7. Classification evaluation metrics and Decision trees

Shabana K M
PhD Research Scholar
Computer Science and Engineering

IIT Palakkad

03 October 2021

INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

Recap

- Naive Bayes classifier
  - Estimating parameters for discrete and continuous attributes
  - Text classification

Thresholding

- classification models such as logistic regression return a probability for binary classification
- map probability value to binary class by defining a **classification threshold**
  - value above the threshold indicates `positive` class
  - value below indicates `negative` class
- thresholds are problem-dependent and therefore must be tuned

Classification accuracy

- ratio of correct predictions to total predictions made

$$\text{accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}}$$

- often represented in percentage
- easy to calculate and intuitive to understand

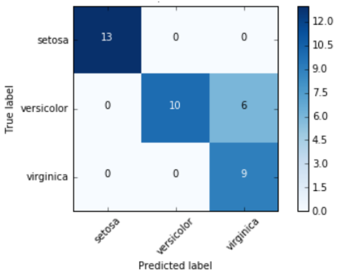
$$\text{error-rate} = 1 - \text{accuracy}$$

- classification accuracy alone can be misleading if
  - there is an unequal number of observations in each class, or
  - there are more than two classes in the dataset

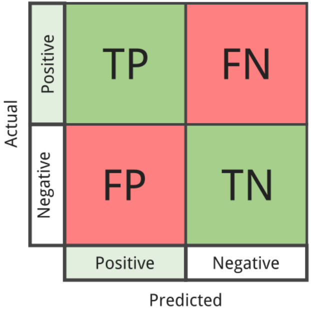
Confusion matrix

- technique for summarizing the performance of a classification algorithm
- can give a better idea of what the model is getting right and what types of errors it is making

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

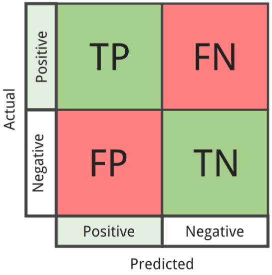


Confusion matrix



- **True Positive (TP):** Predicted `True` and `True` in reality
- **True Negative (TN):** Predicted `False` and `False` in reality.
- **False Positive (FP):** Predicted `True` and `False` in reality
- **False Negative (FN):** Predicted `False` and `True` in reality

Precision and recall



Accuracy

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP}$$

Precision

What proportion of positive identifications was actually correct?

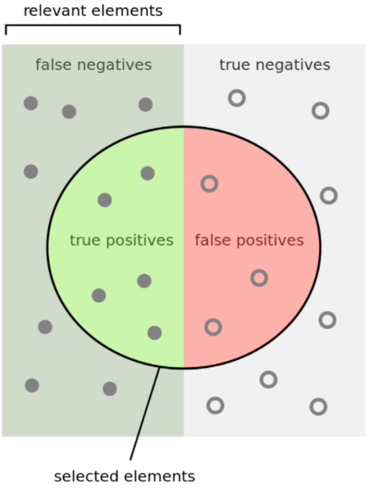
$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall

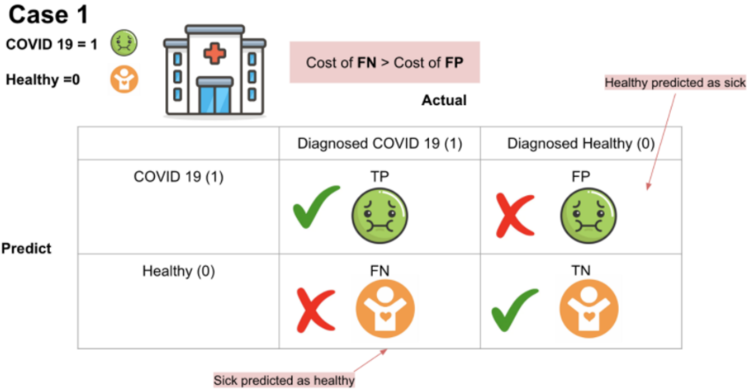
What proportion of actual positives was identified correctly?

$$\text{Recall} = \frac{TP}{TP + FN}$$

Precision and recall



Precision vs Recall



Precision vs Recall

Case 2

Spam = 1

Not Spam =0

Cost of **FP** > Cost of **FN**

Not spam predicted as spam

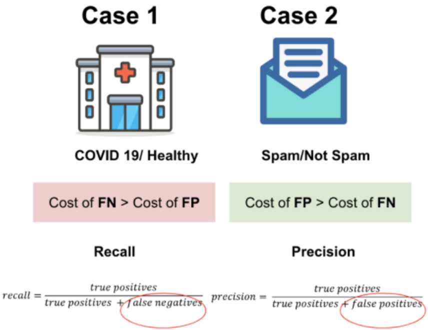
Actual

Predict		Spam (1)	Not Spam (0)
	Spam (1)	TP	FP
	Not Spam (0)	FN	TN

Spam predicted as not spam

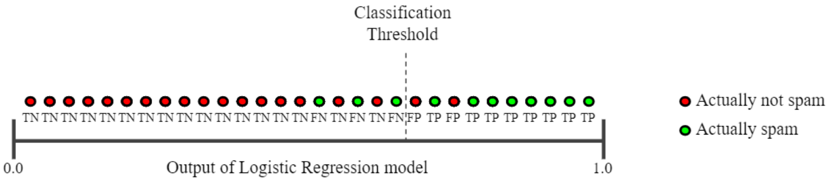
Precision vs Recall

- cost of false negatives not always the same as that of false positives
- the more the `false positives`, the lower the `precision`
- the more the `false negatives`, the lower the `recall`



Precision and recall - Tug of war

Classifying email messages as spam or not spam



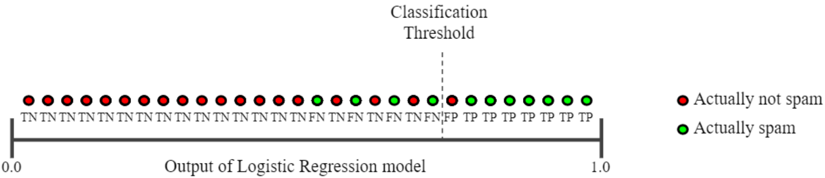
True Positives (TP): 8	False Positives (FP): 2
False Negatives (FN): 3	True Negatives (TN): 17

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{8}{8+2} = 0.8$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{8}{8+3} = 0.73$$

Precision and recall - Tug of war

Increasing classification threshold



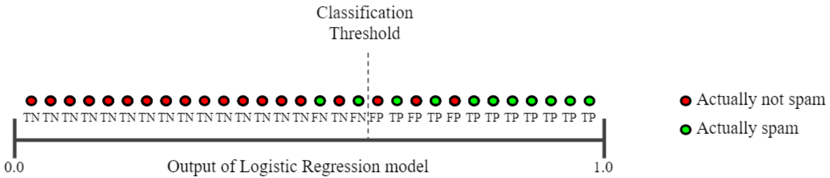
True Positives (TP): 7	False Positives (FP): 1
False Negatives (FN): 4	True Negatives (TN): 18

$$\text{Precision} = \frac{TP}{TP+FP} = \frac{7}{7+1} = 0.88$$

$$\text{Recall} = \frac{TP}{TP+FN} = \frac{7}{7+4} = 0.64$$

Precision and recall - Tug of war

Decreasing classification threshold



True Positives (TP): 9	False Positives (FP): 3
False Negatives (FN): 2	True Negatives (TN): 16

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{9}{9+3} = 0.75$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{9}{9+2} = 0.82$$

F1-score and False Positive Rate

F1 score

- a single score that balances both precision and recall in one number

$$\text{F1-score} = 2 * \frac{precision * recall}{precision + recall}$$

- best value - 1 (perfect precision and recall) and worst - 0
- a good F1 score means the model has low false positives and low false negatives
- can be extended for multi-class classification

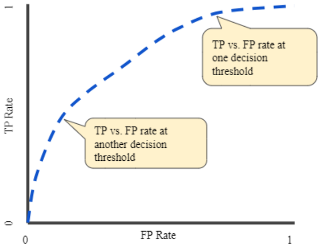
False Positive Rate(FPR)

- ratio between the number of negative items wrongly categorized as positive (false positives) and the total number of actual negative events

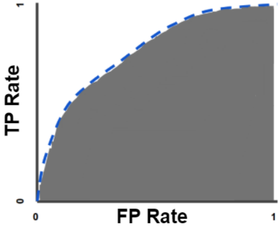
$$\text{FPR} = \frac{FP}{FP + TN}$$

ROC (receiver operating characteristic) curve

- graph showing the performance of a classification model at all classification thresholds
- plots two parameters
  - True Positive Rate (TPR) $= \frac{TP}{TP+FN}$
  - False Positive Rate (FPR) $= \frac{FP}{FP+TN}$
- ROC curve plots TPR vs. FPR at different classification thresholds

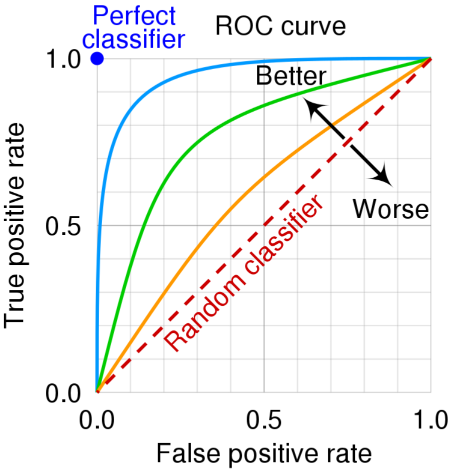


Area under the curve (AUC)

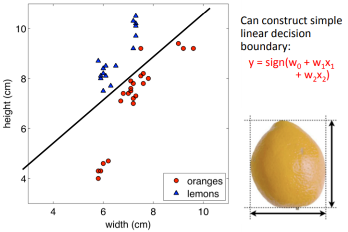


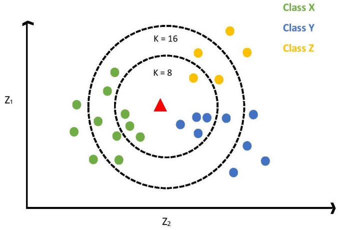
- measures the two-dimensional area underneath the ROC curve from $(0, 0)$ to $(1, 1)$
- provides an aggregate measure of performance across all possible classification thresholds
- value ranges from 0 to 1
  - model whose predictions are 100% wrong has an AUC of 0.0
  - one whose predictions are 100% correct has an AUC of 1.0

ROC curves


Perfect classifier
ROC curve
Better
Worse
Random classifier
1.0
0.5
0.0
0.0
0.5
1.0
True positive rate
False positive rate

Classification techniques we already know





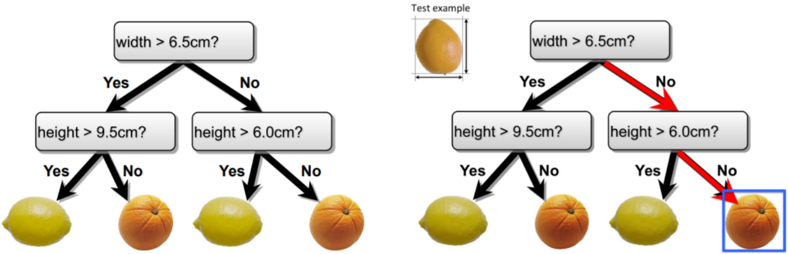
Likelihood Table				
Outlook	Yes	No		
Sunny	3	2	=(5/14)	0.36
Rainy	2	3	=(5/14)	0.36
Overcast	4	0	=(4/14)	0.28
	=(9/14)	=(5/14)		
	0.64	0.36		

P (Sunny | Yes) = 3/9 = 0.33

P (Sunny) = 5/14 = 0.36

P (Yes) = 9/14 = 0.64

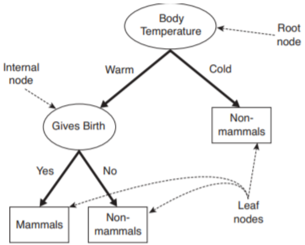
Decision trees - approach



Decision trees

- simple yet popular predictive modeling approach
- model that predicts the value of a target variable by learning simple decision rules inferred from the data features
- can be used for classification as well as regression
- hierarchical data structure that represents data through a divide and conquer strategy
- constructed through algorithmic approaches that identifies ways to split a data set based on different conditions

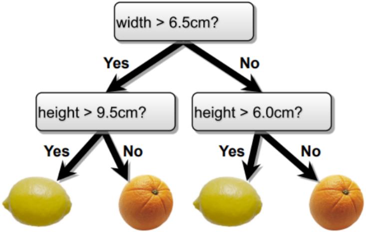
Structure of a decision tree



A decision tree has three types of nodes:

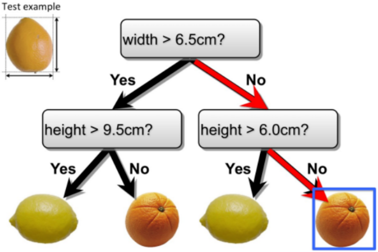
- **Root node:** has no incoming edges and one or more outgoing edges
- **Internal nodes:** has exactly one incoming edge and two or more outgoing edges
- **Leaf or terminal nodes:** has exactly one incoming edge and no outgoing edges

Structure of a decision tree



- The `root node` and each `internal node` tests an attribute
- Branching is determined by the attribute value
  - One branch for each possible outcome of the test
- Leaf nodes are outputs

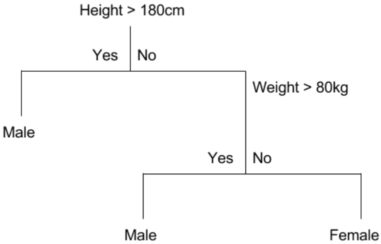
Making predictions using decision trees



1. Start at the root node of the tree
2. Test the attribute specified by this node
3. Move down the tree branch corresponding to the value of the attribute in the given instance
4. Repeat Steps 2 and 3 for the subtree rooted at the new node until a leaf node is reached

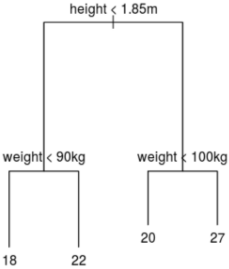
Regression trees vs Classification trees

Classification trees



- target variable can take a discrete set of values

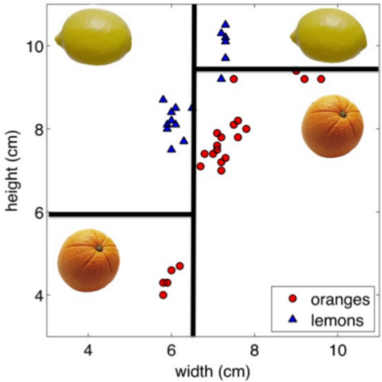
Regression trees

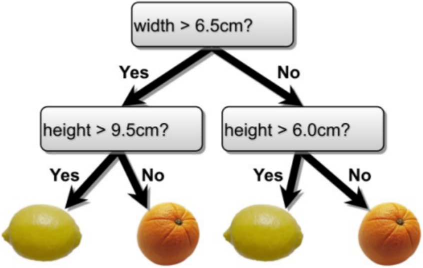


- target variable can take continuous values

Decision boundary

- produces **axis aligned decision boundaries**

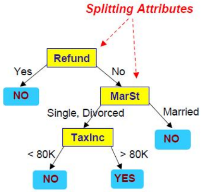




Learning a decision tree

Tid	Refund	Marital Status	Taxable Income	Cheat
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

Training Data

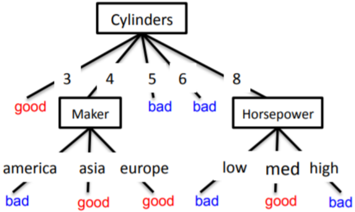


Model: Decision Tree

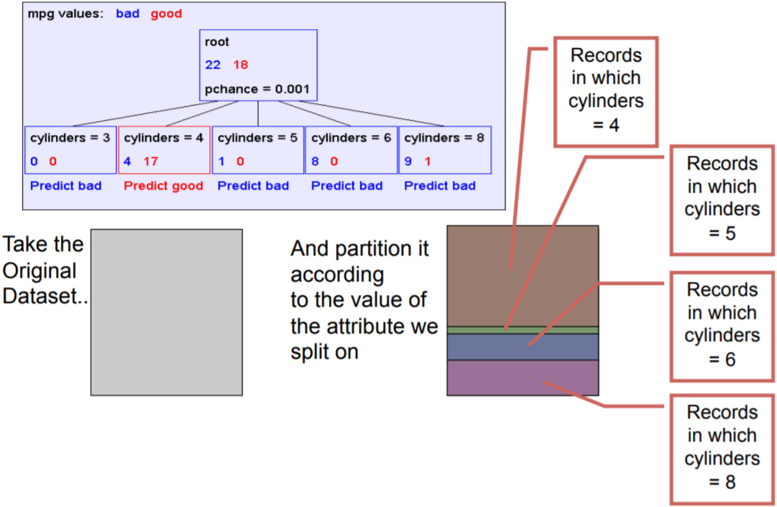
- learning the simplest (smallest) decision tree is an NP-complete problem
- resort to a greedy heuristic
  - start with an empty decision tree
  - split on the **next best attribute**
  - recurse

Greedily learning trees using recursion

mpg	cylinders	displacement	horsepower	weight	acceleration	modelyear	maker
good	4	low	low	low	high	75to78	asia
bad	6	medium	medium	medium	medium	70to74	america
bad	4	medium	medium	medium	low	75to78	europe
bad	8	high	high	high	low	70to74	america
bad	6	medium	medium	medium	medium	70to74	america
bad	4	low	medium	low	medium	70to74	asia
bad	4	low	medium	low	low	70to74	asia
bad	8	high	high	high	low	75to78	america
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
:	:	:	:	:	:	:	:
bad	8	high	high	high	low	70to74	america
good	8	high	medium	high	high	79to83	america
bad	8	high	high	high	low	75to78	america
good	4	low	low	low	low	79to83	america
bad	6	medium	medium	medium	high	75to78	america
good	4	medium	low	low	low	79to83	america
good	4	low	low	medium	high	79to83	america
bad	8	high	high	high	low	70to74	america
good	4	low	medium	low	medium	75to78	europe
bad	5	medium	medium	medium	medium	75to78	europe

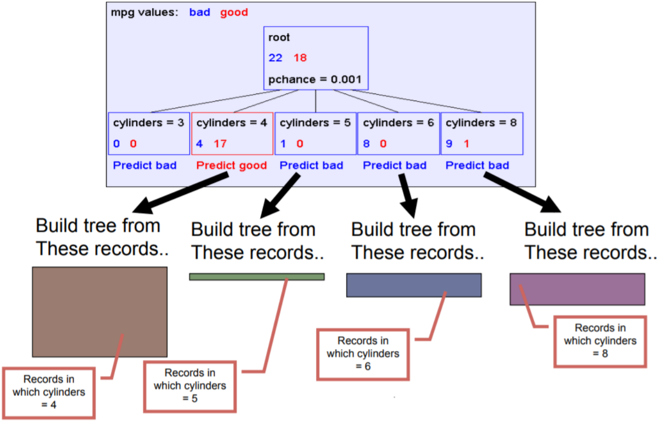


Greedily learning trees using recursion



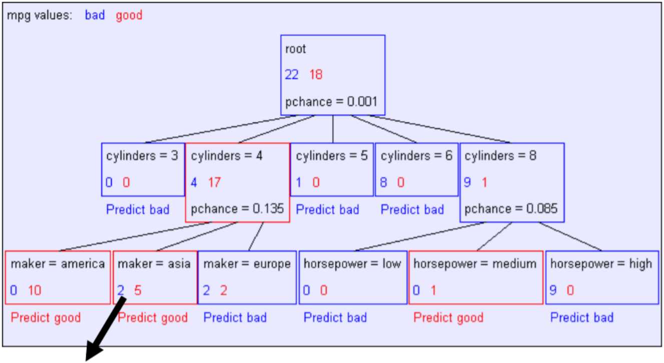
Greedily learning trees using recursion

Recursive step



Greedily learning trees using recursion

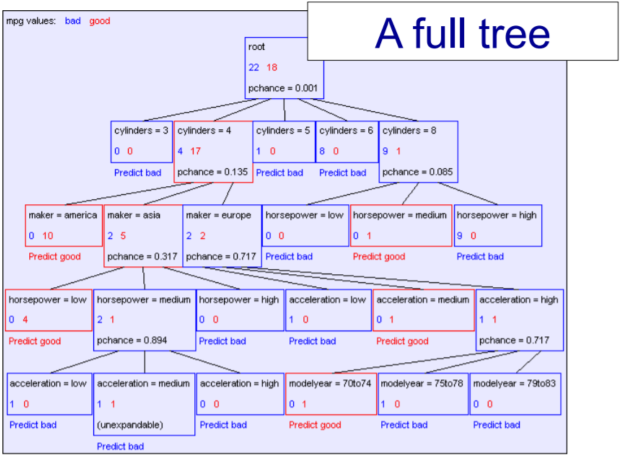
Second level of tree



Recursively build a tree from the seven records in which there are four cylinders and the maker was based in Asia

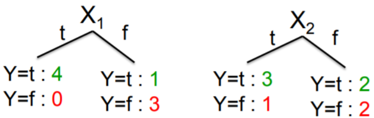
(Similar recursion in the other cases)

Greedily learning trees using recursion



Choosing the best attribute to split - discrete attributes

Should we split on X_1 or X_2?



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F
F	T	F
F	F	F

Idea: Use counts at leaves to define probability distributions, so we can measure uncertainty - *apply concepts from information theory*

Measuring uncertainty

Which attribute is better to split on, X_1 or X_2?

- **Deterministic:** good (all are true or false; just one class in the leaf)
- **Uniform distribution:** bad (all classes in leaf equally probable)
- What about distributions in between?

P(Y=A) = 1/2	P(Y=B) = 1/4	P(Y=C) = 1/8	P(Y=D) = 1/8

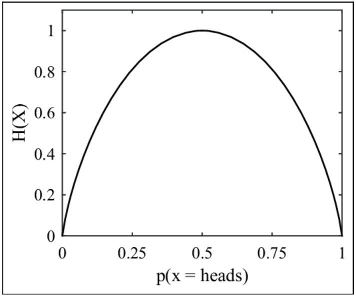
P(Y=A) = 1/4	P(Y=B) = 1/4	P(Y=C) = 1/4	P(Y=D) = 1/4

Quantifying uncertainty

Entropy

`Entropy` of a random variable X,

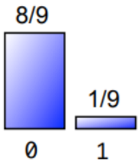
$$H(X) = -\sum_{i=1}^{k} P(X = x_i) log_2(P(X = x_i))$$



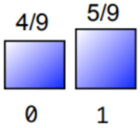
- degree of randomness of elements, or a measure of impurity
- more the uncertainty, more the entropy!

Entropy

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$



$$-\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9} \approx \frac{1}{2}$$



$$-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} \approx 0.99$$

High entropy: Values are less predictable

Low entropy: Values are more predictable

Conditional Entropy

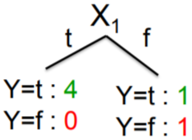
Conditional Entropy $H(Y|X)$ of a random variable Y conditioned on a random variable X,

$$H(Y|X) = -\sum_{i=1}^{k} P(X = x_i)H(Y|X = x_i)$$

Example:

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$H(Y|X_1) = -\, 4/6\,(1 \log_2 1 + 0 \log_2 0)$

$\quad - 2/6\,(1/2 \log_2 1/2 + 1/2 \log_2 1/2)$

$= 2/6$

Splitting the node - discrete attributes

- **Classification tree:** Split the node to minimize entropy
- Let S be set of data points in a node, $c = 1, \dots, C$ are labels:

$$H(S) = -\sum_{c=1}^{C} p(c) log_2(p(c))$$

where $p(c)$ is the proportion of data belonging to class c

  - Entropy $= 0$ if all samples are in the same class
  - Entropy is large if $p(1) = \cdots = p(C)$

P(Y=t) = 5/6
P(Y=f) = 1/6

$H(Y) = - 5/6 \log_2 5/6 - 1/6 \log_2 1/6$
$= 0.65$

X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

Information gain

- The average entropy of a split $S \rightarrow S_1, S_2$

$$\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2)$$

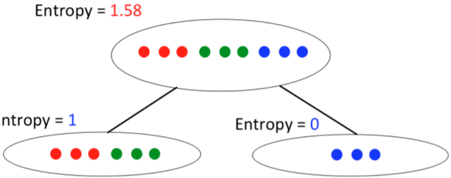
Information gain

- measures how well a given attribute separates the training examples according based on their target value
- $Gain(S, a)$ = expected reduction in entropy of Y due to splitting on attribute a

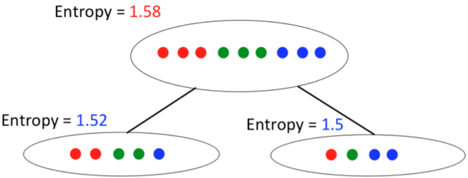
$$\begin{aligned} Gain(S, a) &= H(S) - H(S|a) \\ &= H(S) - \left(\frac{|S_1|}{|S|}H(S_1) + \frac{|S_2|}{|S|}H(S_2)\right) \end{aligned}$$

- the attribute with the maximal information gain is chosen for split

Information gain

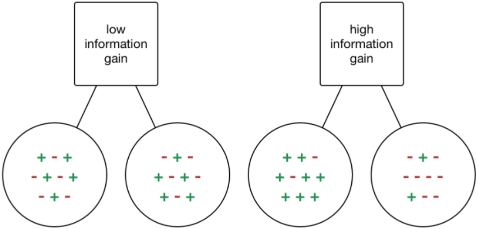


Averaged entropy: 2/3*1 + 1/3*0 = 0.67
Information gain: 1.58 – 0.67 = 0.91



Averaged entropy: 1.51
Information gain: 1.58 – 1.51 = 0.07

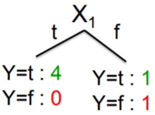
Information gain



Example:

$P(X_1=t) = 4/6$

$P(X_1=f) = 2/6$



X_1	X_2	Y
T	T	T
T	F	T
T	T	T
T	F	T
F	T	T
F	F	F

$$H(Y|X_1) = -\,4/6\,(1 \log_2 1 + 0 \log_2 0)$$
$$-\,2/6\,(1/2 \log_2 1/2 + 1/2 \log_2 1/2)$$
$$= 2/6$$

References

1. https://www.kdnuggets.com/2020/05/model-evaluation-metrics-machine-learning.html
2. https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall
3. https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html
4. https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc
5. https://www.cs.toronto.edu/~urtasun/courses/CSC411_Fall16/06_trees_handout.pdf
6. https://people.csail.mit.edu/dsontag/courses/ml16/slides/lecture11.pdf
7. http://www.stat.ucdavis.edu/~chohsieh/teaching/ECS171_Winter2018/lecture15.pdf
8. https://www.cs.cmu.edu/afs/cs.cmu.edu/academic/class/15381-s06/www/DTs.pdf

Thanks Google for the pictures!