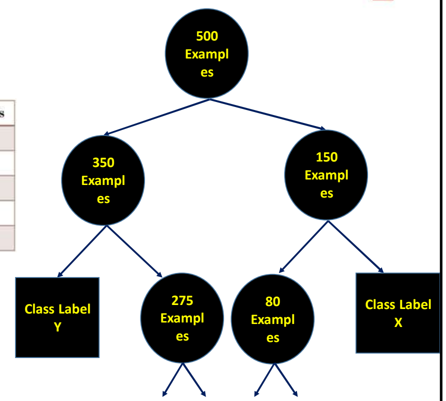
Decision Tree Learning

Ideating Decision Trees

	A_1	A_2	...	A_{200}	Class
#1					
#2					
#3					
...					
#500					



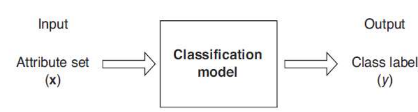
LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contents

LEARN . GROW . EXCEL GOPEEKRISHNAN R

1. Recap of Classification

- **Classification** is the task of assigning objects to one of several pre-defined categories.



- The Input data for classification task: a collection of records.
- Each record is an "example". [NOTE: record == instance == example
- Each example is characterized by a tuple (x, y) where x is an attribute set, and y is a special attribute, designated as the class label.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- A classification model is an abstract relationship between the attribute set and the class label.
- A model can be represented in many ways. e.g., as a tree, a probability (likelihood) table, or simply, a vector of real – valued parameters.
- More formally, we can express it mathematically as a target function **f** that takes as input the **attribute set x** and produces an output corresponding to the predicted **class label**. The model is said to classify an instance (x, y) correctly if **f(x) = y**.

Contd…

- Examples of classification tasks given below.

Task	Attribute set	Class label
Spam filtering	Features extracted from email message header and content	spam or non-spam
Tumor identification	Features extracted from magnetic resonance imaging (MRI) scans	malignant or benign
Galaxy classification	Features extracted from telescope images	elliptical, spiral, or irregular-shaped

Example: Classification

- See the following figure that shows the classification of vertebrates (= an animal having a backbone) into mammals, reptiles, birds, fishes, and amphibians.
- Attribute Set:
  - body temperature
  - skin cover
  - gives birth
  - aquatic creature
  - aerial creature
  - has legs
  - hibernates
- The data set can also be used for binary classification into: mammals and non-mammals (reptiles, birds, fishes, and amphibians).

Contd…

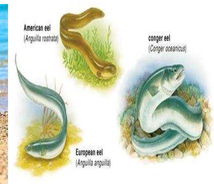
Vertebrate Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
human	warm-blooded	hair	yes	no	no	yes	no	mammal
python	cold-blooded	scales	no	no	no	no	yes	reptile
salmon	cold-blooded	scales	no	yes	no	no	no	fish
whale	warm-blooded	hair	yes	yes	no	no	no	mammal
frog	cold-blooded	none	no	semi	no	yes	yes	amphibian
komodo dragon	cold-blooded	scales	no	no	no	yes	no	reptile
bat	warm-blooded	hair	yes	no	yes	yes	yes	mammal
pigeon	warm-blooded	feathers	no	no	yes	yes	no	bird
cat	warm-blooded	fur	yes	no	no	yes	no	mammal
leopard shark	cold-blooded	scales	yes	yes	no	no	no	fish
turtle	cold-blooded	scales	no	semi	no	yes	no	reptile
penguin	warm-blooded	feathers	no	semi	no	yes	no	bird
porcupine	warm-blooded	quills	yes	no	no	yes	yes	mammal
eel	cold-blooded	scales	no	yes	no	no	no	fish
salamander	cold-blooded	none	no	semi	no	yes	yes	amphibian

Contd…

- (human, warm-blooded, hair, yes, no, no, yes, no, mammal) is a typical *record* or *example*.

Contd…

KOMODO



EEL

PORCUPINE

SALAMANDER

Contd…

- Now, a classification model induced from the vertebrate data set can be used to predict (deduct) the class label of the following vertebrate.

Vertebrate Name	Body Temperature	Skin Cover	Gives Birth	Aquatic Creature	Aerial Creature	Has Legs	Hiber-nates	Class Label
gila monster	cold-blooded	scales	no	no	no	yes	yes	?

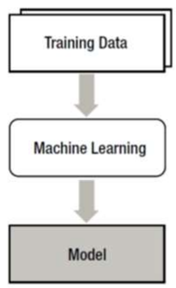
Contd…

- **Classification** – the task of assigning labels to unlabeled data instances.
- **Classifier** – is a systematic approach to build classification models from an input dataset.
- **Examples of classifiers:**
  - Decision – tree classifiers
  - Rule-based classifiers
  - Neural Networks
  - Support Vector Machines
  - Naïve Bayes Classifiers *etc.*

SAINTGITS

Contd…

- Each classifier has an equivalent machine **learning algorithm**.
- The learning algorithm **identifies a model** that **best fits** the relationship between
  - the attribute set and,
  - class label of the input data.
- This input data is the **training data**.



SAINTGITS

Contd…

- **INDUCTION** - the process of *applying* a learning algorithm to build a classification model from the training data.
  - Induction aka "LEARNING A MODEL" | "BUILDING A MODEL".
- **DEDUCTION** - the process of *applying* a classification model on unseen test instances to predict their class labels.

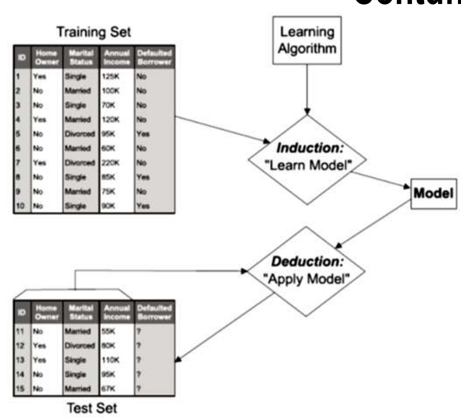
SAINTGITS

Contd…

- The **model** generated by the learning algorithm ***should*** both
  a) ***fit*** the **training data well** and,
  b) correctly ***predict*** the class labels of records it **has never seen before**.

SAINTGITS

Contd…



Concept of Decision Trees

- This section introduces the Decision Tree Classifier.
- This is introduced with an example.
- The example we use is ‘classification of vertebrates’.

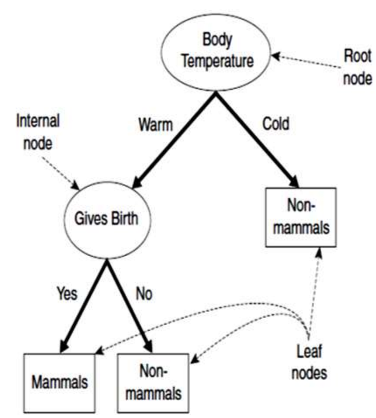
Example #1 – Classification using Decision Trees

- **Suppose a new species is found out by scientists.**
- **How can we tell it is a mammal or non-mammal?**
- **One of the ways is…asking a series of questions….just like doctors ask patients…**
  - ☑ **First question can be like this – whether the new species is warm – or cold – blooded.**
    - ☑ **If it is cold – blooded, definitely it is not a mammal.**
    - ☑ **If is warm – blooded, it can be a bird or a mammal.**
      - ☑ **In the latter – case, we may need to ask a follow – up question: do the females of the species give birth to their young?**
        - ☑ **Those that do give birth are definitely mammals, while those that do not are likely to be non – mammals.**

Contd…

- The previous example illustrates how we can solve a classification problem by asking a series of carefully crafted questions about the attributes of the test instance.
- Each time we receive an answer, we could ask a follow-up question until we can conclusively decide on its class label.
- The series of questions and their possible answers can be organized into a hierarchical structure called A DECISION TREE.
- The following figure shows the decision tree for this classification problem.

Contd…



Contd…

- The tree has three types of nodes.
  - A **root node** – with no incoming links and zero or more outgoing links.
  - **Decision nodes** (Internal nodes) – each of which has exactly one incoming link and two or more outgoing links. *[excludes ROOT NODE from having the requirement of one incoming link.]*
  - **Leaf or terminal nodes** – each of which has exactly one incoming link and no outgoing links.

Note: A root node also can be a decision node.

Contd…

- **Some vital points…**

✓ Each leaf node in the decision tree is associated with a class label.

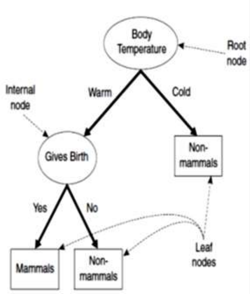
✓ **Decision nodes require choices (decisions) to be made that are typically defined using a single attribute.**

✓ Choices of decision nodes split the data across branches that indicate potential outcomes.

✓ Each possible outcome of a decision node is associated with exactly one child of this node.

Contd…

- The root node of the tree shown in the following figure uses the attribute **Body Temperature** to define an attribute test condition that has two outcomes, warm and cold, resulting in two child nodes.



- **Note:** Root Node also is a decision node

Contd…

- **Given a decision tree, Classifying a test instance is straightforward.**

  a) Starting from the root node, we apply its attribute test condition and follow the appropriate branch based on the outcome of the test.

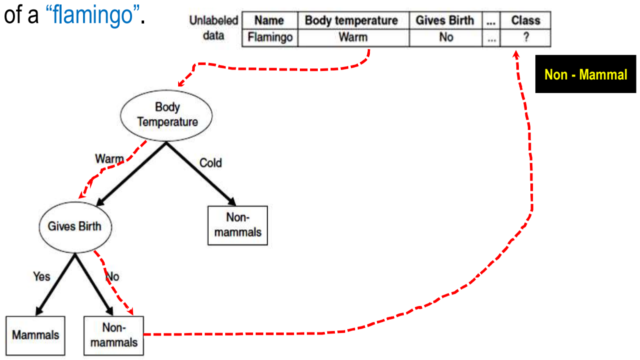
  b) This will lead us either to

     1. *another internal node (for which a new attribute test condition is applied), OR*
     2. *to a leaf node.*

  c) Once a leaf node is reached, we assign the class label associated with the node to the test instance.

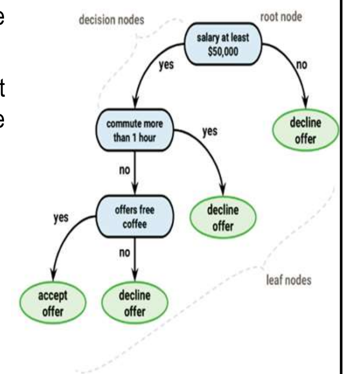
Contd…

- The following figure traces the path used to predict the class label of a "flamingo".



Example #2 – Classification using Decision Trees

- Consider the decision tree to the right.
- The **objective** is – to predict whether a job offer should be accepted or not.



Contd…

- Just like Example#1….we ask a series of questions…
  - ☑ First question can be like this – whether the salary is at the least $50,000 – or less than $50,000.
    - ☑ If salary < $50,0000, definitely the job offer is declined.
    - ☑ If salary >= $50,000, possibility is there to accept the offer.
      - ☑ In the latter – case, we may need to ask a follow – up question: does the job require travelling more than 1 hour?

Contd…

- ☑ If commutation requires more than one hour, the job offer is declined.
- ☑ If commutation requires less than one hour, the job offer may be accepted.
  - ☑ In the latter case, we may need to ask a follow – up question: does the job offers free coffee?
    - ☑ If the job offers a free coffee, the job is accepted.
    - ☑ If the job does not offer a free coffee, the job is declined.

Contd…

- **QUESTIONS:** Given the following unlabeled data. Find their classes.

JobId	Salary_Offer	Travelling_Time	Offers_Free_Coffee	Class
#3013	$65,900	40 Mnts	Yes	?

JobId	Salary_Offer	Travelling_Time	Offers_Free_Coffee	Class
#3010	$65,900	40 Mnts	No	?

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- Decision trees are perhaps the **most widely used machine learning technique**.
- Therefore, this can be applied to model almost any type of data.
- But, there are some situations where trees may not be an ideal fit.
  - One such case is a task where the data has a large number of nominal features with many levels or it has a large number of numeric features.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Divide and Conquer Approach

- It is NOW CLEAR that, the classification is easy once a decision tree has been built / constructed.
- Now, we need to understand, **HOW SUCH A DECISION TREE IS BUILT** with the input training data.
- For this, divide – and – conquer strategy is adopted.
- IN GENERAL, a divide – and – conquer strategy does the following…
  - *First, the given data set is split into subsets.*
  - *Then, these subsets are split repeatedly into even smaller subsets and so on…*
  - *This is continued until the process stops when the algorithm determines the data within the subsets are sufficiently homogeneous, OR another stopping criterion has been met.*

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Building Decision Trees based on DAC Strategy

- Steps for constructing a decision tree using divide – and – conquer method.
- Imagine that…a bare root node is growing into a mature tree.
  - **FIRST,** the root node represents the entire data set, since no splitting has happened, till now.
  - **NEXT,** the algorithm has to choose a feature to split upon. It chooses the feature that is the most predictive of the target data.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- **THEN**, the examples are partitioned into groups according to the distinct values of this feature, and the first set of branches are formed.
- **NEXT,** the algorithm works down on each branch (of this first split).

Contd…

- The algorithm continually **divides** and conquers the data, choosing the best candidate feature each time to create another decision tree.
- This process of Divide-And-Conquer **<u>might stop at a node</u>** in a case when:
  1. ***All (or nearly all) of the examples at the node have the same class.***
  2. ***There are no remaining features to distinguish among the examples.***
  3. ***The tree has grown to a pre-defined size limit.***

LEARN . GROW . EXCEL GOPEEKRISHNAN R

C5.0 Decision Tree Algorithm

Function C5.0()

This function applies the divide and conquer strategy onto the example set **S** to create a decision tree **DT.**

1. ***If*** all examples in **S** belong to the same class ***'c'***, *then:*

return(a new leaf and label it with ***'c'***)

Else:

a) Select an attribute **A** according to some <u>impurity function</u>.

b) Generate a new node **DT** with **A** as a test.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

c) For each value v_i in **A**

i) Let **Si** = all examples in **S** with **A =** v_i.

// **<u>Note:</u>** there are other examples of **S**

// here **A** $\neq v_i$

ii) Use **C5.0()** to construct a decision tree ***DTi*** for example sets ***Si***.

iii) Generate an edge that connects ***DT*** and ***DTi***.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Illustration – Decision tree construction

- Imagine that <You> work for a Hollywood Studio….as Production Manager.
- **Objective:** to decide whether the studio should produce screenplays written by new authors.
- After returning from a vacation, assume your desk is stacked with a number of such proposals.
- <You> do not have time to read all of them.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- <You> decided to develop a decision tree algorithm to predict whether a potential movie (from these proposals) would fall into any one of these three categories:
  - Critical Success
  - Mainstream Hit
  - Box-Office Bust

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- **But, where is PAST DATA????**
  - *Refer the library of Studio.*
- **What kind of past data do we need to refer to?**
- **Data / Factors that led to the success and failure of the Studio's most 30 (only) releases. These include…**

  A) Estimated shooting budget

  B) Number of A – list celebrities lined up for acting

  C) Level of Success

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

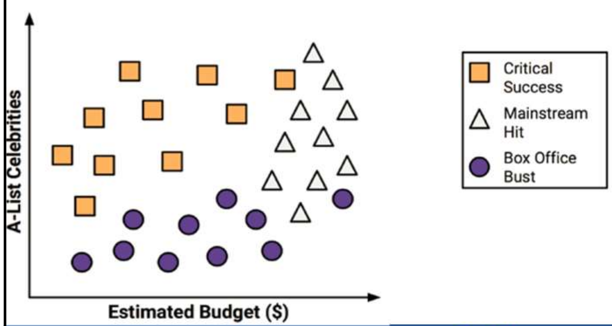
	A_1	A_2	C
Film Name	Estimated Shooting Budget ($)	Number of A – List Celebrities lined up for acting	Level of Success
Film #1	3,50,000	9	Mainstream Hit
Film #2	3,56,000	14	Critical Success
Film #3	14,00,000	5	Box Office Bust
.			
.			
.			
Film #30	45,000,000	25	Critical Success

LEARN . GROW . EXCEL GOPEEKRISHNAN R

SAINTGITS

Contd…

- Based on these…
- <you> produce a scatterplot…



SAINTGITS

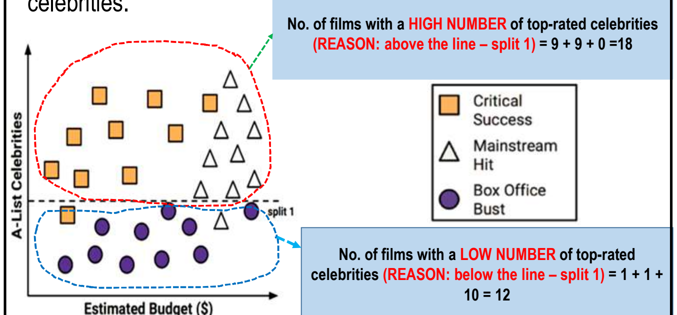
Contd…

- Using divide-and-conquer strategy, we can **build a simple decision tree** from this data.

SAINTGITS

Contd…

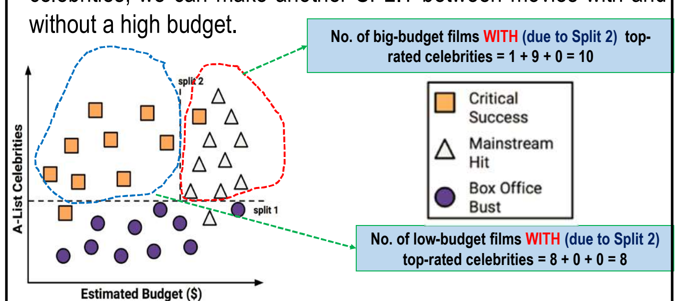
- First, we SPLIT the movies ***with*** and ***without*** top – rated celebrities.



SAINTGITS

Contd…

- Next, among the group of movies WITH a large number of celebrities, we can make another SPLIT between movies with and without a high budget.

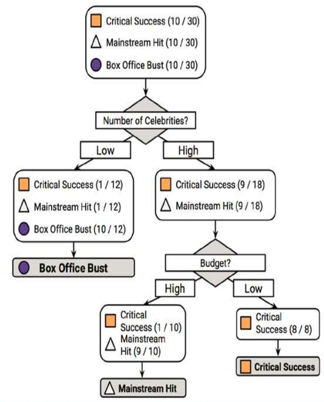


Contd…

- **Now, we have partitioned the data into three groups.**
- **Group#1**
  - Top-left-corner: composed entirely of **critically acclaimed films**.
  - This group is composed of films with a higher-number of celebrities but with relatively lower budget.
- **Group#2**
  - Top-right-corner: majority of the movies are **box-office hits** with high budgets and a large number of celebrities.
- **Group#3**
  - Bottom-portion: little star power but budgets ranging from small to large, composed of **flops**.

Contd…

- This model for predicting future successes of movies can be represented in the following decision tree.

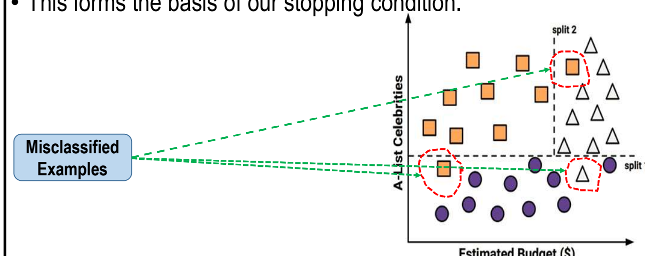


Contd…

- If we want, we can continue to divide–and–conquer the data by splitting it.
- This splitting could be based on the increasingly specific ranges of celebrity count (Y–axis) and the budget count (X–axis).
- This divide–and–conquer could be done **until** each of the currently misclassified values reside in each of its tiny partition, and is correctly classified.
- However, it is **not advisable** to **overfit** a decision tree in this way. *(That is, do not indefinitely apply divide-and-conquer strategy)*

Contd…

- We avoid the problem of overfitting by stopping the algorithm here (i.e., decision tree building using D-A-C strategy), since more than 80% of the examples in each group are from a single class.
- This forms the basis of our stopping condition.



Entropy and Information Gain

- This process of Divide-And-Conquer **might stop at a node** in a case when:
  1. ***All (or nearly all) of the examples at the node have the same class.***
  2. ***There are no remaining features to distinguish among the examples.***
  3. ***The tree has grown to a pre-defined size limit.***

Contd…

- The **first challenge:**
  - **Identify** which attribute to split upon.
  - This attribute **is called** the **next-split-attribute**.

Contd…

- The decision to find the **best** **next-split-attribute** depends on a metric **called** **purity**.
- The degree to which a subset of examples belong to the same class **is known** as **purity**.
- Any subset that is composed only of examples of a single class **is said to be** **pure**.

Contd…

- There are various measurements of purity that can be used to find the best next-split-attribute.
- The C5.0 decision tree algorithm uses **Entropy**.

Contd…

- It is typically measured in bits.
- If there are only two possible classes ($\therefore$ 2 bits), entropy values can range from **0** to **1**.
- If there are n classes, entropy values can range from **0** to **$\log_2 n$**.
- In either case,
  - ❑a minimum value (low value) indicates that the sample data drawn is completely homogeneous.
  - ❑a maximum value (high value) indicates that the sample data drawn is as much diverse as possible.

Contd…

- **Entropy** quantifies the randomness / uncertainty present within a data set.
- **When entropy value is high, prediction is difficult; and, when entropy value is low, prediction becomes easy.**
- It is a concept borrowed from Thermodynamics.
- In Mathematical notation, entropy is:

$$\textbf{Entropy(S)} = \sum_{i=1}^{c} -pi.\,log_2(pi)$$

  - *S: Given segment (subset) of data.*
  - *c: number of class levels.*
  - *P_i: is the probability of some event.*

Contd…

- **Information Gain** is a measure of how good an attribute is for predicting the class of each of the example of the training data.
- We will select the attribute with the highest information gain as the next-split-attribute.
- It is calculated as the difference between the entropy before the split (S_1) and, the (entropies of the) partitions resulting from the split (S_2).

$$\textbf{InfoGain(F) = Entropy(}S_1\textbf{) – Entropy(}S_2\textbf{)}$$

Example 1: Construct the decision tree for the data…

Owns_home?	Married	Gender	Employed	Credit_Rating	Risk_Class
Yes	Yes	Male	Yes	A	B
No	No	Female	Yes	A	A
Yes	Yes	Female	Yes	B	C
Yes	No	Male	No	B	B
No	Yes	Female	Yes	B	C
No	No	Female	Yes	B	A
No	No	Male	No	B	B
Yes	No	Female	Yes	A	A
No	Yes	Female	Yes	A	C
Yes	Yes	Female	Yes	A	C

Contd...

Owns_home?	Married	Gender	Employed	Credit_Rating	Risk_Class
~~Yes~~	~~Yes~~	~~Male~~	~~Yes~~	~~A~~	~~B~~
No	No	Female	Yes	A	A
Yes	Yes	Female	Yes	B	C
~~Yes~~	~~No~~	~~Male~~	~~No~~	~~B~~	~~B~~
No	Yes	Female	Yes	B	C
No	No	Female	Yes	B	A
~~No~~	~~No~~	~~Male~~	~~No~~	~~B~~	~~B~~
Yes	No	Female	Yes	A	A
No	Yes	Female	Yes	A	C
Yes	Yes	Female	Yes	A	C

LEARN . GROW . EXCEL GOPEEKRISHNAN R

After *Split#1*

Owns home?	Married	Employed	Credit Rating	Risk Class
No	No	Yes	A	A
Yes	Yes	Yes	B	C
No	Yes	Yes	B	C
No	No	Yes	B	A
Yes	No	Yes	A	A
No	Yes	Yes	A	C
Yes	Yes	Yes	A	C

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Example 2: Construct the decision tree for the data...

Day	Outlook	Temp.	Humidity	Wind	Play Outside?
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

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Pruning the Decision Tree

- A decision tree can continue to grow indefinitely, choosing splitting features and dividing the data *(of the chosen feature)* into smaller and smaller partitions <u>until</u>

  - A) each example is perfectly classified

    OR

  - B) the algorithm runs out of features to split on.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Contd…

- If a decision tree grows excessively (= overly) large, many decisions it makes *(during construction with the training data)* will be overly specific and the model will be **overfitted** *to the training data*.
- But it will fail miserably for test data.
- The process of **pruning** a decision tree involves reducing its size such that it generalizes better to unseen data.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Pre – Pruning *Or, Early Stopping*

One of the solutions to this indefinite growth…

A) *Stopping the tree from growing once it reaches a certain number of decisions.*

OR

B) *Stopping the tree from growing when the decision nodes (particularly at the bottom levels) contain only a small number of examples.*

- This seems to be an appealing strategy.
- But, if we stop the growth once the decision nodes contain a small no. of examples *(this may include examples belonging to other classes too!!!)*, this would miss some important examples that it would have learned, had it grown to a larger size.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Post – Pruning

- Here,
  - we intentionally let the tree to become large…
  - then, prune the leaf nodes to reduce the size of the tree to a more appropriate level.
- This is the best approach than pre – pruning *(where missings can be there!!!)*, because it is difficult to determine the depth of a decision tree without growing it first.
- This will ensure that, the algorithm has discovered all the important examples.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

And…we do..

- **Subtree Raising** or **Subtree Replacement**

1. *Allow the tree to become as large so that it overfits the training data.*

2. *Then, nodes and branches that have little effect on the classification errors are removed.*

- In some cases, entire branches are **moved further up** the tree. This is equivalent in saying replacement of advanced decisions by simpler decisions.
- These processes of grafting branches are known as **Subtree Raising** or **Subtree Replacement**.

LEARN . GROW . EXCEL GOPEEKRISHNAN R

Strengths Vs. Weaknesses of C5.0

Strengths	Weaknesses
1. An all-purpose classifier that does well on most problems.	**1.** Decision tree models are often biased toward splits on features having a large number of levels.
2. Highly automatic learning process, which can handle numeric or nominal features, as well as missing data.	**2.** It is easy to overfit or underfit the model.
3. Excludes unimportant features.	**3.** Can have trouble modeling some relationships due to reliance on axis-parallel splits.
4. Can be used on both small and large datasets.	**4.** Small changes in the training data can result in large changes to decision logic.
5. Results in a model that can be interpreted without a mathematical background (for relatively small trees).	**5.** Large trees can be difficult to interpret and the decisions they make may seem counterintuitive.
6. More efficient than other complex models.	

References

1. Machine Learning with R, Second Edition, Brett Lantz, PACKT Publishing.