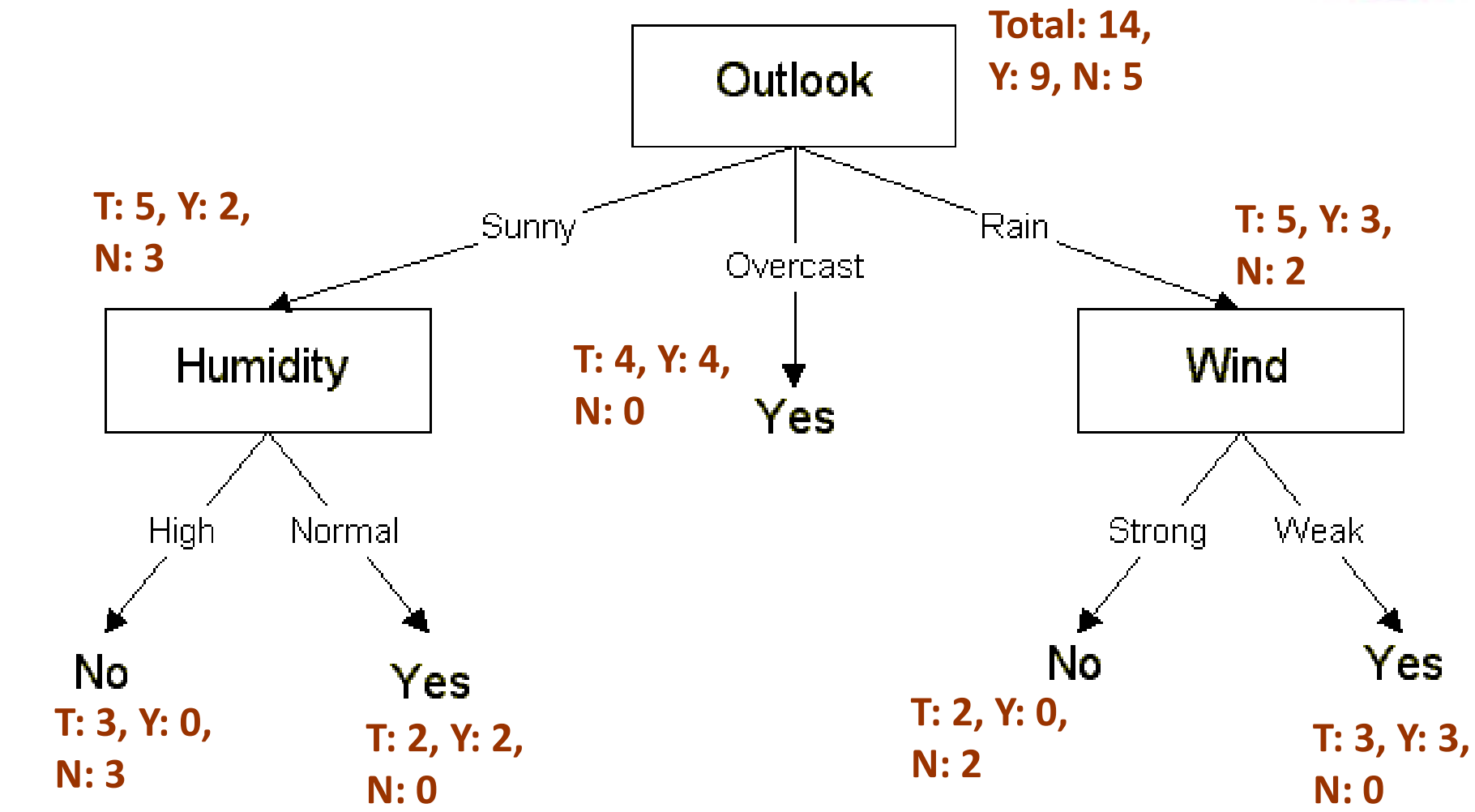
gain (wind) = *info*(*T*)- (8/14) *info*(*weak*)- (6/14) *info(strong)* =.048

Illustration

Similarly,

- gain(outlook)=.246
- Gain(Temparature)=.029
- Gain(Humidity)=.151
- Gain(Windy)=.048

Illustration



How to Select an Attribute (2)

- *gain ratio* Criteria
  - *Normalize the gain* to remove the effect of highly branching attributes

$$gainRatio\,(A_i) = \frac{gain\,(A_i)}{splitInfo\,(A_i)}$$

How to Select an Attribute (2)

- *split info*:
  - information generated because of splitting
  - Only uses the partition
  - Does NOT consider class membership

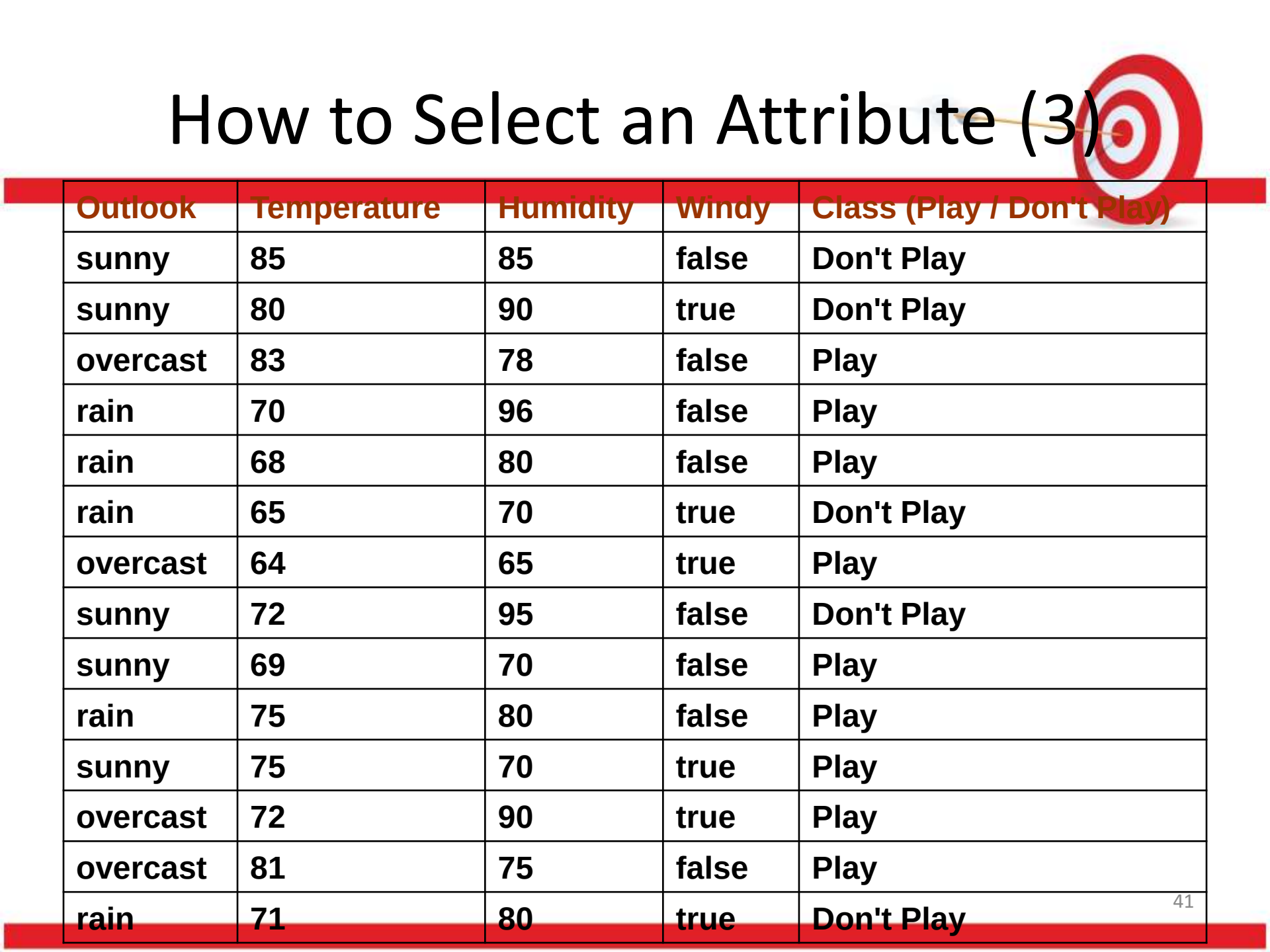
$$splitInfo\,(A_i) = -\sum_{j=1}^{j=n_i} \left(\frac{\left\|T_i^j\right\|}{\|T\|} \times \log\left(\frac{\left\|T_i^j\right\|}{\|T\|}\right)\right)$$

How to Select an Attribute (2)

- *split info*:
  - The higher is the branching factor, the bigger is the value

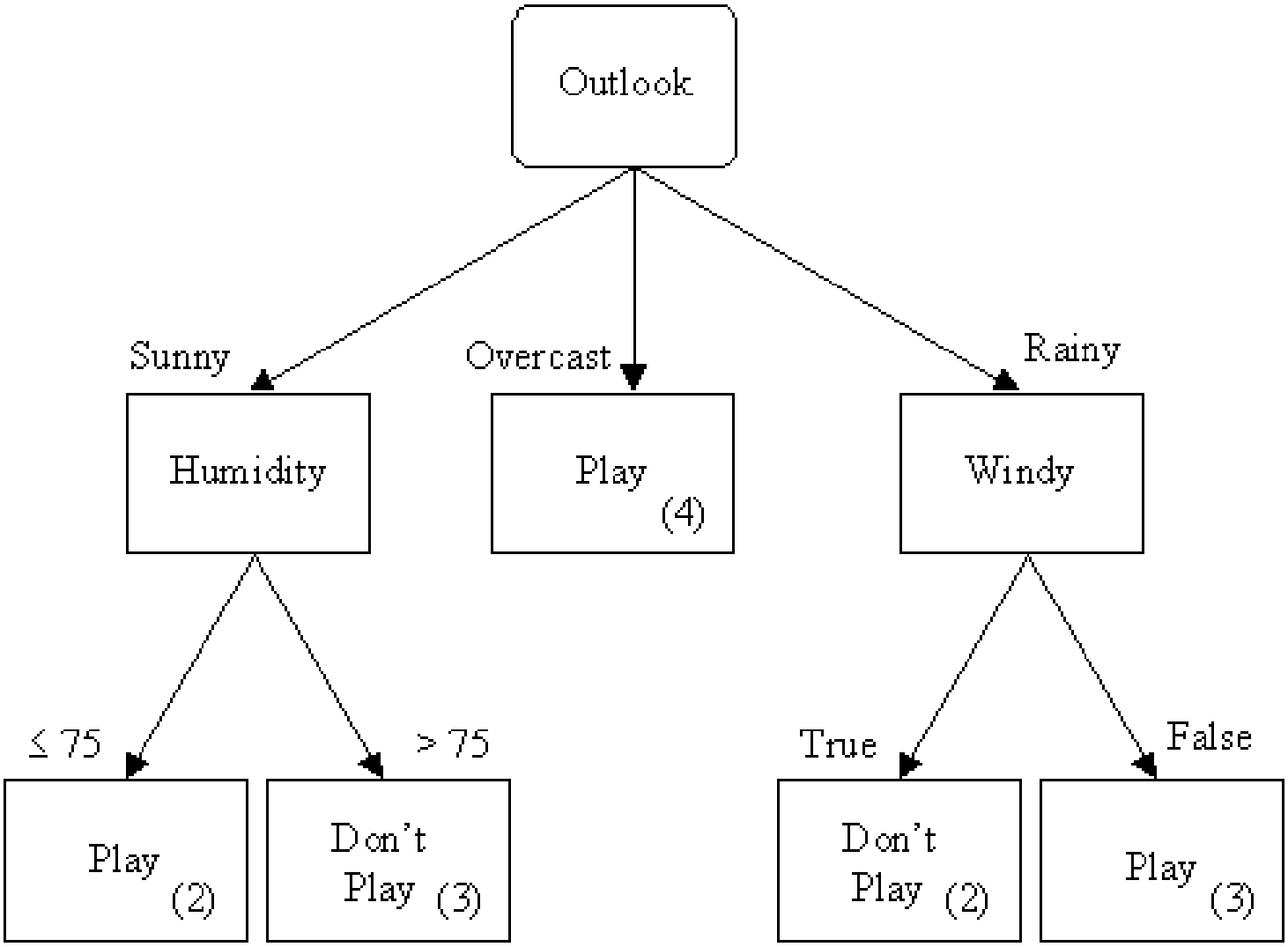
$$splitInfo\,(A_i) = -\sum_{j=1}^{j=n_i} \left(\frac{\left\|T_i^j\right\|}{\|T\|} \times \log\left(\frac{\left\|T_i^j\right\|}{\|T\|}\right)\right)$$

How to Select an Attribute (3)

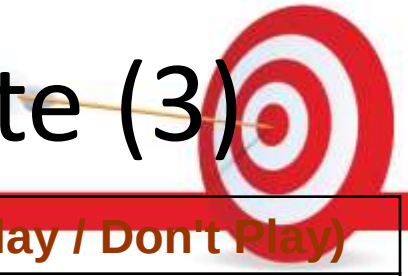


Outlook	Temperature	Humidity	Windy	Class (Play / Don't Play)
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

How to Select an Attribute (3)



How to Select an Attribute (3)



Outlook	Temperature	Humidity	Windy	Class (Play / Don't Play)
sunny	85	85	false	Don't Play
sunny	80	90	true	Don't Play
overcast	83	78	false	Play
rain	70	96	false	Play
rain	68	80	false	Play
rain	65	70	true	Don't Play
overcast	64	65	true	Play
sunny	72	95	false	Don't Play
sunny	69	70	false	Play
rain	75	80	false	Play
sunny	75	70	true	Play
overcast	72	90	true	Play
overcast	81	75	false	Play
rain	71	80	true	Don't Play

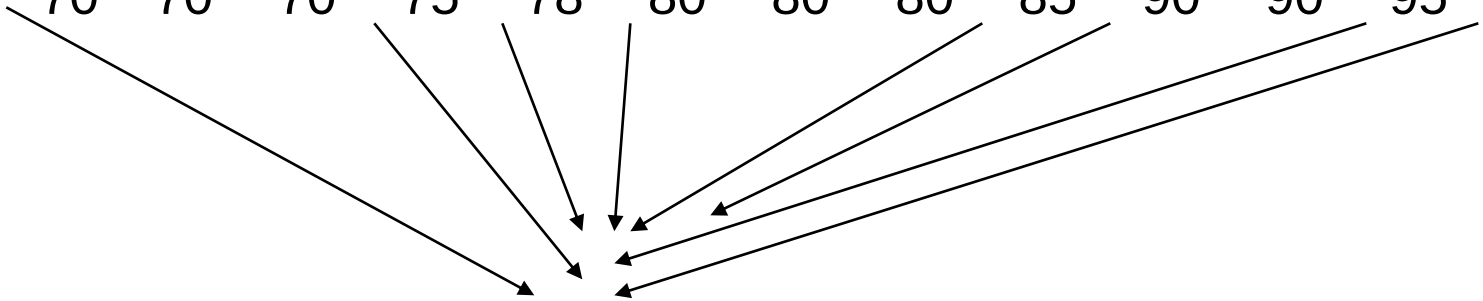
How to Select an Attribute (3)

- Sort the values:

85 90 78 96 80 70 65 95 70 80 70 90 75 80

- After Sort:

65 70 70 70 75 78 80 80 80 85 90 90 95 96

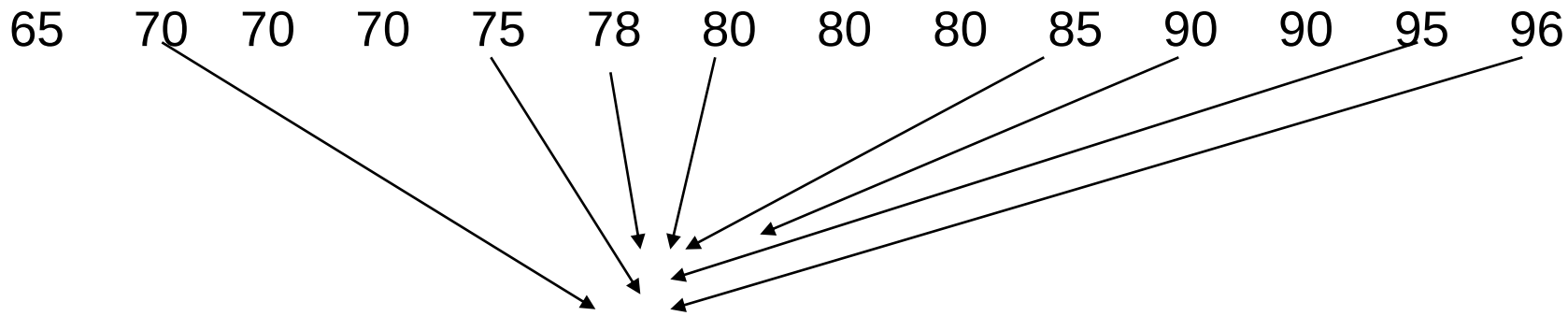


Humidity > mid value

- At each mid value, make a separate test

How to Select an Attribute (3)

- After Sort:



Humidity > max in the interval

- Some Algorithms use the maximum value instead of mid value

Pruning a DT

- Why?
  - Simpler is better
  - Manage extremely big tree
  - Big tree over fits on training data, but is poor on test data

Pruning a DT

- When to prune?
  - During tree generation: pre pruning
    - Don’t partition
  - Remove some of the tree once it has been built: post pruning

Pruning a DT

- Pre pruning
  - Use a threshold to ignore partition
  - Too low threshold won’t benefit at all
  - Too high threshold terminate quickly

Pruning a DT

- Post pruning
  - Check which branch can be removed
  - Error based pruning
    - Cost complexity pruning
    - Reduced error pruning
    - Both uses separate data for error rate calculation

Pruning a DT

- Problem of Post pruning
  - Reduces training data
  - Original tree may be produced from small data
- Solution:
  - Use cross validation -