CSCI446/946 Big Data Analytics

Week 7 Advanced Analytical Theory and Methods: Classification

School of Computing and Information Technology
University of Wollongong Australia

Advanced Analytical Theory and Methods: Classification

- Overview of Classification
- Decision Tree
- Naïve Bayes
- Diagnostics of Classifier
- Additional Classification Models

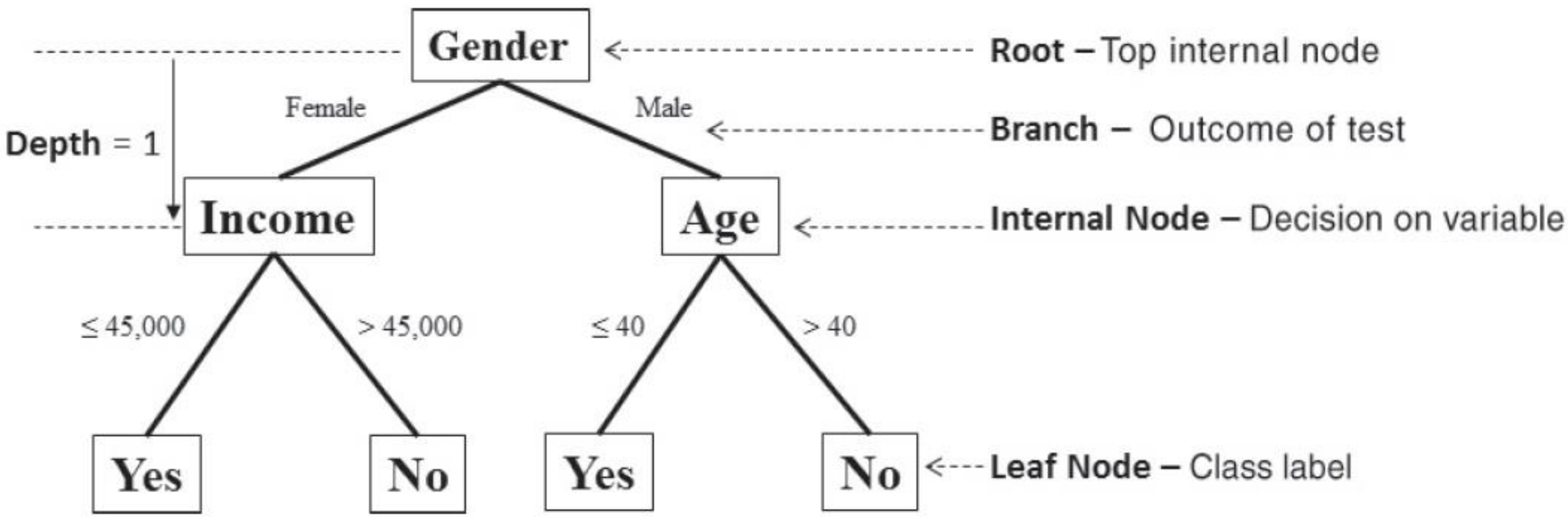
All the figures, tables and codes are from the book "Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data" unless indicated otherwise.

Overview of Classification

- Classification is a fundamental learning method that appears in applications related to data mining
- The primary task performed by classifiers is to assign class labels to new observations
- Classification methods are supervised
  - Start with a training set of labelled observations
  - Predict the outcome for new observations

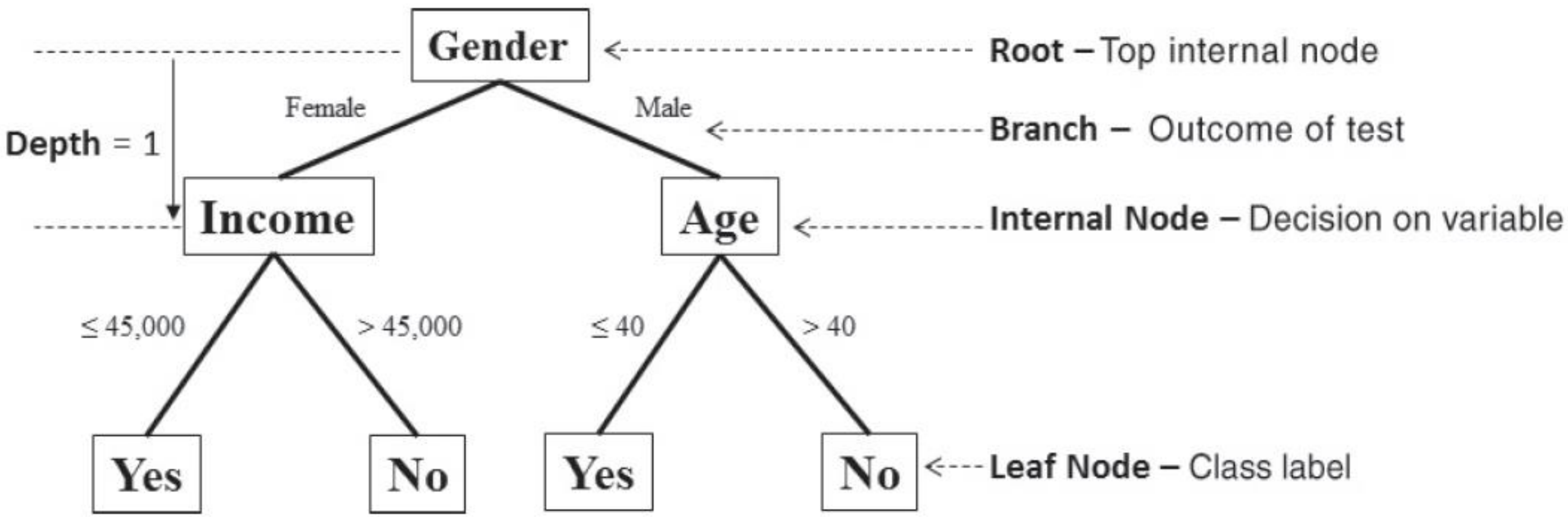
Decision Tree

- A decision tree uses a tree structure to specify sequences of decisions and consequences
- Given input variable $X = \{x_1, x_2, \ldots, x_n\}$, the goal is to predict an output variable Y



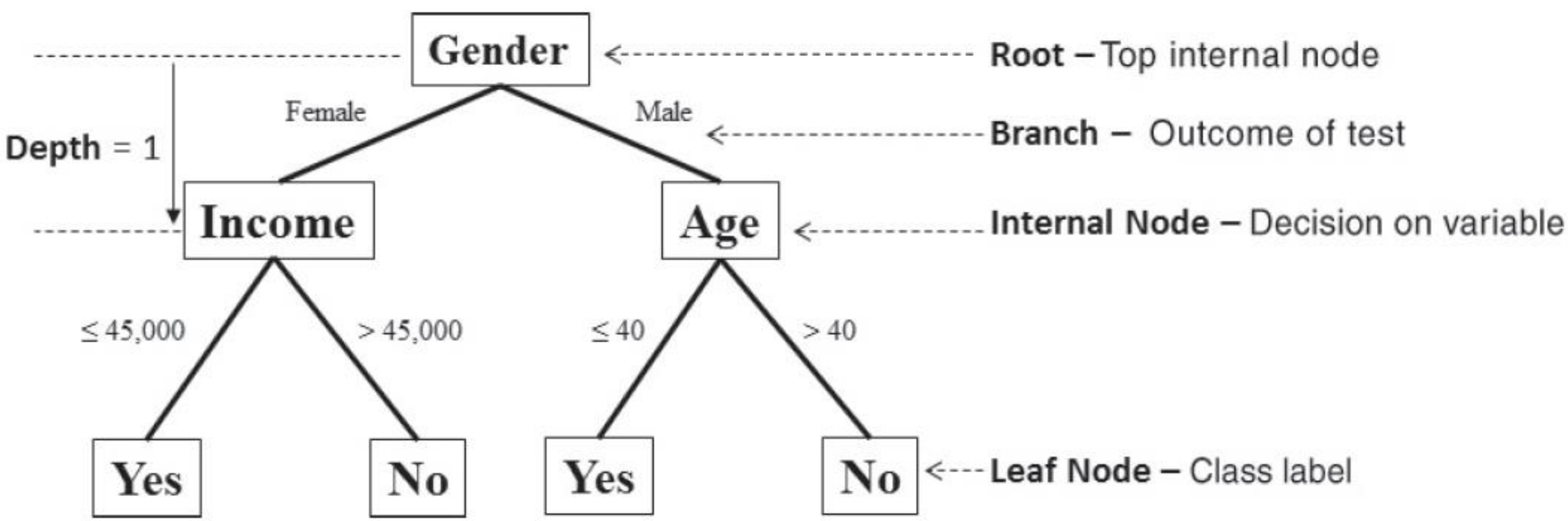
Decision Tree

- Each node tests a particular input variable
- Each branch represents the decision made
- Classifying a new observation is to traverse this decision tree.



Decision Tree

- The depth of a node is the minimum number of steps required to reach the node from root
- Leaf nodes are at the end of the last branches on the tree, representing class labels



Decision Tree

- Use cases
  - Classify animals: questions (like cold-blooded or warm-blooded, mammal or not mammal) are answered to arrive at a certain classification
  - Checklist of symptoms during a doctor's evaluation of a patient
  - Retailers use decision tree to predict response rates to marketing and promotions
  - Financial institutions use it for loan application

Decision Tree

- An example: A bank markets its term deposit product. So the bank needs to predict which clients would subscribe to a term deposit
  - The bank collects a dataset of 2000 previous clients with known "subscribe or not".
  - Input variables to describe each client are
    - Job, marital status, education level, credit default, housing loan, personal loan, contact type, previous campaign contact

Decision Tree

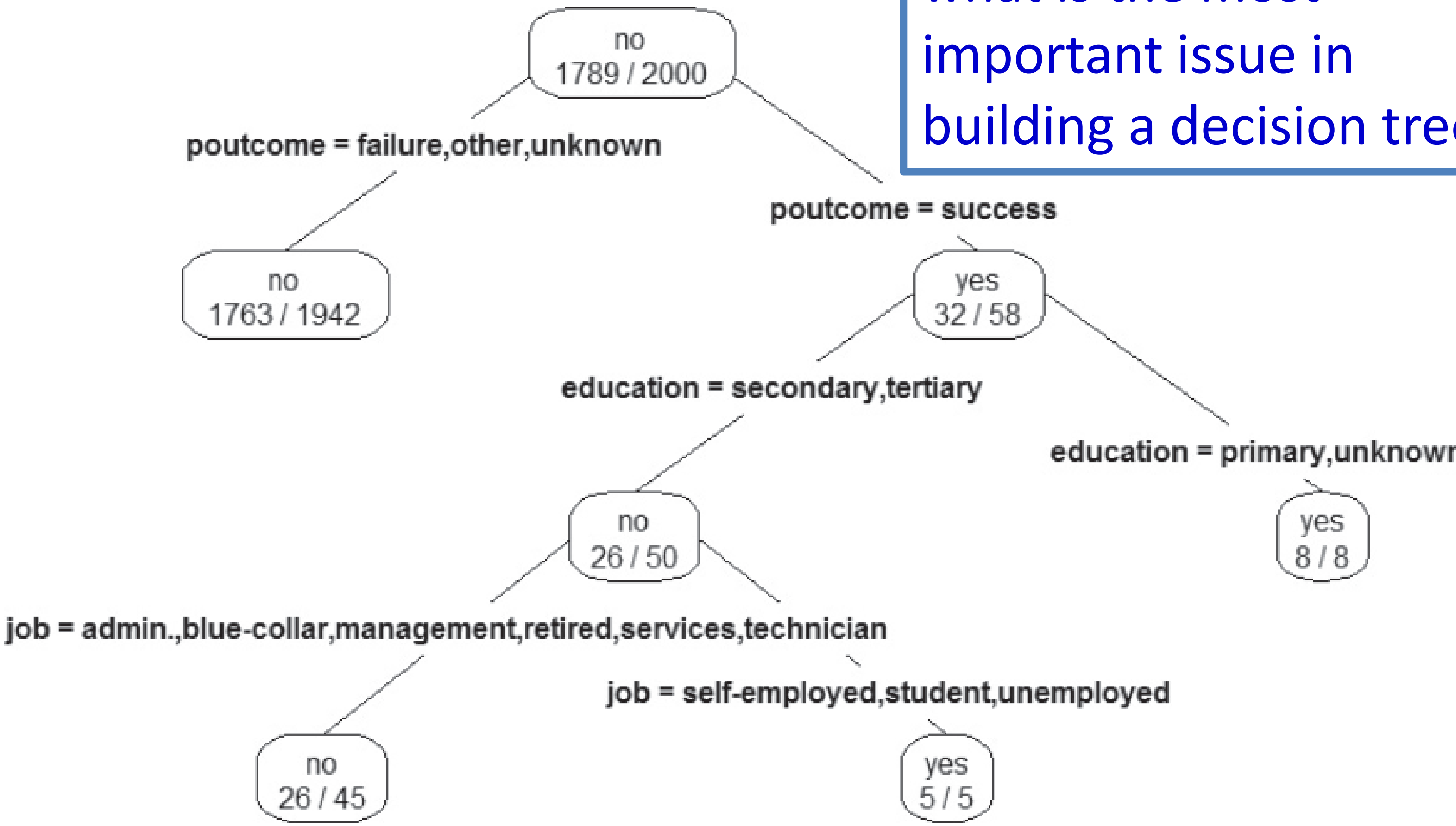
	job	marital	education	default	housing	loan	contact	poutcome	subscribed
1	management	single	tertiary	no	yes	no	cellular	unknown	no
2	entrepreneur	married	tertiary	no	yes	yes	cellular	unknown	no
3	services	divorced	secondary	no	no	no	cellular	unknown	yes
4	management	married	tertiary	no	yes	no	cellular	unknown	no
5	management	married	secondary	no	yes	no	unknown	unknown	no
6	management	single	tertiary	no	yes	no	unknown	unknown	no
7	entrepreneur	married	tertiary	no	yes	no	cellular	failure	yes
8	admin.	married	secondary	no	no	no	cellular	unknown	no
9	blue-collar	married	secondary	no	yes	no	cellular	other	no
10	management	married	tertiary	yes	no	no	cellular	unknown	no
11	blue-collar	married	secondary	no	yes	no	cellular	unknown	no
12	management	divorced	secondary	no	no	no	unknown	unknown	no
13	blue-collar	married	secondary	no	yes	no	cellular	unknown	no
14	retired	married	secondary	no	no	no	cellular	unknown	no
15	management	single	tertiary	no	yes	no	cellular	unknown	no

…

The training dataset of the bank example

Decision Tree

From your point of view, what is the most important issue in building a decision tree?



A decision tree built over the bank marketing training dataset

The General Algorithm of DT

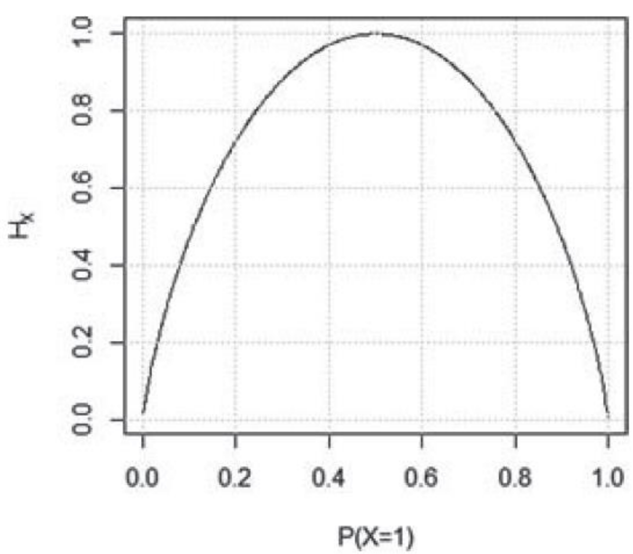
- The objective of a decision tree algorithm
  - Construct a tree T from a training set S
- The algorithm picks the most informative attribute to branch the tree and does this recursively for each of the sub-trees.
- The most informative attribute is identified by
  - Information gain, calculated based on Entropy

The General Algorithm of DT

- Entropy

Given a class X and its label $x \in X$, let $P(x)$ be the probability of x. H_{x}, the entropy of X, is defined as

$$H_X = -\sum_{\forall x \in X} P(x) \log_2 P(x)$$



Entropy of coin flips, where X=1 represents heads

Question:
In the previous bank marketing dataset, there are 2000 customers in total. Among them, 1789 subscribed term deposit. What is the entropy of the output variable "subscribed" ($H_{subscribed}$)?

The General Algorithm of DT

- Conditional entropy

Given an attribute X, its value x, its outcome Y, and its value y, conditional entropy $H_{Y|X}$ is the remaining entropy of Y given X,

$$H_{Y|X} = \sum_{x} P(x) H(Y|X = x)$$

$$= -\sum_{\forall x \in X} P(x) \sum_{\forall y \in Y} P(y|x) \log_2 P(y|x)$$

The General Algorithm of DT

- Assume the attribute X is "contact"
  - Its value x takes one value in {cellular, telephone, unknown}
- The outcome Y is "subscribed"
  - Its value y takes one value in {no, yes}

	Cellular	Telephone	Unknown
P(contact)	0.6435	0.0680	0.2885
P(subscribed=yes | contact)	0.1399	0.0809	0.0347
P(subscribed=no | contact)	0.8601	0.9192	0.9653

The General Algorithm of DT

The conditional entropy of the `contact` attribute is computed as shown here.

$$H_{Y|X} = \sum_{x} P(x) H(Y|X = x)$$

$$= -\sum_{\forall x \in x} P(x) \sum_{\forall y \in Y} P(y|x) \log_2 P(y|x)$$

$$\begin{aligned} H_{subscribed|contact} &= -\big[0.6435 \cdot (0.1399 \cdot \log_2 0.1399 + 0.8601 \cdot \log_2 0.8601) \\ &+ 0.0680 \cdot (0.0809 \cdot \log_2 0.0809 + 0.9192 \cdot \log_2 0.9192) \\ &+ 0.2885 \cdot (0.0347 \cdot \log_2 0.0347 + 0.9653 \cdot \log_2 0.9653)\big] \\ &= 0.4661 \end{aligned}$$

	Cellular	Telephone	Unknown
P(contact)	0.6435	0.0680	0.2885
P(subscribed=yes | contact)	0.1399	0.0809	0.0347
P(subscribed=no | contact)	0.8601	0.9192	0.9653

The General Algorithm of DT

- Information gain

The information gain of an attribute A is defined as the difference between the base entropy and the conditional entropy of the attribute,

$$InfoGain_A = H_S - H_{S|A}$$

$$InfoGain_{contact} = H_{subscribed} - H_{subscribed|contact}$$
$$= 0.4862 - 0.4661 = 0.0201$$

- It compares
  - The degree of purity of the parent node before a split
  - The degree of purity of the child node after a split

The General Algorithm of DT

- The algorithm splits on the attribute with the largest information gain at each round

Attribute	Information Gain
poutcome	0.0289
contact	0.0201
housing	0.0133
job	0.0101
education	0.0034
marital	0.0018
loan	0.0010
default	0.0005

The General Algorithm of DT

- The algorithm constructs sub-trees recursively until one of the following criteria is met
  - All the leaf nodes in the tree satisfy the minimum purity threshold (i.e., are pure enough)
  - There is no sufficient information gain by splitting on more attribute (i.e., not worth anymore)
  - Any other stopping criterion is satisfied (such as the maximum depth of the tree)

Decision Tree Algorithms

- Popular decision tree algorithms
  - ID3, C4.5 and CART

ID3 (A, P, T)
  if $T \in \phi$
    return ϕ
  if all records in T have the same value for P
    return a single node with that value
  if $A \in \phi$
    return a single node with the most frequent value of P in T
  Compute information gain for each attribute in A relative to T
  Pick attribute D with the largest gain
  Let $\{ d_1, d_2 \ldots d_m \}$ be the values of attribute D
  Partition T into $\{ T_1, T_2 \ldots T_m \}$ according to the values of D
  return a tree with root D and branches labeled $d_1, d_2 \ldots d_m$
        going respectively to trees ID3(A-{D}, P, T_1),
        ID3(A-{D}, P, T_2), . . . ID3(A-{D}, P, T_m)

Evaluating a Decision Tree

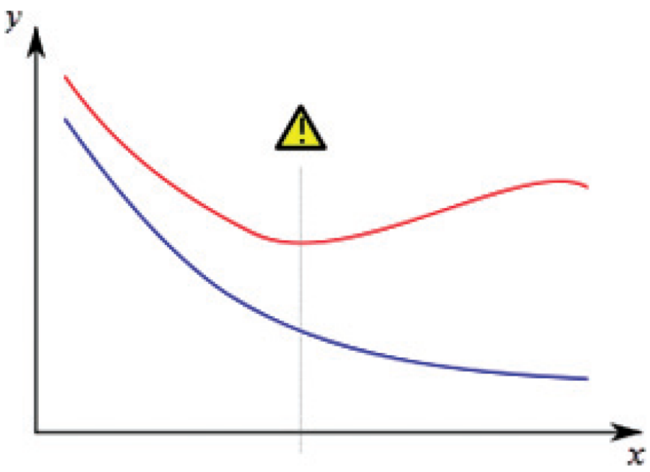
- Decision tree uses greedy algorithms
  - It always chooses the option that seems the best available at that moment
  - However, the option may not be the best overall and this could cause overfitting
  - An ensemble technique can address this issue by combining multiple decision trees that use random splitting

Evaluating a Decision Tree

- Ways to evaluate a decision tree
  - Evaluate whether the splits of the tree make sense and whether the decision rules are sound (say, with domain experts)
  - Having too many layers and obtaining nodes with few members might be signs of overfitting
  - Use standard diagnostics tools for classifiers

Evaluating a Decision Tree

- Overfitting in decision tree
  - The lack of training data
  - The biased training data
  - Too many layers or nodes



- Avoid overfitting
  - Stop growing the tree early before all training data are perfectly classified
  - Grow the full tree and then post-prune the tree

Properties of Decision Tree

- Computationally inexpensive, easy to classify
- Classification rules can be understood
- Handle both numerical and categorical input
- Handle variables that have a nonlinear effect on the outcome, better than linear models
- Not a good choice if there are many irrelevant input variables
  - Feature selection will be needed

Decision Tree in R

- `rpart` library is for modelling decision tree
- `rpart.plot` enables the plotting of a tree
- An example
  - Predict whether to play golf
  - Input variables: weather outlook, temperature, humidity, and wind

```
install.packages("rpart.plot")   # install package rpart.plot
library("rpart")   # load libraries
library("rpart.plot")
```

Decision Tree in R

```
play_decision <- read.table("DTdata.csv",header=TRUE,sep=",")
play_decision
   Play  Outlook Temperature Humidity  Wind
1   yes    rainy        cool   normal FALSE
2    no    rainy        cool   normal  TRUE
3   yes overcast         hot     high FALSE
4    no    sunny        mild     high FALSE
5   yes    rainy        cool   normal FALSE
6   yes    sunny        cool   normal FALSE
7   yes    rainy        cool   normal FALSE
8   yes    sunny         hot   normal FALSE
9   yes overcast        mild     high  TRUE
10   no    sunny        mild     high  TRUE
```

```
fit <- rpart(Play ~ Outlook + Temperature + Humidity + Wind,
             method="class",
             data=play_decision,
             control=rpart.control(minsplit=1),
             parms=list(split='information'))
```

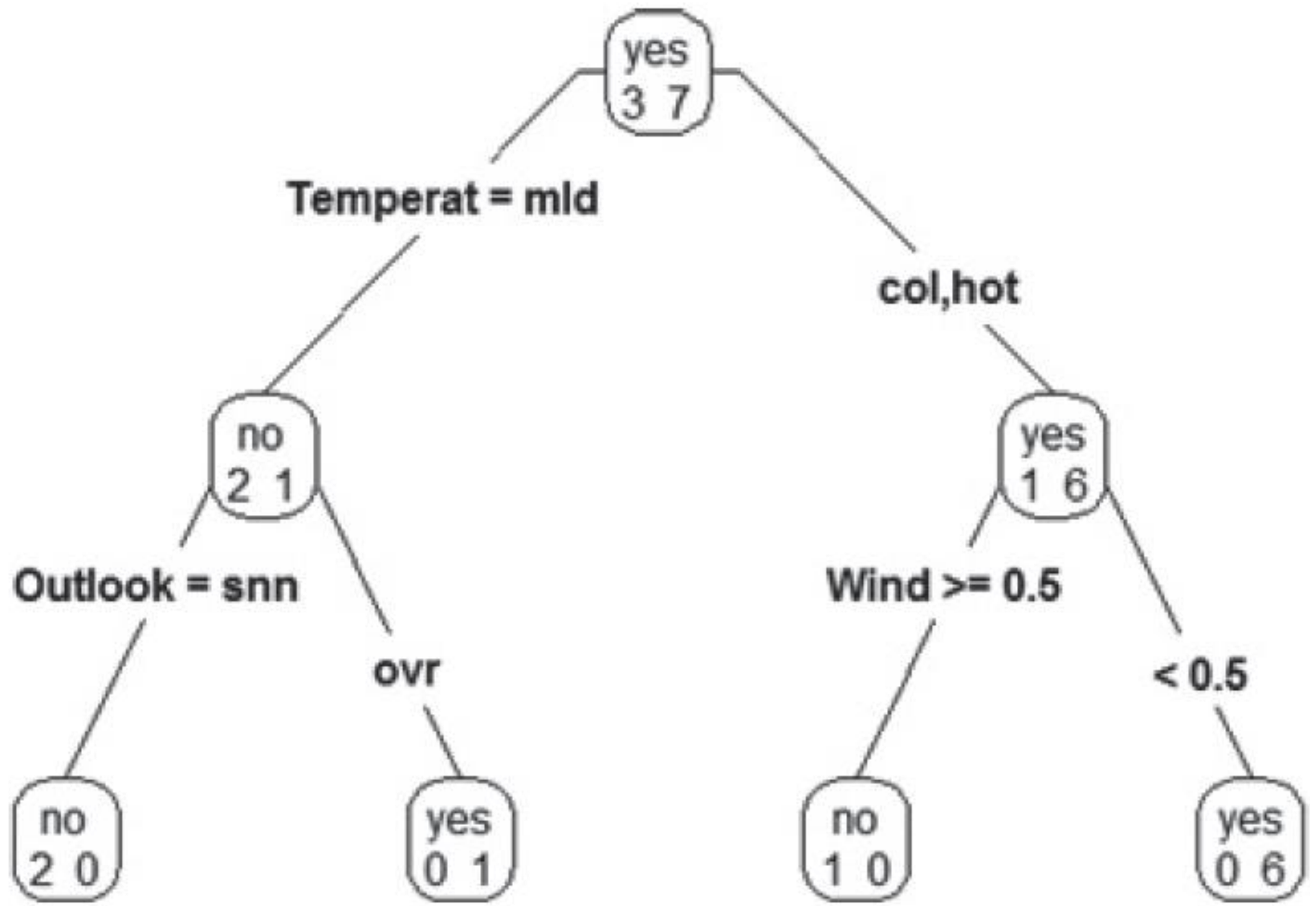
Decision Tree in R

```
summary(fit)
Call:
rpart(formula = Play ~ Outlook + Temperature + Humidity + Wind,
  data = play_decision, method = "class",
  parms = list(split = "information"),
      control = rpart.control(minsplit = 1))
  n= 10
         CP nsplit rel error   xerror      xstd
1 0.3333333      0         1 1.000000 0.4830459
2 0.0100000      3         0 1.666667 0.5270463

Variable importance
       Wind     Outlook Temperature
         51          29          20

Node number 1: 10 observations,    complexity param=0.3333333
  predicted class=yes  expected loss=0.3  P(node) =1
    class counts:     3     7
   probabilities: 0.300 0.700
  left son=2 (3 obs) right son=3 (7 obs)
  Primary splits:
      Temperature splits as  RRL,     improve=1.3282860, (0 missing)
      Wind        < 0.5 to the right, improve=1.3282860, (0 missing)
      Outlook     splits as  RLL,     improve=0.8161371, (0 missing)
      Humidity    splits as  LR,      improve=0.6326870, (0 missing)
  Surrogate splits:
      Wind < 0.5 to the right, agree=0.8, adj=0.333, (0 split)

Node number 2: 3 observations,    complexity param=0.3333333
  predicted class=no   expected loss=0.3333333  P(node) =0.3
    class counts:     2     1
   probabilities: 0.667 0.333
```

```
  left son=4 (2 obs) right son=5 (1 obs)
  Primary splits:
      Outlook splits as  R-L,     improve=1.9095430, (0 missing)
      Wind    < 0.5 to the left,  improve=0.5232481, (0 missing)

Node number 3: 7 observations,    complexity param=0.3333333
  predicted class=yes  expected loss=0.1428571  P(node) =0.7
    class counts:     1     6
   probabilities: 0.143 0.857
  left son=6 (1 obs) right son=7 (6 obs)
  Primary splits:
      Wind        < 0.5 to the right, improve=2.8708140, (0 missing)
      Outlook     splits as  RLR,     improve=0.6214736, (0 missing)
      Temperature splits as  LR-,     improve=0.3688021, (0 missing)
      Humidity    splits as  RL,      improve=0.1674470, (0 missing)

Node number 4: 2 observations
  predicted class=no   expected loss=0  P(node) =0.2
    class counts:     2     0
   probabilities: 1.000 0.000

Node number 5: 1 observations
  predicted class=yes  expected loss=0  P(node) =0.1
    class counts:     0     1
   probabilities: 0.000 1.000

Node number 6: 1 observations
  predicted class=no   expected loss=0  P(node) =0.1
    class counts:     1     0
   probabilities: 1.000 0.000

Node number 7: 6 observations
  predicted class=yes  expected loss=0  P(node) =0.6
    class counts:     0     6
   probabilities: 0.000 1.000
```

Decision Tree in R

```
rpart.plot(fit, type=4, extra=1)
```



Decision Tree in R

- Prediction outcome for a new observation

```
newdata <- data.frame(Outlook="rainy", Temperature="mild",
                      Humidity="high", Wind=FALSE)
```

```
predict(object, newdata = list(),
        type = c("vector", "prob", "class", "matrix"))
```

```
predict(fit,newdata=newdata,type="prob")

  no yes
1  1   0
```

```
predict(fit,newdata=newdata,type="class")

 1
 no
Levels: no yes
```

Summary on Decision Tree

From http://www.projectdecisions.org/index-cartoon-decisiontree.html

Naïve Bayes Classifier

- A probabilistic classification method based on Bayes' theorem
- A naïve Bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of other features (conditional independence assumption)
- Output includes a class label and its corresponding probability score

Bayes' Theorem

C is the class label $C \in \{c_1, c_2, \ldots c_n\}$

A is the observed attributes $A = \{a_1, a_2, \ldots a_m\}$

$$P(C|A) = \frac{P(A|C) \cdot P(C)}{P(A)}$$

$$\text{Posteriori probability} = \frac{\text{likelihood} \cdot \text{priori probability}}{\text{evidence}}$$

Bayes’ Theorem

- Two examples to understand this theorem

An example better illustrates the use of Bayes’ theorem. John flies frequently and likes to upgrade his seat to first class. He has determined that if he checks in for his flight at least two hours early, the probability that he will get an upgrade is 0.75; otherwise, the probability that he will get an upgrade is 0.35. With his busy schedule, he checks in at least two hours before his flight only 40% of the time. Suppose John did not receive an upgrade on his most recent attempt. What is the probability that he did not arrive two hours early?

Another example involves computing the probability that a patient carries a disease based on the result of a lab test. Assume that a patient named Mary took a lab test for a certain disease and the result came back positive. The test returns a positive result in 95% of the cases in which the disease is actually present, and it returns a positive result in 6% of the cases in which the disease is not present. Furthermore, 1% of the entire population has this disease. What is the probability that Mary actually has the disease, given that the test is positive?

Bayes' Theorem

- A more practical form of Bayes' theorem

$$P(c_i|A) = \frac{P(a_1, a_2, \ldots, a_m | c_i) \cdot P(c_i)}{P(a_1, a_2, \ldots, a_m)}, i = 1, 2, \ldots n$$

C is the class label $C \in \{c_1, c_2, \ldots c_n\}$

A is the observed attributes $A = \{a_1, a_2, \ldots a_m\}$

- Given A, how to calculate $P(c_i|A)$?

Naïve Bayes Classifier

- With two simplifications, Bayes' theorem induces a Naïve Bayes classifier
- First, Conditional independence assumption
  - Each attribute is conditionally independent of every other attribute given a class label c_i

$$P(a_1, a_2, \ldots, a_m | c_i) = P(a_1|c_i) P(a_2|c_i) \cdots P(a_m|c_i) = \prod_{j=1}^{m} P(a_j|c_i)$$

  - This simplifies the computation of $P(A|c_i)$

Naïve Bayes Classifier

- Second, ignore the denominator *P(A)*
  - Removing the denominator has no impact on the relative probability scores
- In this way, the classifier becomes

$$P(c_i|A) \propto P(c_i) \cdot \prod_{j=1}^{m} P(a_j|c_i) \qquad i = 1,2,\ldots n$$

$$P(c_i|A) \propto \log P(c_i) + \sum_{j=1}^{m} \log P(a_j|c_i) \qquad i = 1,2,\ldots n$$

Naïve Bayes Classifier

- An example
  - With the bank marketing dataset, use Naïve Bayes Classifier to predict if a client would subscribe to a term deposit
- Building a Naïve Bayes classifier requires to calculate some statistics from training dataset
  - $P(A|c_i)$ for each class i=1,2,…,n
  - $P(a_j|c_i)$ for each attribute j=1,2,…,m in each class

$$P(c_i|A) \propto P(c_i) \cdot \prod_{j=1}^{m} P(a_j|c_i) \qquad i = 1,2,\ldots n$$

Naïve Bayes Classifier

- $P(A|c_i)$ for each class

$$P(\textit{subscribed} = \text{yes}) \approx 0.11 \text{ and } P(\textit{subscribed} = \text{no}) \approx 0.89$$

The training set contains several attributes: `job`, `marital`, `education`, `default`, `housing`, `loan`, `contact`, and `poutcome`.

- $P(a_j|c_i)$ for each attribute in each class

$$P(single \mid subscribed = yes) \approx 0.35$$

$$P(married \mid subscribed = yes) \approx 0.53$$

$$P(divorced \mid subscribed = yes) \approx 0.12$$

$$P(single \mid subscribed = no) \approx 0.28$$

$$P(married \mid subscribed = no) \approx 0.61$$

$$P(divorced \mid subscribed = no) \approx 0.11$$

Naïve Bayes Classifier

- Testing a Naïve Bayes classifier on a new data

j	a_j	$P(a_j \mid \text{subscribed} = \text{yes})$	$P(a_j \mid \text{subscribed} = \text{no})$
1	job = management	0.22	0.21
2	marital = married	0.53	0.61
3	education = secondary	0.46	0.51
4	default = no	0.99	0.98
5	housing = yes	0.35	0.57
6	loan = no	0.90	0.85
7	contact = cellular	0.85	0.62
8	poutcome = success	0.15	0.01

$$P(yes|A) \propto 0.11 \cdot (0.22 \cdot 0.53 \cdot 0.46 \cdot 0.99 \cdot 0.35 \cdot 0.90 \cdot 0.85 \cdot 0.15) \approx 0.00023$$

$$P(no|A) \propto 0.89 \cdot (0.21 \cdot 0.61 \cdot 0.51 \cdot 0.98 \cdot 0.57 \cdot 0.85 \cdot 0.62 \cdot 0.01) \approx 0.00017$$

Naïve Bayes Classifier

- An issue on rare event
  - What if one of the attribute values does NOT appear in a class c_i in a training dataset?
  - $P(a_j|c_i)$ for this attribute value will equal zero!
  - $P(c_i|A)$ will simply become zero!
- Smoothing technique
  - It assigns a small nonzero probability to rare events not included in a training dataset

Naïve Bayes Classifier

- Laplace smoothing (add-one smoothing)
  – It pretends to see every outcome once more than it actually appears

$$P^{*}(x) = \frac{count(x)+1}{\Sigma_x[count(x)+1]}$$

$$P'(single \mid subscribed = yes) = (20+1)/[(20+1)+(70+1)+(10+1)]$$

$$P^{**}(x) = \frac{count(x)+\varepsilon}{\Sigma_x[count(x)+\varepsilon]} \qquad \varepsilon \in [0,1]$$

Naïve Bayes Classifier

- Advantages
  - Simple to implement, commonly used for text classification
  - Handle high-dimensional data efficiently
  - Robust to overfitting with smoothing technique
- Disadvantages
  - Sensitive to correlated variables (Why?)
  - Not reliable for probability estimation

Naïve Bayes in R

- Two methods
  - Build the classifier from the scratch
  - Call `naiveBayes` function from `e1071` package

```
install.packages("e1071")   # install package e1071
library(e1071)   # load the library

# read the data into a table from the file
sample <- read.table("sample1.csv",header=TRUE,sep=",")
# define the data frames for the NB classifier
traindata <- as.data.frame(sample[1:14,])
testdata <- as.data.frame(sample[15,])
```

Naïve Bayes in R

```
model <- naiveBayes(Enrolls ~ Age+Income+JobSatisfaction+Desire,
                    traindata)
# display model
model

# predict with testdata
results <- predict (model,testdata)
# display results
results
[1] Yes
Levels:  No Yes
```

```
# use the NB classifier with Laplace smoothing
model1 = naiveBayes(Enrolls ~., traindata, laplace=.01)
```

Diagnostics of Classifiers

- Confusion matrix

		Predicted Class	
		Positive	Negative
Actual Class	Positive	True Positives (TP)	False Negatives (FN)
	Negative	False Positives (FP)	True Negatives (TN)

		Predicted Class		
		Subscribe	Not Subscribed	Total
Actual Class	Subscribed	3	8	11
	Not Subscribed	2	87	89
Total		5	95	100

Diagnostics of Classifiers

- Metrics used to evaluate classifiers

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$FPR = \frac{FP}{FP + TN} \qquad TPR = \frac{TP}{TP + FN} \qquad FNR = \frac{FN}{TP + FN}$$

$$Precision = \frac{TP}{TP + FP} \qquad TPR\ (or\ Recall) = \frac{TP}{TP + FN}$$

Additional Classification Models

- Bagging
  - Bootstrap technique, ensemble method
- Boosting
  - Weighted combination, ensemble method
- Random Forest
  - Combination of decision trees, ensemble method
- Support Vector Machines
  - Max-margin linear classifier, kernel trick

Summary

- Decision trees and Naïve Bayes classifier

Concerns	Recommended Method(s)
Output of the classification should include class probabilities in addition to the class labels.	Logistic regression, decision tree
Analysts want to gain an insight into how the variables affect the model.	Logistic regression, decision tree
The problem is high dimensional.	Naïve Bayes
Some of the input variables might be correlated.	Logistic regression, decision tree
Some of the input variables might be irrelevant.	Decision tree, naïve Bayes
The data contains categorical variables with a large number of levels.	Decision tree, naïve Bayes
The data contains mixed variable types.	Logistic regression, decision tree
There is nonlinear data or discontinuities in the input variables that would affect the output.	Decision tree

Q&A

Images Courtesy of Google Image