BITS Pilani

Pilani Campus

Machine Learning

AIML CZG565

M5 : Decision Tree Classifier

Course Faculty of M.Tech Cluster

BITS – CSIS - WILP

Machine Learning

Disclaimer and Acknowledgement

- These content of modules & context under topics are planned by the course owner Dr. Sugata, with grateful acknowledgement to many others who made their course materials freely available online
- We here by acknowledge all the contributors for their material and inputs.
- We have provided source information wherever necessary
- Students are requested to refer to the textbook w.r.t detailed content of the presentation deck shared over canvas
- We have reduced the slides from canvas and modified the content flow to suit the requirements of the course and for ease of class presentation

Slide Source / Preparation / Review:
From BITS Pilani WILP**:** Prof.Sugata, Prof.Chetana, Prof.Rajavadhana, Prof.Monali, Prof.Anita, Prof.Sangeetha, Prof.Swarna
External: CS109 and CS229 Stanford lecture notes, Dr.Andrew NG and many others who made their course materials freely available online

Course Plan

M1	Introduction
M2	Machine learning Workflow
M3	Linear Models for Regression
M4	Linear Models for Classification
M5	Decision Tree
M6	Instance Based Learning
M7	Support Vector Machine
M8	Bayesian Learning
M9	Ensemble Learning
M10	Unsupervised Learning
M11	Machine Learning Model Evaluation/Comparison

Module 5

- Information Theory
- Entropy Based Decision Tree Construction
- Avoiding Overfitting – Not included in mid term exam
- Minimum Description Length – Not included in mid term exam
- Handling Continuous valued attributes – Not included in mid term exam

Decision Tree Classifier

BITS Pilani

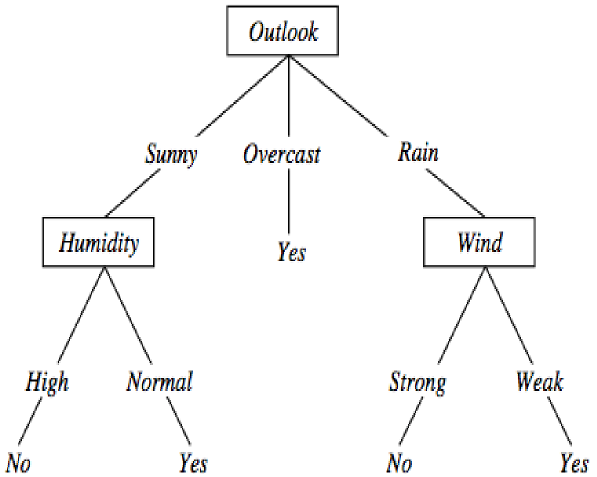
Decision trees

- Decision Trees is one of the most widely used and practical methods of classification
- Method for approximating discrete-valued functions
- Learned functions are represented as decision trees (or if-then-else rules)
- Expressive hypotheses space

Types of Classification

Decision Theory: Interpretation

Model Building



$$IF\ OUTLOOK = Overcast\ THEN\ PLAY = Yes$$

ELSE

$$IF\ OUTLOOK = Rain\ AND\ \ WIND = Strong\ \ THEN\ PLAY = No$$

Logistic regression, SVMs , tree based classifiers (e.g. decision tree) Traditional neural networks, Nearest neighbor

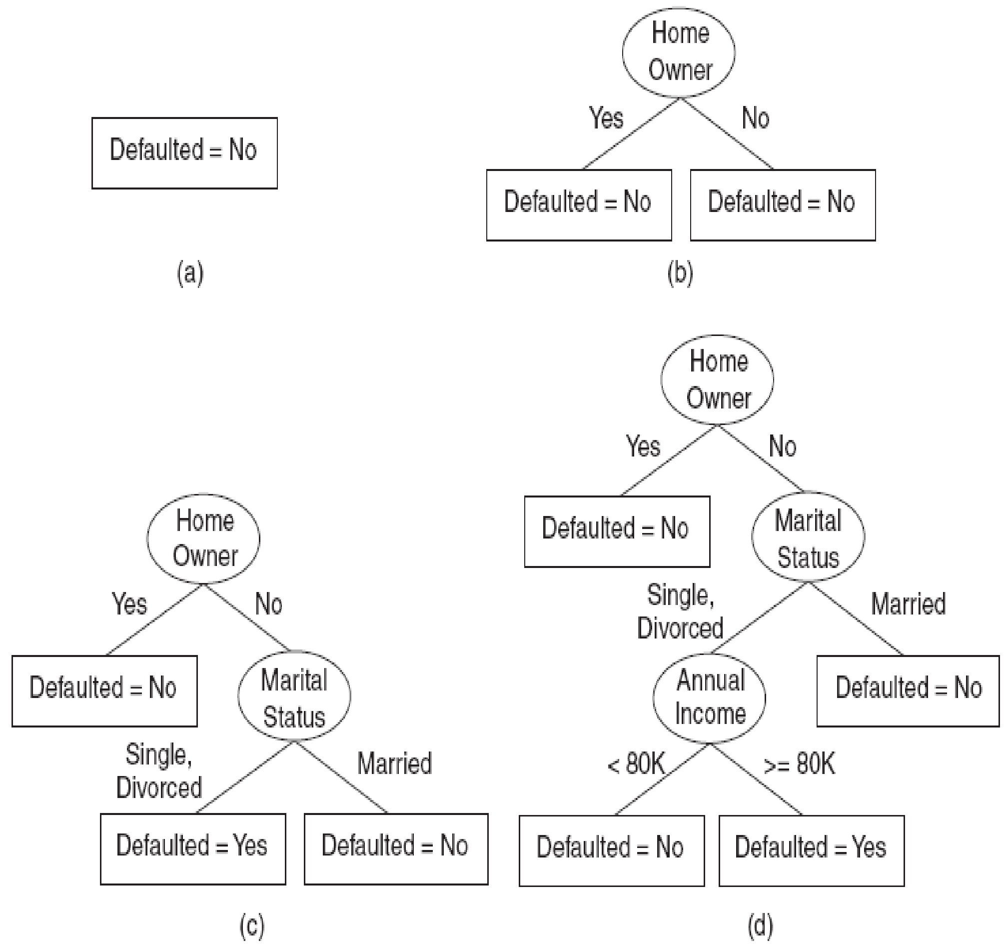
Sky	AirTemp	Humidity	Wind	Forecast	Enjoy Sport?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	No
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	Normal	Breeze	Same	Yes
Sunny	Hot	Normal	Breeze	Same	No
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes
Rainy	Warm	Normal	Breeze	Same	Yes

BITS Pilani, Pilani Campus

Decision Tree Construction: Example

binary categorical continuous class

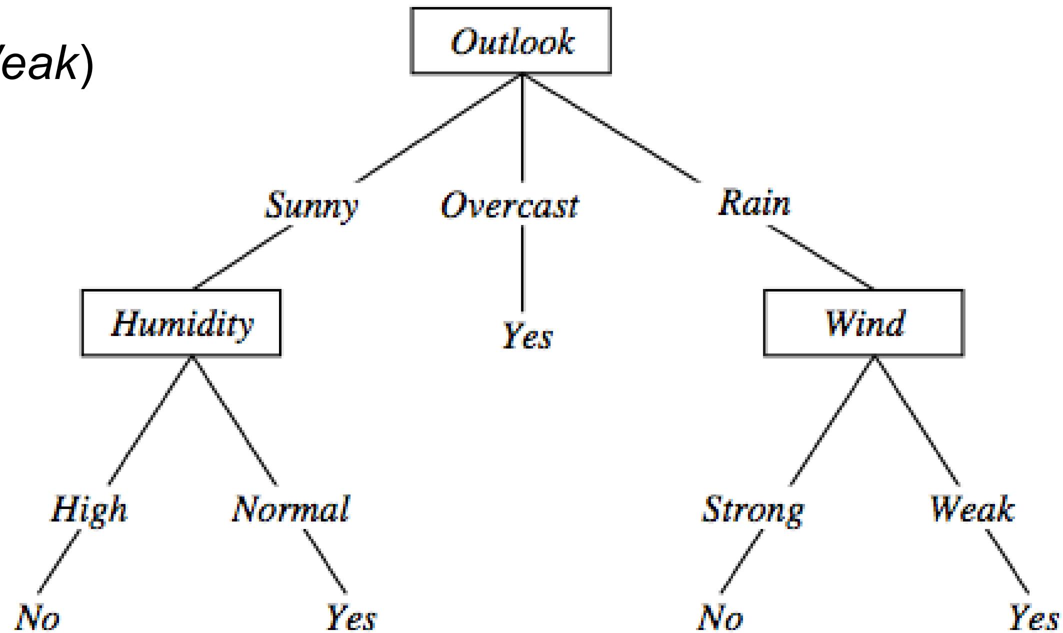
Tid	Home Owner	Marital Status	Annual Income	Defaulted Borrower
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



Decision Trees – Representation & Expressiveness

Decision trees represent a disjunction of conjunctions on constraints on the value of attributes:

$(Outlook = Sunny \wedge Humidity = Normal) \vee$
$(Outlook = Overcast) \vee$
$(Outlook = Rain \wedge Wind = Weak)$



$\langle Outlook{=}Sunny, Temp{=}Hot, Humidity{=}High, Wind{=}Strong \rangle$ **No**

Measure of Information

- The amount of information (surprise element) conveyed by a message is inversely proportional to its probability of occurrence. That is $I_k \, \alpha \, \frac{1}{p_k}$

- The mathematical operator satisfies above properties is the logarithmic operator. $I_k = \log_r \frac{1}{p_k} \;\; units$

Entropy

- Entropy of discrete random variable $X=\{x_1, x_2 \ldots x_n\}$

$$\mathrm{H}(X) = \mathrm{E}[\mathrm{I}(X)] = \mathrm{E}[-\log(\mathrm{P}(X))].$$

since: $\log_2(1/P(event)) = -\log_2 P(event)$

- As uncertainty increases, entropy increases
- Entropy across all values

$$\boxed{\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)}$$

Entropy in general

- Entropy measures the amount of information in a random variable

$$H(X) = -p_+ \log_2 p_+ - p_- \log_2 p_- \qquad X = \{+, -\}$$

for binary classification [two-valued random variable]

$$H(X) = -\sum_{i=1}^{c} p_i \log_2 p_i = \sum_{i=1}^{c} p_i \log_2 1/p_i \qquad X = \{i, \ldots, c\}$$

for classification in *c* classes

Entropy in binary Classification

- Entropy measures the *impurity* of a collection of examples. It depends from the distribution of the random variable *p.*
  - S is a collection of training examples
  - p_+ the proportion of positive examples in S
  - p_- the proportion of negative examples in S

- $Entropy\ (S) \equiv -p_+ \log_2 p_+ - p_- \log_2 p_-$ $[0 \log_2 0 = 0]$
- $Entropy\ ([14+, 0-]) = -14/14 \log_2 (14/14) - 0 \log_2 (0) = 0$
- $Entropy\ ([9+, 5-]) = -9/14 \log_2 (9/14) - 5/14 \log_2 (5/14) = 0.94$
- $Entropy\ ([7+, 7-]) = -7/14 \log_2 (7/14) - 7/14 \log_2 (7/14) =$
- $= 1/2 + 1/2 = 1$ $[\log_2 1/2 = -1]$

Note: the log of a number < 1 is negative, $0 \leq p \leq 1$, $0 \leq entropy \leq 1$

https://www.easycalculation.com/log-base2-calculator.php

Information gain as entropy reduction

- *Information gain* is the *expected* reduction in entropy caused by partitioning the examples on an attribute.
- The higher the information gain the more effective the attribute in classifying training data.
- Expected reduction in entropy knowing *A*

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

Values(*A*) possible values for *A*

S_v subset of *S* for which *A* has value *v*

Maximizing the gain = minimizing the weighted average impurity measure of children nodes

Example

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

Example: Information gain

Day	Outlook	Temperature	Humidity	Wind	PlayTennis
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

- Let
  - $Values(Wind) = \{Weak, Strong\}$
  - $S = [9+, 5-]$
  - $S_{Weak} = [6+, 2-]$
  - $S_{Strong} = [3+, 3-]$
- Information gain due to knowing *Wind*:

$$Gain(S, Wind) = Entropy(S) - 8/14\ Entropy(S_{Weak}) - 6/14\ Entropy(S_{Strong})$$

$$= 0.94 - 8/14 \times 0.811 - 6/14 \times 1.00$$

$$= 0.048$$

Example

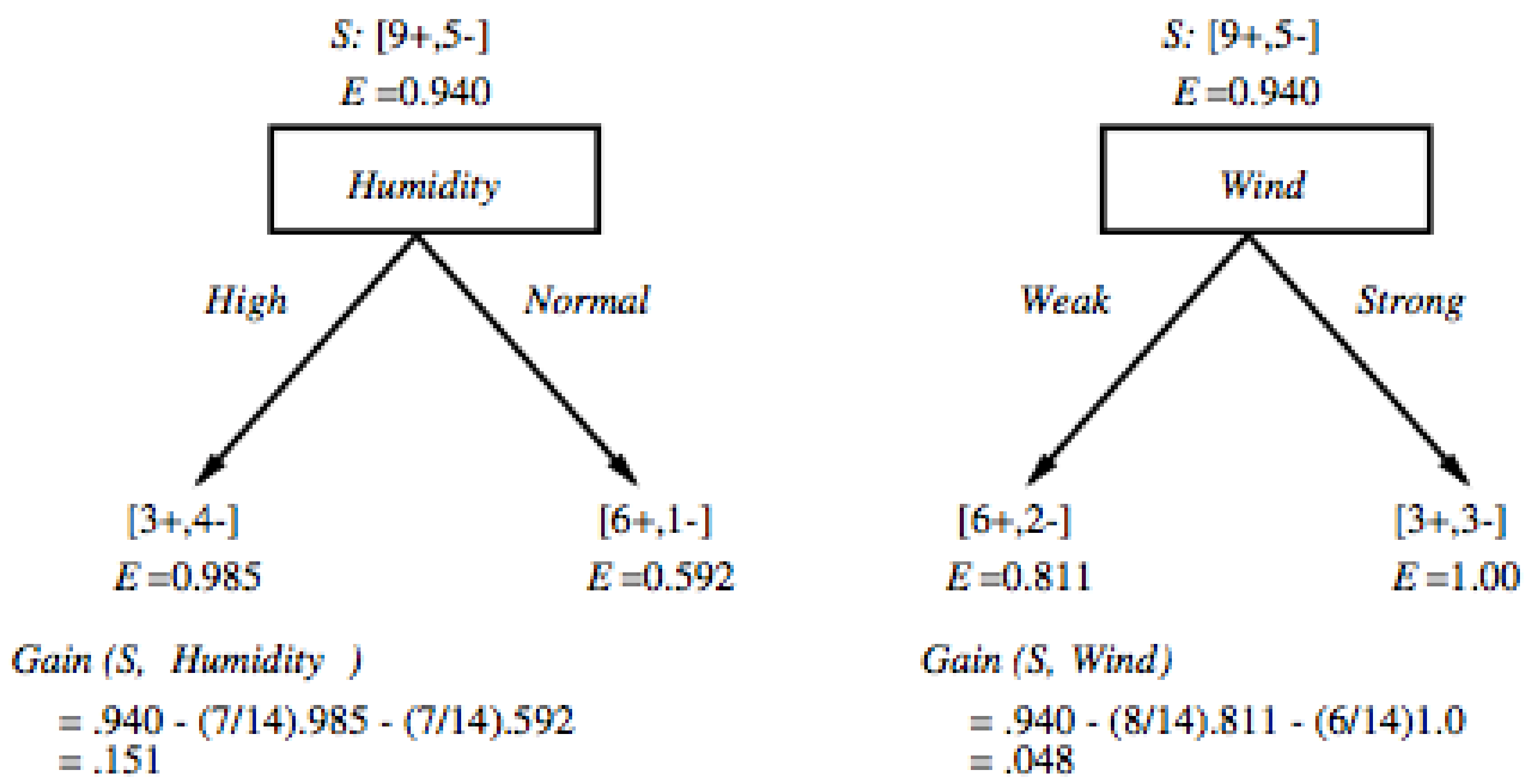
Day	Outlook	Temperature	Humidity	Wind	PlayTennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	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

Splitting Criteria

- Splitting criterion tells us which attribute to test at node N by determining the "best" way to separate or partition the tuples in D into individual classes
- Also, tells us which branches to grow from node N with respect to the outcomes of the chosen test.
- Splitting criterion indicates the splitting attribute and may also indicate either a split-point or a splitting subset
- Partition is **"Pure"** i.e. all the tuples in it belong to the same class.
- Splitting attribute **A** can be:
  - **A is discrete valued:** outcome correspond to the known values of A.
  - **A is continuous-valued:** the test at node N has two possible outcomes, corresponding to the conditions A<=split_point and A > split_point, respectively.
  - **A is dicrete-values and Binary tree:** The test at node N is of the form "$A \in S_A$?," where S_A is the splitting subset for A,. If a given tuple has value a_j of A and if $a_j \in S_A$, then the test at node N is satisfied.

Which attribute is the best classifier?

Which attribute is the best classifier?



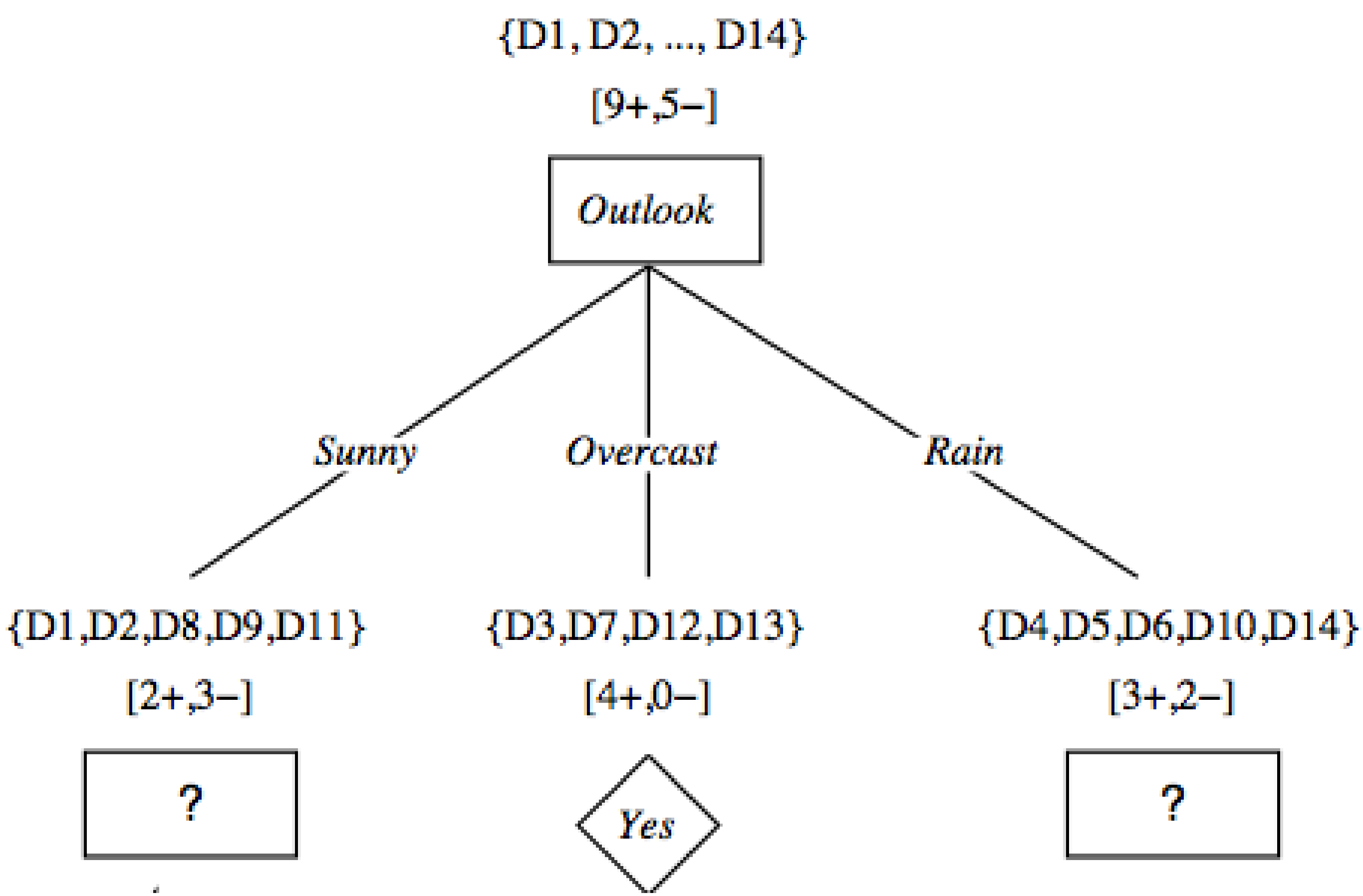
First step: which attribute to test at the root?

- Which attribute should be tested at the root?
  - *Gain*(*S*, *Outlook*) = **0.246**
  - *Gain*(*S*, *Humidity*) = 0.151
  - *Gain*(*S*, *Wind*) = 0.084
  - *Gain*(*S*, *Temperature*) = 0.029
- ***Outlook*** provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Outlook*
  - partition the training samples according to the value of *Outlook*

Attribute Selection Measures

- Also, known as Splitting rules
- Measure is a heuristic for selecting the splitting criterion that “best” separates a given data partition D, of class-labeled training tuples into individual classes.
- Partition should be pure (i.e., all the tuples that fall into a given partition would belong to the same class).
- Provides ranking to each attribute of training tuple, and the attribute having “best” score is chosen as the splitting attribute.

After first step



Second step

- Working on *Outlook=Sunny* node:

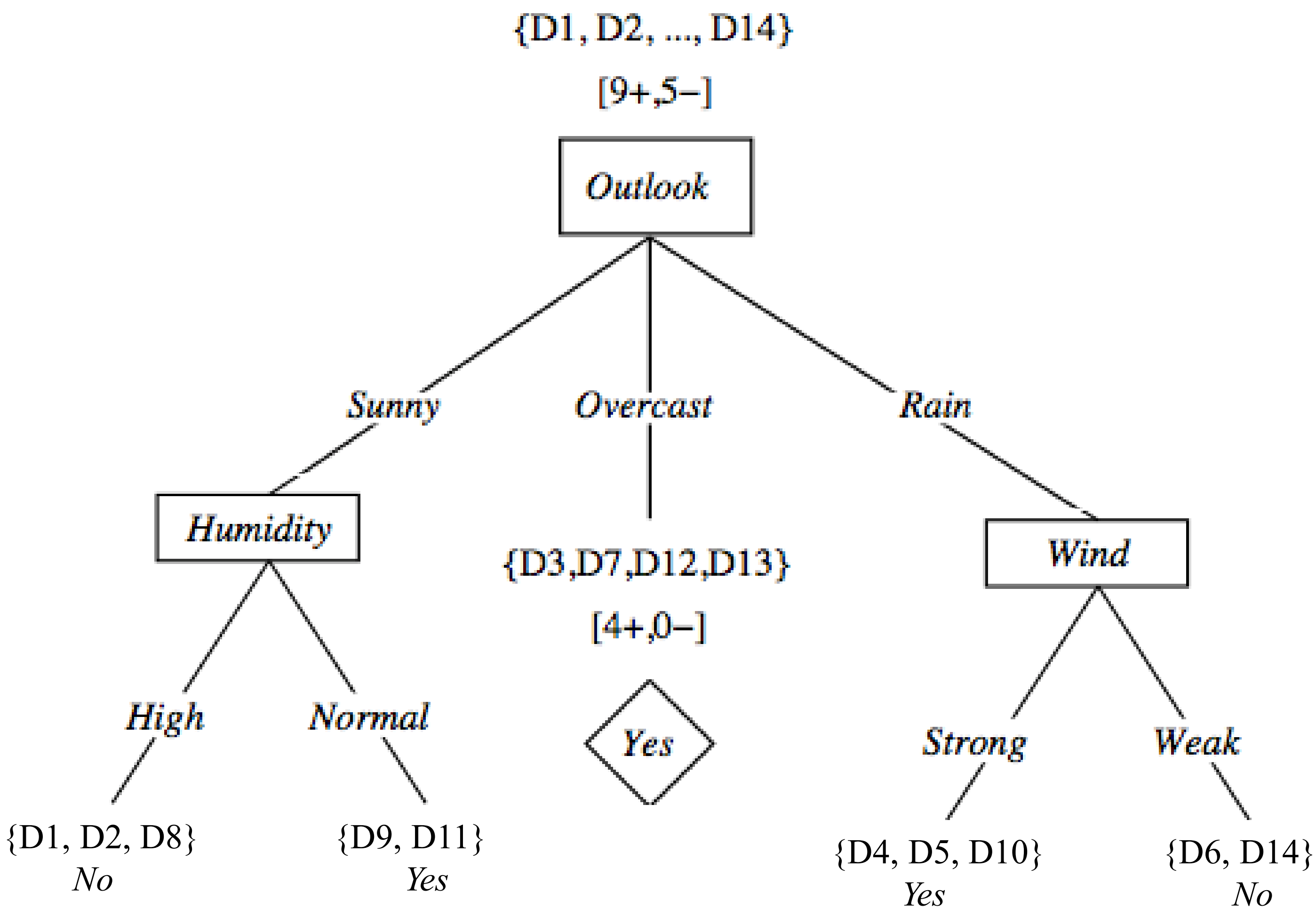
  $Gain(S_{Sunny}, Humidity) = 0.970 - 3/5 \times 0.0 - 2/5 \times 0.0 = \mathbf{0.970}$

  $Gain(S_{Sunny}, Wind) = 0.970 - 2/5 \times 1.0 - 3.5 \times 0.918 = 0\ .019$

  $Gain(S_{Sunny}, Temp.) = 0.970 - 2/5 \times 0.0 - 2/5 \times 1.0 - 1/5 \times 0.0 = 0.570$

- ***Humidity*** provides the best prediction for the target
- Lets grow the tree:
  - add to the tree a successor for each possible value of *Humidity*
  - partition the training samples according to the value of *Humidity*

Second and third steps



Attribute Selection Measure: Information Gain (ID3/C4.5) - Summary

- Select the attribute with the highest information gain
- This attribute minimizes the expected number of tests needed to classify a given tuple.
- Let p_i be the probability that a tuple in D belongs to class C_i, estimated by $|C_{i,D}|/|D|$, m is the number of distinct classes, v is the number of distinct values in an attribute.
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (after using A to split D into v partitions) to classify D

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

  - The smaller the expected information required, greater the purity of the partitions.

Attribute Selection Measure: Information Gain (ID3/C4.5) - Summary

- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

- Defined as the difference between the original information requirement (i.e. based on just the proportion of the classes) and the new requirement (i.e. obtained after partitioning of attribute A).
- Attribute with highest information gain Gain(A), is chosen as the splitting attribute at node N.

ID3: algorithm -Summary

ID3(X, T, *Attrs*) X: training examples:
T: target attribute (e.g. *PlayTennis*),
Attrs: other attributes, initially all attributes

Create *Root* node
If all X's are +, *return* *Root* with class +
If all X's are –, *return* *Root* with class –
If *Attrs* is empty *return* *Root* with class most common value of T in X
else
  $A \leftarrow$ best attribute; decision attribute for Root $\leftarrow A$
  For each possible value v_i of A:
  - add a new branch below *Root*, for test $A = v_i$
  - $X_i \leftarrow$ subset of X with $A = v_i$
  - *If* X_i is empty *then* add a new leaf with class the most common value of T in X
    else add the subtree generated by ID3(X_i, T, *Attrs* – {A})
return *Root*

Prefer shorter hypotheses: Occam's razor

- Why prefer shorter hypotheses?
- Arguments in favor:
  - There are fewer short hypotheses than long ones
  - If a short hypothesis fits data unlikely to be a coincidence
  - Elegance and aesthetics
- Arguments against:
  - Not every short hypothesis is a reasonable one.
- Occam's razor says that when presented with competing hypotheses that make the **same** predictions, one should select the solution which is simple"

Decision trees

- Advantages:
  - – Inexpensive to construct
  - – Extremely fast at classifying unknown records
  - – Easy to interpret for small-sized trees
  - – Can easily handle redundant or irrelevant attributes (unless the attributes are interacting)
- Disadvantages:
  - – Space of possible decision trees is exponentially large.
  - -- Greedy approaches are often unable to find the best tree.
  - – Does not take into account interactions between attributes
  - – Each decision boundary involves only a single attribute

Decision Trees – Problem Type 1

Entropy of the Decision
Feature EnjoySport:

= **Entropy** $_{\textbf{DecisionClass}}$**(S) = Σ-p(c) log$_2$ p(c)**

= E(3,1) = – (3/4) $\log_2$ (3/4) – (1/4) $\log_2$ (1/4) = 0.4421+ 0.5 = 0.94

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes
Rainy	Warm	Normal	Breeze	Same	Yes

Note:

- Class balancing is recommended before building
- Value = 0 to log base 2 k, where k is the number of classes you have.

Decision Trees – Problem Type 1

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes

Entropy of the Decision Feature EnjoySport:

= **Entropy** $_{\textbf{DecisionClass}}$**(S) =** $\boldsymbol{\Sigma\text{-}p(c)\log_2 p(c)}$

= $E(3,1) = -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.4421 + 0.5 = 0.94$

Sky	Enjoy	Not Enjoy
Sunny	3	0
Rainy	0	1

Entropy of the attribute Sky:

= p(Sunny) . E(3,0) + p(Rainy). E(0,1)

= $(3/4)[-(3/3)\log_2(3/3) - (0/3)\log_2(0/3)] + (1/4)[-(0/1)\log_2(0/1) - (1/1)\log_2(1/1)]$

= 0

Air Temp	Enjoy	Not Enjoy
Warm	3	0
Cold	0	1

Entropy of the attribute AirTemp:

= p(Warm) . E(3,0) + p(Cold). E(0,1)

= $(3/4)[-(3/3)\log_2(3/3) - (0/3)\log_2(0/3)] + (1/4)[-(0/1)\log_2(0/1) - (1/1)\log_2(1/1)]$

= 0

Decision Trees – Problem Type 1

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes

Entropy of the Decision Feature EnjoySport:

= **Entropy** $_{\textbf{DecisionClass}}$**(S) = Σ-p(c)** $\mathbf{\log_2 p(c)}$

= $E(3,1) = -(3/4)\log_2(3/4) - (1/4)\log_2(1/4) = 0.94$

Humidity	Enjoy	Not Enjoy
Normal	1	0
High	2	1

Entropy of the attribute Humidity :

= p(Normal) . E(1,0) + p(High). E(2,1)

= $(1/4)[-(1/1)\log_2(1/1) - (0/1)\log_2(0/1)] + (3/4)[-(2/3)\log_2(2/3) - (1/3)\log_2(1/3)]$

= 0 + ¾ [0.39 + 0.52] = 0.68

Wind	Enjoy	Not Enjoy
Strong	3	1

Entropy of the attribute Wind :

= p(Strong) . E(3,1)

= $(4/4)[-(3/4)\log_2(3/4) - (1/4)\log_2(1/4)]$

= 1(0.94) = 0.94

Decision Trees – Problem Type 1

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes

Entropy of the Decision Feature EnjoySport:

= **Entropy** $_{\textbf{DecisionClass}}$**(S) =** $\boldsymbol{\Sigma}$**-p(c) log$_2$ p(c)**

= E(3,1) = – (3/4) $\log_2$ (3/4) – (1/4) $\log_2$ (1/4) = 0.94

Forecast	Enjoy	Not Enjoy
Same	2	0
Change	1	1

Entropy of the attribute Forecast :

= p(Same) . E(2,0) + p(Change). E(1,1)

= (2/4) [– (2/2) $\log_2$ (2/2) – (0/2) $\log_2$ (0/2)] + (2/4) [– (1/2) $\log_2$ (1/2) – (1/2) $\log_2$ (1/2)]

= 0 + 0.5 = 0.5

Decision Trees – Problem Type 1

Entropy of the Decision Feature EnjoySport: 0.94

Entropy of the attribute Sky: 0

Entropy of the attribute AirTemp: 0

Entropy of the attribute Humidity: 0.68

Entropy of the attribute Wind: 0.94

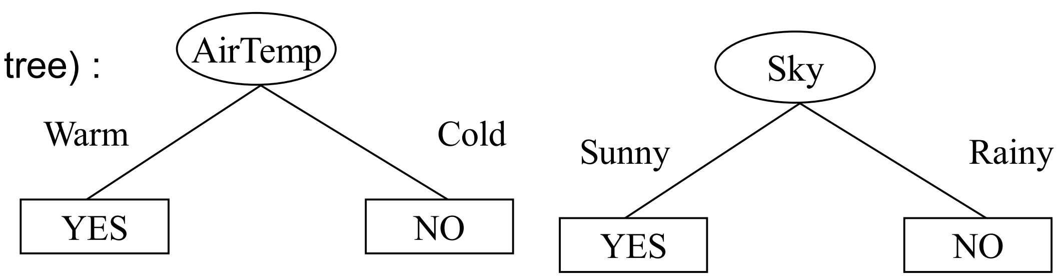
Entropy of the attribute Forecast: 0.5

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes

Attribute that generalizes the model faster = That which reduces the randomness if used to decide = That which have the highest info gain = SKY & AIRTEMP

Decision Tree constructed

(both are equally correct decision tree) :

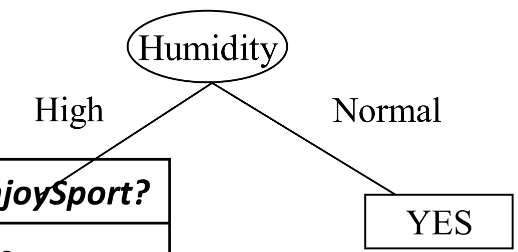


Decision Trees – Problem Type 2

(If there are heterogeneous nodes, continue the decision tree construction)

Sub Problem which has more scope to Model: None

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes



Sky	AirTemp	Wind	Forecast	*EnjoySport?*
Sunny	Warm	Strong	Same	Yes
Rainy	Cold	Strong	Change	No
Sunny	Warm	Strong	Change	Yes

Assumed sample data in the node

Repeat the step again for the remaining features till:

- All training data are learnt
- No more learning is possible
- Constraints posed by the hyperparameter satisfied (More on this will be discussed post mid term for handling overfitting)

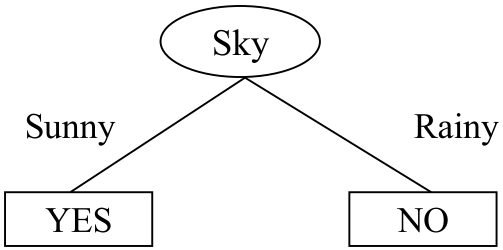
Convergence of Algorithm

- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Decision Trees – Problem Type 3

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes
Rainy	Warm	Normal	Breeze	Same	????

Predict the class for new input instance →



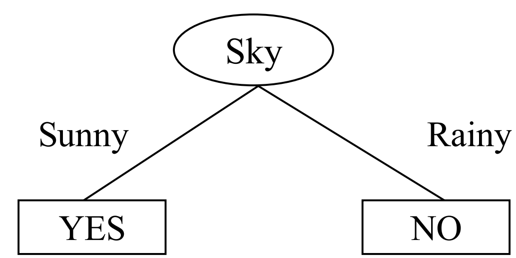
Problem Type 4 – Evaluate the model Confusion Matrix

- The evaluation discussed under module 4 is applicable here
- By default , model's predictive quality is evaluated **only by the unseen validation/test instances.**
- When training performance (eg., accuracy) is required for comparison, the training instances are used for assessment

Actual class\Predicted class	C_1	$\neg C_1$
C_1	**True Positives (TP)**	**False Negatives (FN)**
$\neg C_1$	**False Positives (FP)**	**True Negatives (TN)**

•$CM_{i,j}$ indicates the number of tuples of class i that were labelled by the classifier as class j.

Sky	AirTemp	Humidity	Wind	Forecast	EnjoySport ?
Sunny	Warm	Normal	Strong	Same	Yes
Sunny	Warm	High	Strong	Same	Yes
Rainy	Cold	High	Strong	Change	No
Sunny	Warm	High	Strong	Change	Yes
Rainy	Warm	Normal	Breeze	Same	Yes



Applications

Suited for following classification problems:

- Applications whose Instances are represented by attribute-value pairs.
- The target function has discrete output values
- Disjunctive descriptions may be required
- The training data may contain missing attribute values

Real world applications

- Biomedical applications
- Manufacturing
- Banking sector
- Make-Buy decisions

Practice Exercises (for Students)

1. For this similar problem discussed in class build a decision tree classifier using ID3 algorithm ie., Information gain and entropy measures. Grow the complete decision tree.
2. Can "Object ID" be used as one of key feature to construct the tree? Justify your answer on the results of entropy calculated.
3. Use the test data to construct the confusion matrix and find the precision , recall, F-score.
4. Compare the training accuracy vs test accuracy for the model built in part 1)

	Training Data			
Object ID	Shape	Colour	Size	Action
OB101	Round	Green	Small	Reject
OB102	Square	Black	Big	Allow
OB103	Square	Brown	Big	Allow
OB104	Round	Brown	Small	Reject
OB105	Square	Green	Big	Allow
OB105	Square	Brown	Small	Reject
OB106	Oval	Green	Big	Reject
OB107	Oval	Brown	Small	Allow
OB108	Oval	Green	Small	Reject

	Test Data / Prune Set/ Validation Set			
Object ID	Shape	Colour	Size	Action
OB109	Oval	Black	Small	Reject
OB110	Round	Brown	Big	Allow
OB111	Square	Brown	Big	Allow
OB112	Oval	Green	Small	Allow

Additional References

Decision Tree

- https://www.youtube.com/watch?v=eKD5gxPPeY0&list=PLBv09BD7ez_4temBw7vLA19p3tdQH6FYO&index=1

Thank you !

Required Reading for completed session :

T1 - Chapter # 6 (Tom M. Mitchell, Machine Learning)

R1 – Chapter # 3,#4 (Christopher M. Bhisop, Pattern Recognition & Machine Learning)

Important Note to Students:

- Check the canvas announcement for any exam related details/ question pattern/ sample questions.
- Handout mapped prescribed book sections, class discussions , practice exercises shared in all the slides must be revised for the exam preparation
- Since the mid term exam is CLOSED BOOK mode, no reference material is allowed and hence formula needs to be learnt by the students
- Queries may not be answered once the exam starts.
- **Kindly Plan ahead to get your preparation queries resolved before the EXAM START DATE of your batch**.