Lecture 05
Data Exploring & Classification Tasks

Oran Kittithreerapronchai[1]

[1]Department of Industrial Engineering, Chulalongkorn University
Bangkok 10330 THAILAND

last updated: August 7, 2023

Outline

source: General references [NC20, TSK16, BG19, Pat14]

What is Data Mining

We are drowning in data, but starving for knowledge!

source: Rutherford D. Rogers, —a librarian at Yale University

- **Data mining:** process of extraction of interesting patterns from huge data
- **Alternative names:** Knowledge discovery in databases (KDD)
- **Application:**
  - **Marketing analysis:** target marketing, associations of sales and features, customer profiling, attraction factors for new customers
  - **Fraud detection:** ring of mal-practice doctors, unnecessary screening, dishonest employees' pattern

Pattern Discovery

There can be thousands of patterns, yet none of them matters!

source: Anonymous

- **Pattern Discovery Approach:**
  - **Suggested approach:** Human-centered, query-based, focused mining
  - **Feature selection approach:** filtering method VS wrapping method
- **Interestingness Measuring**
  - **Objective:** defined by statistics and structures of patterns, e.g., support, confidence
  - **Subjective:** defined by user belief in the data, e.g., unexpectedness, novelty.

Data Mining Grouping

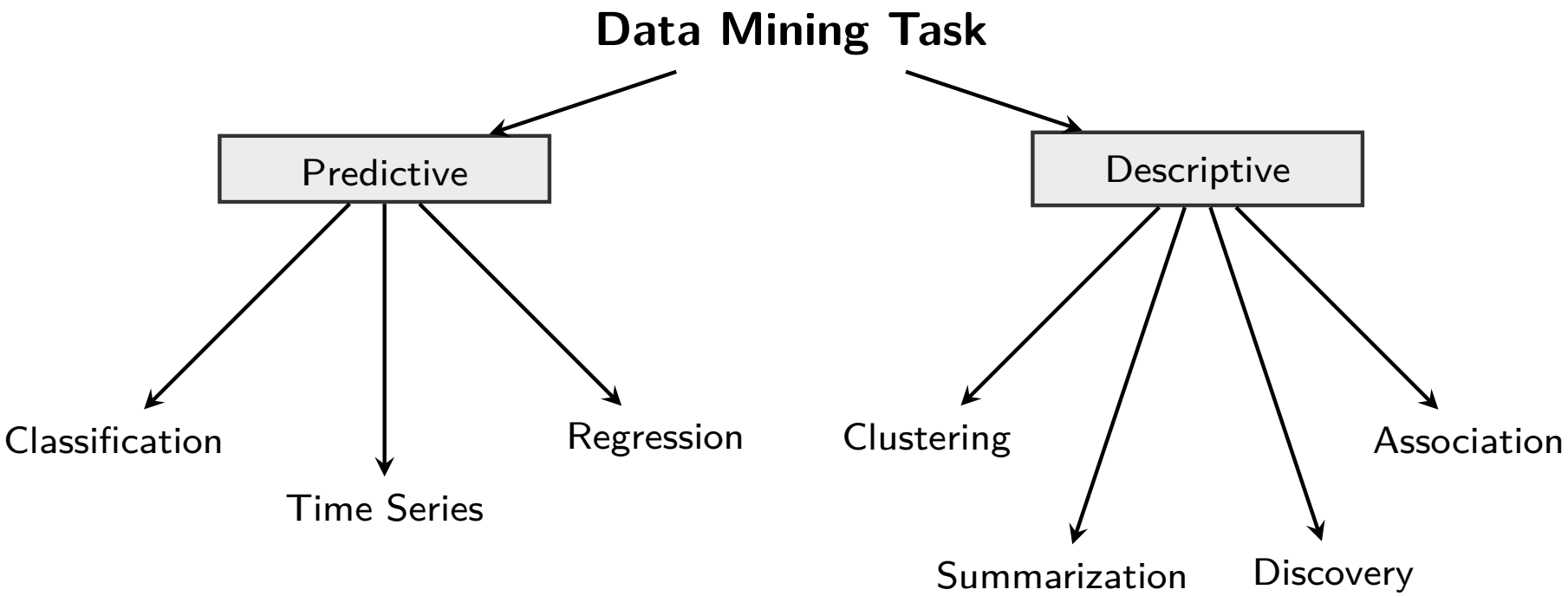
- **Data Mined:** data warehouse, transactional, stream, spatial, time-series, multi-media, heterogeneous
- **Task:** characterization, association, classification, clustering, anomaly
- **Techniques Utilized:** machine learning, statistics, visualization
- **Application Adapted:** retail, telecommunication, banking, fraud analysis

Knowledge Discovery

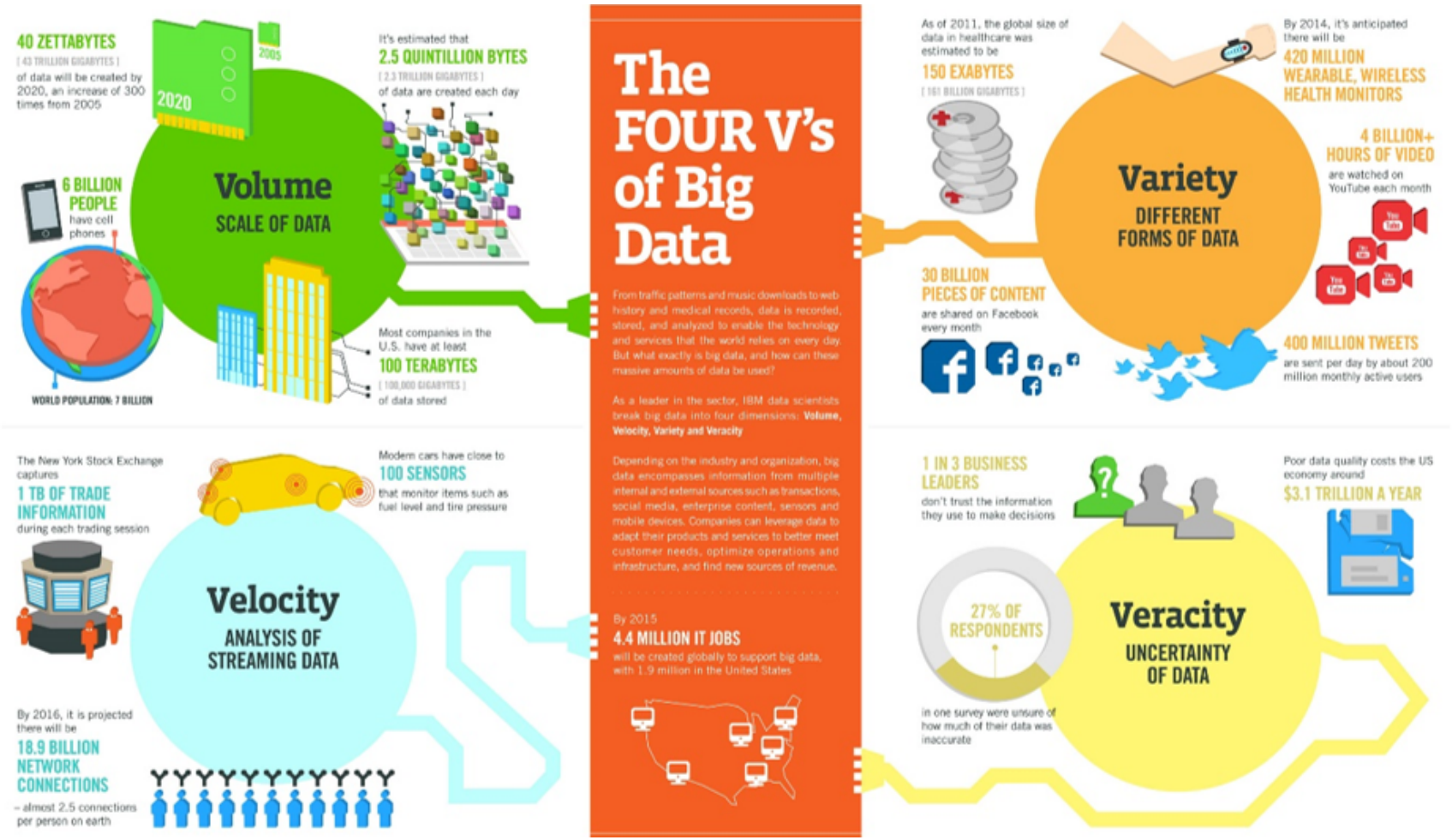
- feature selection / query
- dimension reduction
- normalization
- data subsetting

- filtering patterns
- visualization
- pattern interpretation
- API

Data Mining Task



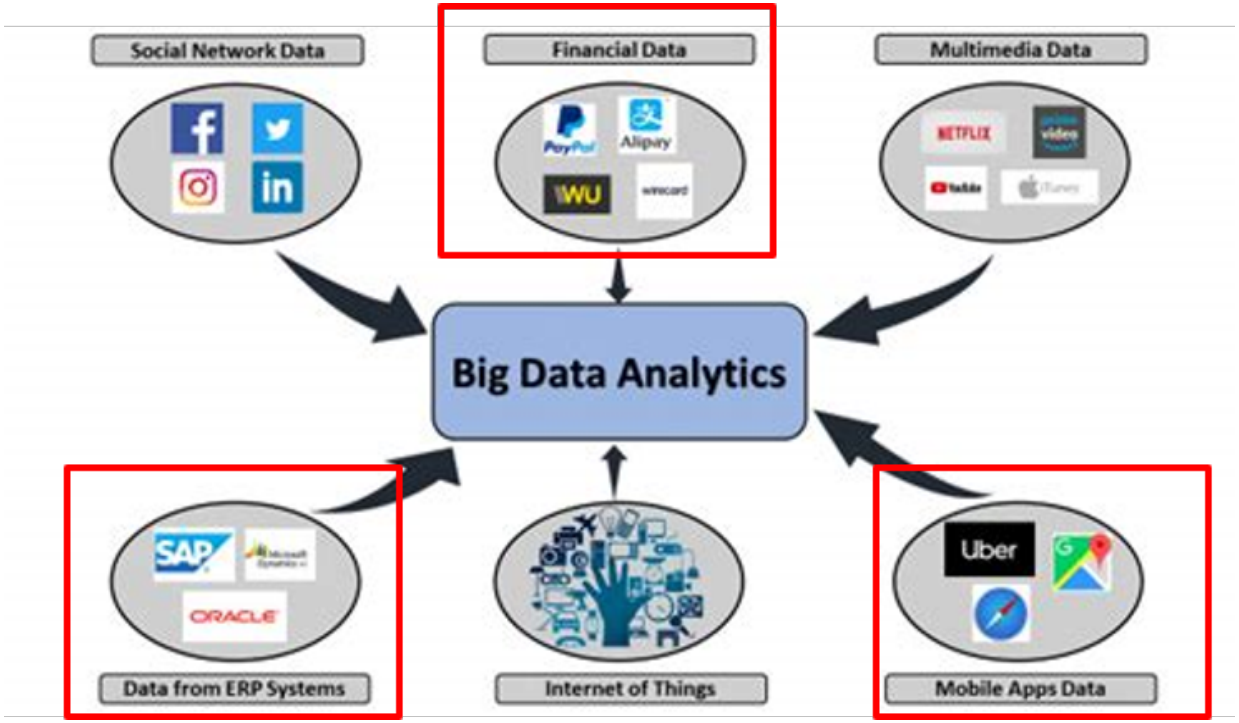
4V of Data Science



Data 101

- **What is Data?:** Collection of objects and attribute
- **Object is also called:** record, point, case, sample, entity, or instance
- **Attribute is:** property of an object, e.g. vehicle (engine, steering, compartment, fuel tank, lighting)
- **Classification of attribute:**
  - *Nominal/Catagory:* numbers, eye color, zip codes
  - *Ordinal:* rankings, grades, satisfaction survey
  - *Interval:* calendar dates
  - *Ratio:* sugar concentration
- **Type of record:** data frame, data matrix, data transaction, data attribute
- **Problem with data:** noise, duplicate, outliers, missing values

Data Collection



Main Classification of Data

- **sources:** platform, internal VS external
- **structure:** e.g., SQL, annual report, twitter, CCTV

Data Frame & Data Matrix

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

Data Frame

Projection of x Load	Projection of y load	Distance	Load	Thickness
10.23	5.27	15.22	2.7	1.2
12.65	6.25	16.22	2.2	1.1

Data Matrix

```
data.frame(iris)
as.data.table(iris)
```

```
matrix(iris[,1:4])
array(1:8,dim=c(2,2,2))
```

Data Transaction & Data Attribute

CID	TID
1	bread, soda, milk
2	beer, bread
3	bread, soda, diaper, milk
4	beer, bread, diaper, milk
5	soda, diaper, milk
6	beer, soda

Data Transaction

```
list(
 cid1=c("bread","soda","milk")
 cid2=c("beer","bread")
 cid3=c("bread","soda","diaper","milk")
)
```

	team	coach	play	ball	score	game	win	lost	timeout	season
Document 1	3	0	5	0	2	6	0	2	0	2
Document 2	0	7	0	2	1	0	0	3	0	0
Document 3	0	1	0	0	1	2	2	0	3	0

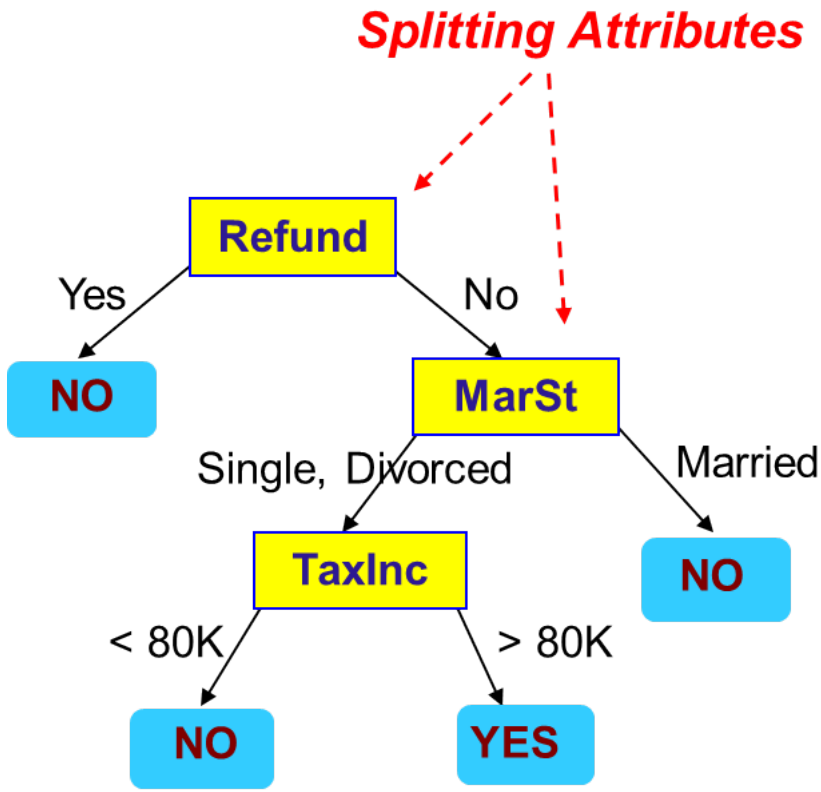
Data attribute

```
require(MASS)
table(Cars93$Manufacturer,Cars93$Type)
```

Decision Tree

	categorical	categorical	continuous	class
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

$\Rightarrow$



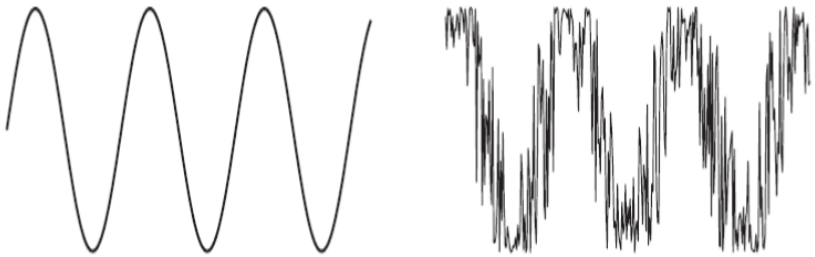
Data Arrangement

	1	2	3	4	5	6
1	0	1	0	1	1	0
2	1	0	1	0	0	1
3	0	1	0	1	1	0
4	1	0	1	0	0	1
5	0	1	0	1	1	0
6	1	0	1	0	0	1
7	0	1	0	1	1	0
8	1	0	1	0	0	1
9	0	1	0	1	1	0

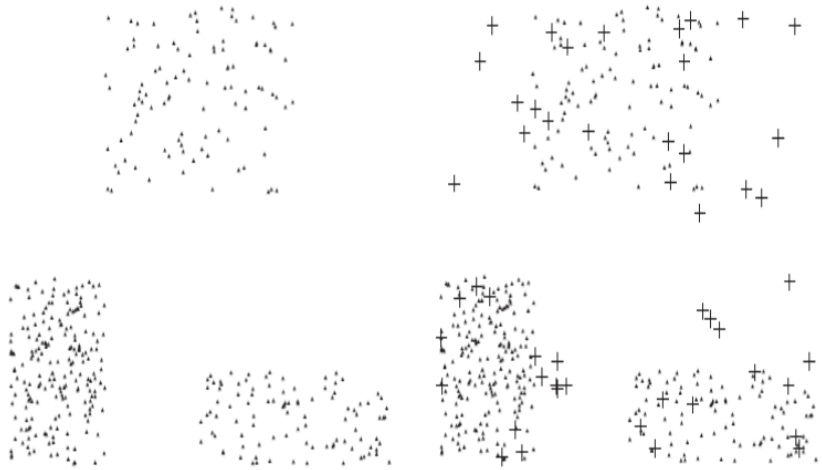
	6	1	3	2	5	4
4	1	1	1	0	0	0
2	1	1	1	0	0	0
6	1	1	1	0	0	0
8	1	1	1	0	0	0
5	0	0	0	1	1	1
3	0	0	0	1	1	1
9	0	0	0	1	1	1
1	0	0	0	1	1	1
7	0	0	0	1	1	1

```
set.seed(186)
smpMat <- matrix(rep(0,9*6),ncol=6)
smpMat[sample(1:(9*6),size = 16)] <- 1
colnames(smpMat) <- paste("col",1:6,sep="")
rownames(smpMat) <- paste("row",1:9,sep="")

colOrd <- order(colSums(smpMat),decreasing = T)
rowOrd <- order(rowSums(smpMat),decreasing = T)
smpMat[rowOrd,colOrd]
```

Noises



Time Series

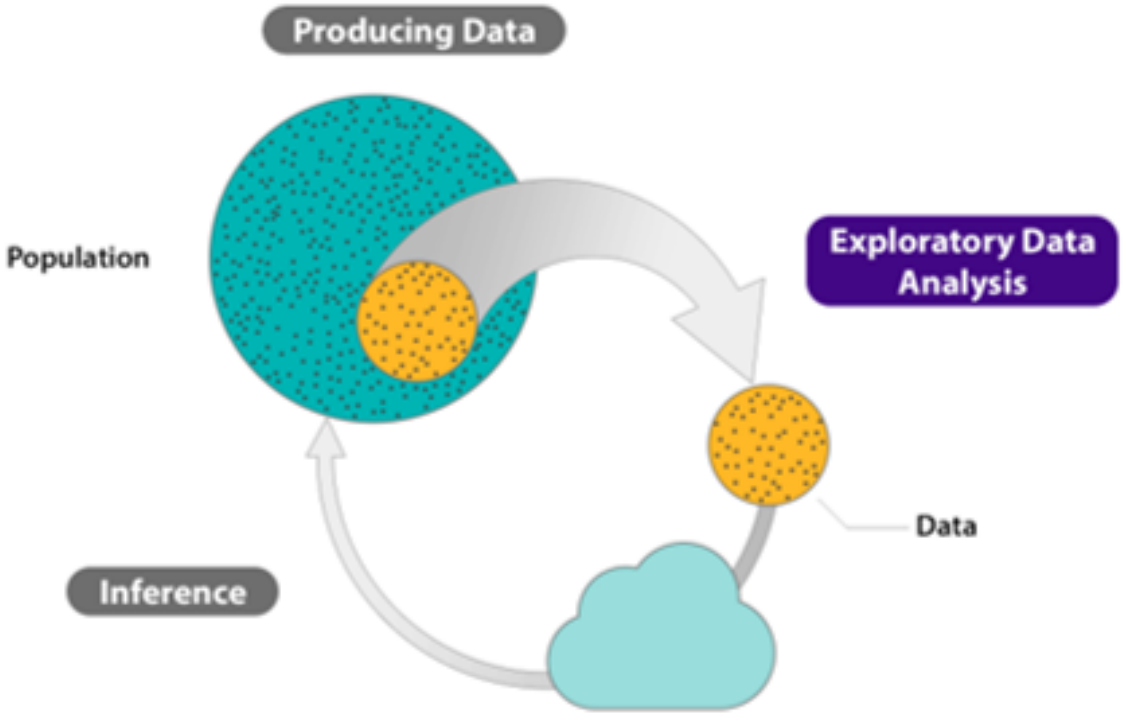


Spatial Context

Noise Properties

- zero-mean normal distributed, i.e., $\mathcal{N}(0, \sigma_n) \rightarrow$ MA
- insignificance, i.e., $\sigma_n \ll \sigma_{pro}$

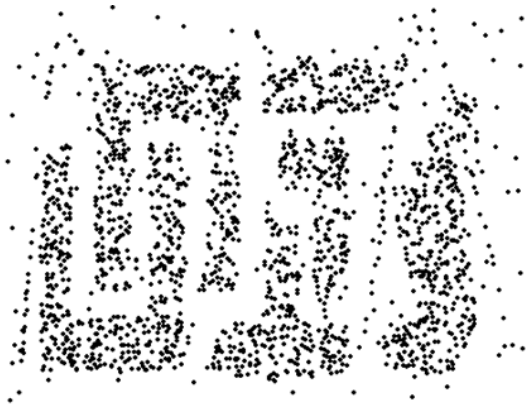
Population VS Sample



Sampling Method

- **Probabilistic** e.g., random, stratifier, cluster
- **Non-Probabilistic** e.g., quota, purpose, volunteer, haphazard

Sample Size Matter

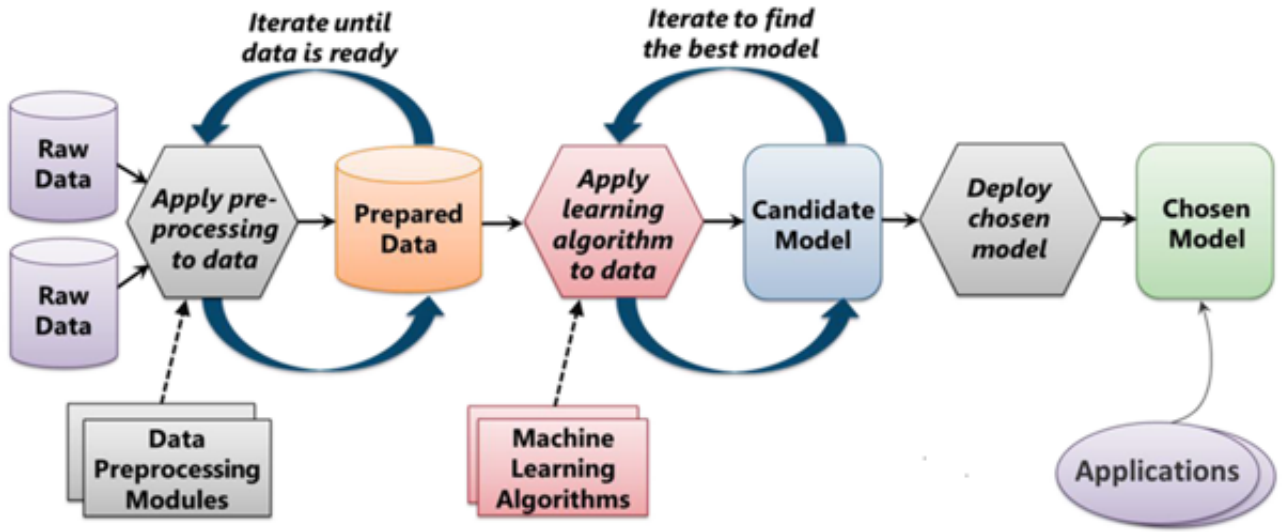




Types of Important Sampling

- **Purposive Sampling:** query some feature, e.g. extreme value, typical VS critical case
- **Random Sampling:** equality sampling w/ or w/o cluster
- **Stratified Sampling:** selected features sampling (definitely, bias)

Key Message: Think! What do you want to get from sampling?

Basis Approach to Data Analysis



Step 0 **Clarify objective:** analysis questions → data requirement

Step 1 **Gather data:** source & relationship → digitalize data

Step 2 **Clean data:** outlier, missing, duplicate

Step 3 **Explore data:** descriptive & visualize → crosscheck with operators

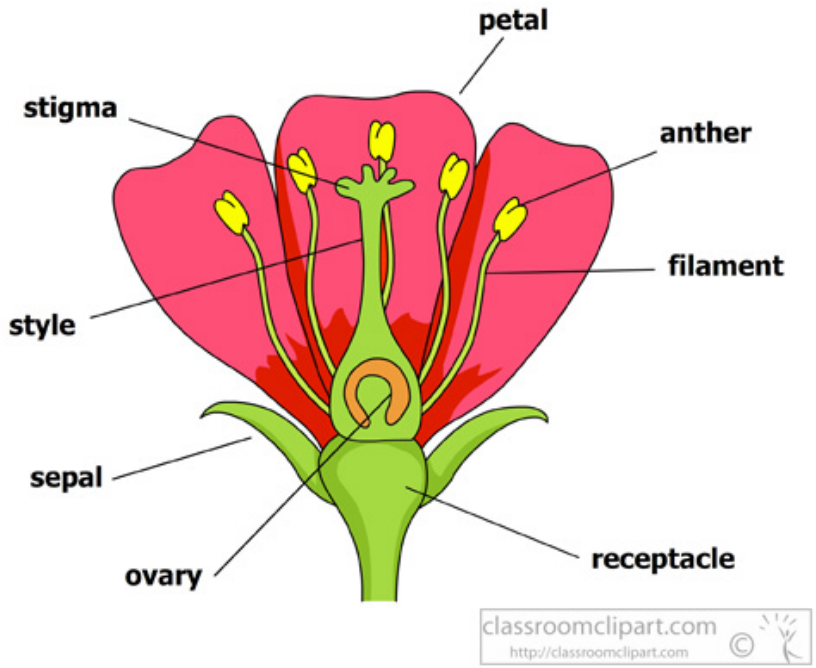
Step 4 **Select model:** identify relationship

$$\text{Data} = \text{Model} + \text{Error}$$

Step 5 **Test model:**

Step 6 **Communicate insights:**

Knowing Data: Iris Types



Metadata of Iris

col. name	example	description
`Sepal.Length`	5.1	length of sepal
`Sepal.Width`	3.1	width of sepal
`Petal.Length`	1.7	length of petal
`Petal.Width`	0.2	width of petal
`Species`	virginica	type of iris

Iris Types

versicolor

virginica

setosa

Exploratory Data Analysis in R

- **Describe:**

```
summary(iris)     ; str(iris)     ;  head(iris,n=10)
tail(iris,n=10)   ;  table(iris[,2])

set.seed(953)     ;  iris[sample(1:nrow(iris),size=10),]
sapply(iris,length)      ## why error w/ 'mean'?
quantile(iris$Sepal.Length)      ; aggregate(.~Species,mean,data=iris)
```

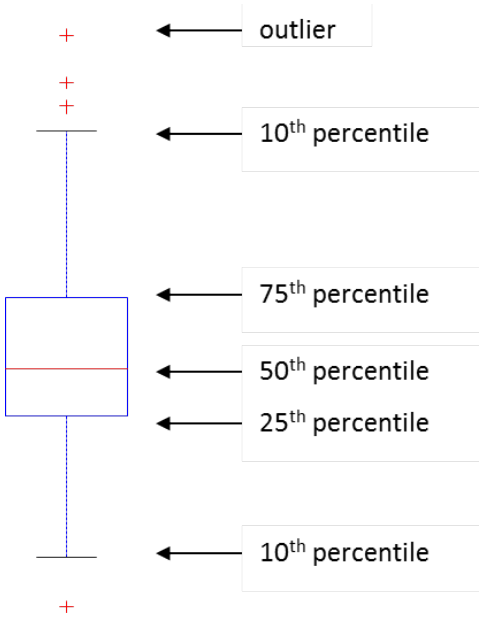
- **Advance:**

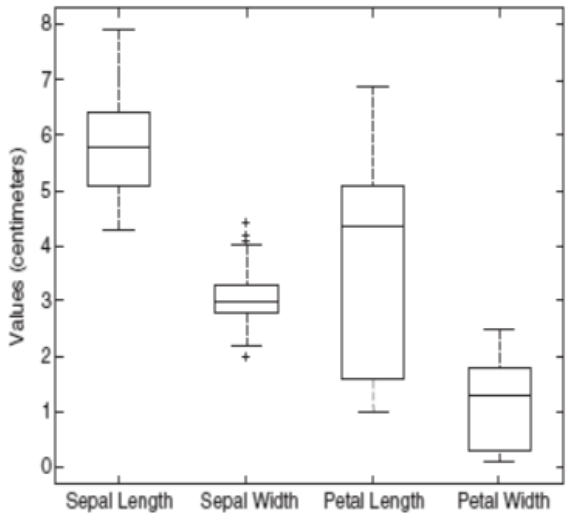
```
require(corrplot)
corrplot(cor(iris[,1:4]),method="ellipse",type="upper",order="hclust")

require(moments)
moments::all.moments(iris$Petal.Length)
kurtosis(iris$Sepal.Width)       ;  skewness(iris$Petal.Length)

iris.stdr <- sapply(iris[,1:4],function(o){ (o-min(o))/(max(o)-min(o)) } )
iris.norm <- sapply(iris[,1:4],function(o){ (o-mean(o))/sd(o) } )
```

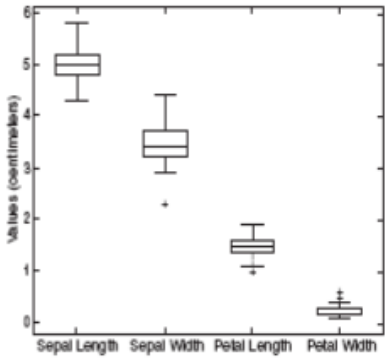
- **Visualize:**

```
stem( iris$Petal.Length)
hist(iris$Sepal.Length,n=10)
plot(density(iris$Sepal.Length))   ;  rug(iris$Sepal.Length)
plot(iris$Sepal.Length,iris$Petal.Length,cex=0.5)
pairs(iris[1:4],pch = 21, col=iris$Species)

attach(iris)
plot(Sepal.Length,Sepal.Width, pch = 21, bg = c("red", "green", "blue")[Species])
detach(iris)
```

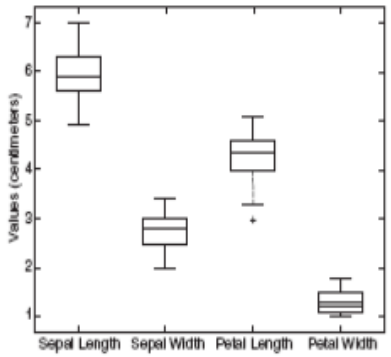
Box-Whisker Plot



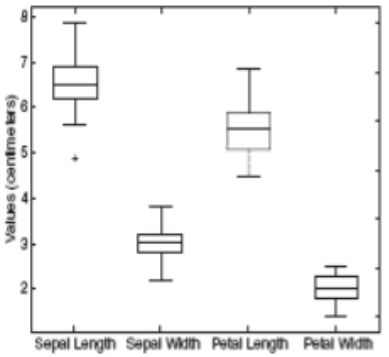




(a) Setosa.



(b) Versicolour.



(c) Virginica.

Basic Data Handling & Data Quarry

- **Checking**

```
dim(unique(iris))      ;   iris[duplicated(iris),]
all(complete.cases(iris))  ;  all(complete.cases(MASS::Cars93))
```

- **Grouping**

```
temp <- iris
widthRange  <- c(0.75,1.75)
lengthRange <- c(2.50,5.00)
temp$dWidth <- "low"
midIdx <- which(temp$Petal.Width >= widthRange[1] & temp$Petal.Width < widthRange[2] )
temp$dWidth[midIdx]  <- "medium"
hihIdx <- which(temp$Petal.Width >= widthRange[2])
temp$dWidth[ ] <- "high"
temp$dLength <- "low"
midIdx <- which(temp$Petal.Length >= lengthRange[1] & temp$Petal.Length < lengthRange[2])
temp$dLength[midIdx]  <- "medium"
hihIdx <- which(temp$Petal.Length >= lengthRange[2])
temp$dLength[hihIdx] <- "high"
```

- **Results**

```
table(temp$dLength,temp$dWidth,temp$Species)
ftable(temp$dLength,temp$dWidth,temp$Species)
xtabs(Petal.Width~dLength+dWidth,data = temp)

tempList <- as.data.frame(ftable(temp[c(7,6,5)]))
tempList[which(tempList$Freq > 0),]
```

Advance Data Handling & Data Quarry

- data.table

```
require(data.table)
iris.DT <- as.data.table(iris)
iris.DT[Petal.Width > 2.0 & Species == "virginica"][order(-Sepal.Length)]
iris.DT[Petal.Width > 2.0 & Species == "virginica",.N]
iris.DT[,.(.N,avgPW=mean(Petal.Width),sdpSL=sd(Sepal.Length)),by=Species]
iris.DT[,lapply(.SD,mean),by=Species]
```

- dplyr

```
require(dplyr)
glimpse(iris)
filter(iris,Petal.Width > 2.0 & Species == "virginica")
head(select(iris,starts_with("Petal")))
head(arrange(iris,Petal.Width,desc(Petal.Length)))
tail( summarise(iris,mean(Petal.Length),sd(Petal.Length) ))
summarise(group_by(iris,Species),mean(Petal.Length),sd(Petal.Length) )

iris %>% filter(Petal.Width > 2.0 & Species == "virginica") %>%
  select(starts_with("Petal")) %>% arrange(Petal.Width,desc(Petal.Length)) -> result
```

- ggplot2

```
require(ggplot2)
gg1 <- ggplot(iris,aes(Petal.Width)) + geom_histogram() + geom_rug()
gg2 <- ggplot(iris,aes(x=Species,y=Petal.Width)) + geom_violin(fill="blue")
gg3 <- ggplot(iris,aes(x=Petal.Length,y=Petal.Width)) + geom_point() + geom_smooth()
```

Multi-Dimensional Data Analysis (OLAP)

- **What:** A method to represent raw data as array (fact & quantity)
- **Criteria:**

type	width	length
low	$[0.00, 0.75)$	$[0.0, 2.5)$
medium	$[0.75, 1.75)$	$[2.5, 5.0)$
high	$[1.75, \infty)$	$[5.0, \infty)$

- **Result:**

Length	Width	Species	Count
low	low	Setosa	50
medium	medium	Versicolour	47
medium	medium	Virginica	1
medium	high	Versicolour	1
medium	high	Virginica	5
high	medium	Versicolour	2
high	medium	Virginica	4
high	high	Virginica	40

Useful Command in R

- **Data:**

```
intSix <- 6L         ; relSix <- 6.   ; cplSix <- 6.+0i        ## atomic data structure
txtSix <- "Six"      ; isSiz  <- T    ; facSix <- as.factor(6)## atomic data structure
uniVec <- runif(16) ; pnorm(0)    ;  dnorm(0) ## density at x=0
mixVec <- c(rnorm(6),runif(10)  ## vector
matrix(unifVec,ncol=4)     ;  array(mixVec,c(4,2,2)
myList <- list(unif=unifVec,mix=mixVec)
my.DF  <- as.data.frame(myList)  ; is.data.frame(my.DF) ;  typeof(my.DF)
unlist(mixVec)
```

- **Date/Time:**

```
Sys.Date()           ; Sys.time()     ; Sys.getenv()
Sys.setlocale(locale="Thai")

require(lubridate)
dateList <- as_date(format(Sys.Date() + -30:30 ,"%Y-%m-%d"))

format(dateList[6],"%Y-%m-%d %H::%M::%S %Z")

table(weekdays(dateList))
max(dateList) - min(dateList)
as.period(max(dateList) - min(dateList))
```

- **String**

```
  first  <- 'The'     ; third   <- 'Statistician'
  myName <- paste( c(first, 'Chemical', third), sep = ' ')
  strsplit(my.name, split = ' ')
  my.name.idx  <- grepl('ic',my.name)

  require(stringr)  ## package for easy string manipulation
```

Useful Package on Data Manipulation

- `readr` combined package for read data (S, xlsx, json, web scrap, xml)
- `tidyr` arrange and organize column/row/value or **data tidying**
- `stringr` combine search split of string or **string manipulation**
- `dplyr` data query with SQL-like
- `reshape2` data **table reshape** short-format and long-format tables
- `data.table` heavy duty/multi-core version of `data.frame` with joining and query features as well as table reshape
- `lubridate` convert and algebra of date-time
- `caret` multi-core data mining/machine learning for cross-validation
- `summarytools` lazy data frame summary and exploration

Classification Task

Definition

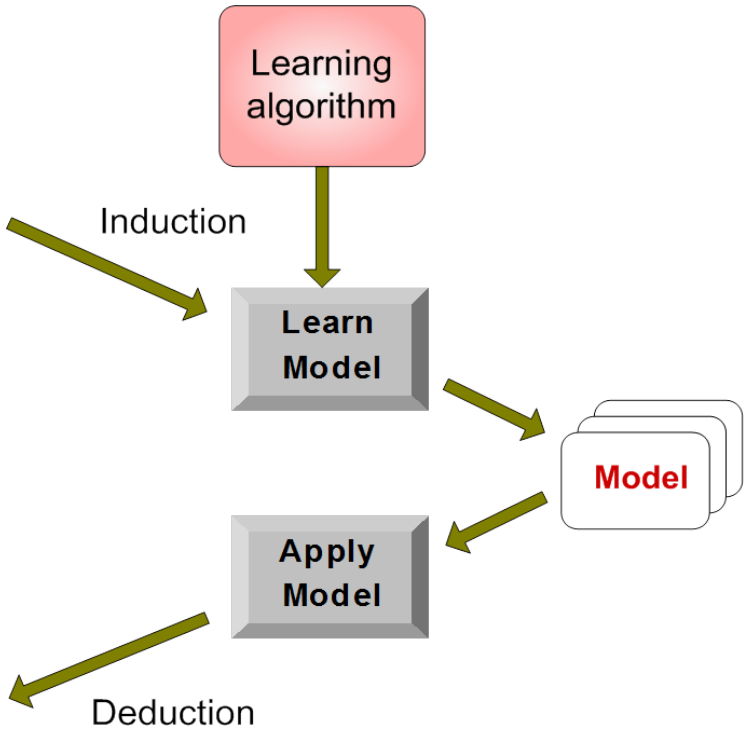
Given a training data, find a predicting model that can assign class for testing data/ unseen record

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Issues in Classification

Data Preparation

- **Data Cleaning** pre-processing data to reduce noise & handle missing values
- **Feature selection** removing irrelevant/redundant attributes
- **Data transformation** generalizing and normalizing data

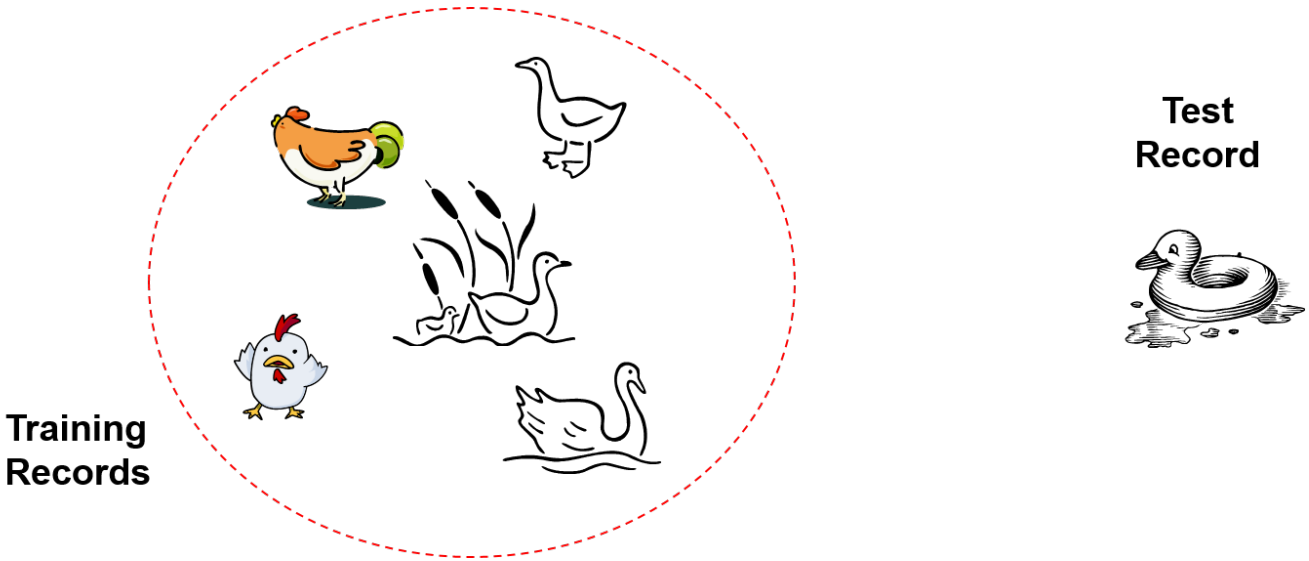
Model Evaluation

- **Accuracy:** measurement, e.g., GINI, entropy
- **Speed:** time to construct/use the model
- **Robustness:** how well to handle noise and NA
- **Interpretability:** insight provided by model $\rightarrow$ usefulness & clarity of rules

Example and Type

- **Example**
  - Predicting tumor cells (benign or malignant)
  - Classifying credit card transactions (legitimate or fraudulent)
- **Type**
  - k-Nearest Neighborhood
  - Decision Tree/ Random Forest
  - Rule-based/ logistic regression
  - Support Vector Machines
  - Artificial Neural Network
- **Performance:** Error rate $= \frac{\#\text{ wrong prediction}}{\text{total prediction}}$

k-Nearest Neighborhood

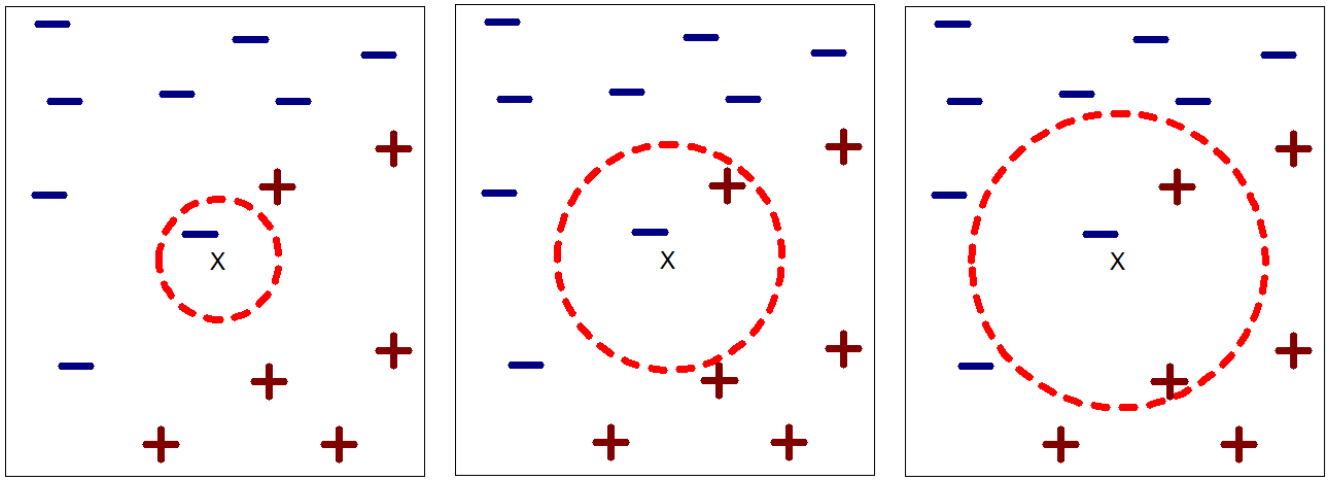


Idea

If it walks like a duck, quacks like a duck, then it's probably a duck

- **input:** training data, distance, k-comparison
- **output:** the class of majority of k-nearest neighborhood

k–NN Classification



(a) 1-nearest neighbor (b) 2-nearest neighbor (c) 3-nearest neighbor

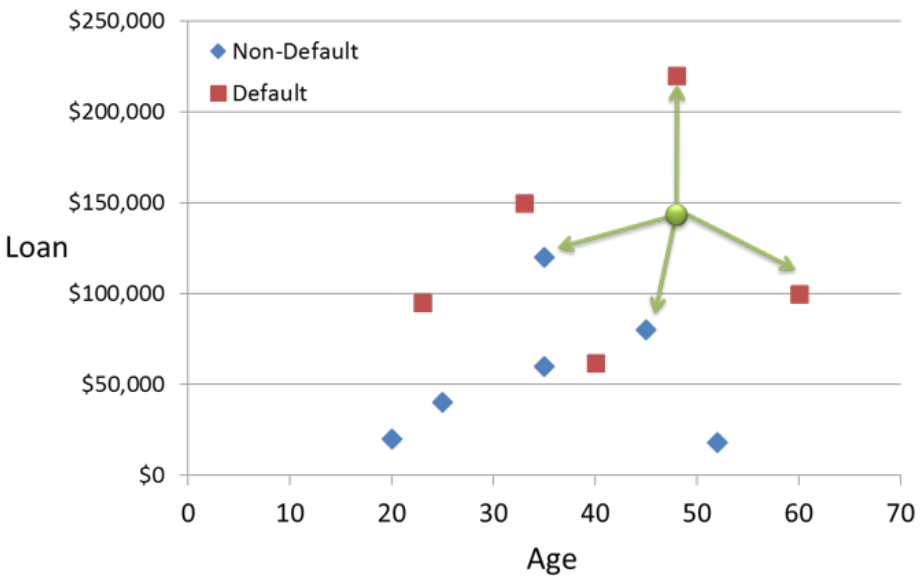
Issues

- **Advantage:** little information as long as distance function
- k–**NN Method:** "lazy leaner"– no model, not adaptive, no computation easy
- k**-value:** too large $\rightarrow$ overlap ; too small $\rightarrow$ noise
- **R command:** knn$(\cdot)$ in 'class' package

k–NN distance and example

Classification Question

Is 48 year old with loan 142k dollar default on his loan?

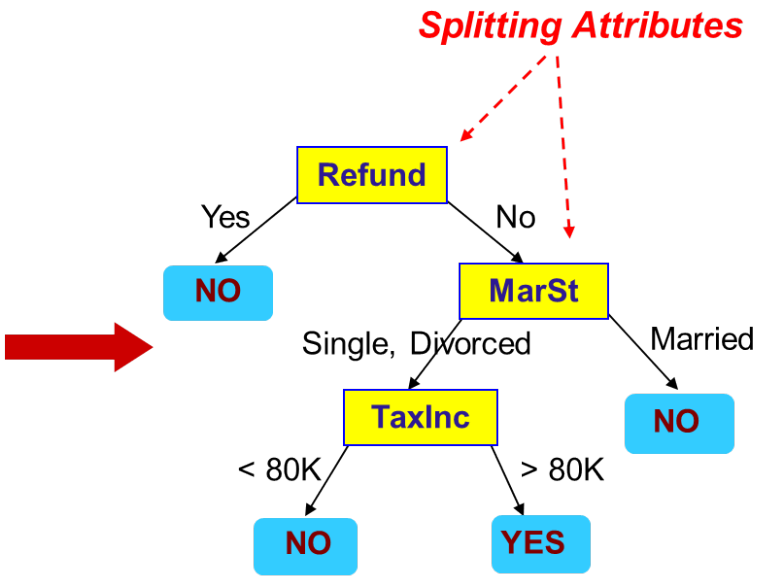


Age	Loan	Default?	Distance
25	40k	N	102k
35	60k	N	82k
45	80k	N	62k
20	20k	N	122k
35	120k	N	122k
52	18k	N	124k
23	95k	Y	47k
40	62k	Y	47k
60	100k	Y	42k
48	220k	Y	42k
33	150k	Y	42k

- How does your result change if loan is scaled?
- Based on figure, can you divide the 'default' area? ## why not?

Decision Tree

	categorical	categorical	continuous	class
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



- **root node:** top node with no incoming edges/ more outgoing edges
- **internal node:** node, one incoming edge/ two outgoing edges
- **leaf:** nodes, exact one incoming edge/ no outgoing edges

Hunt's Algorithm

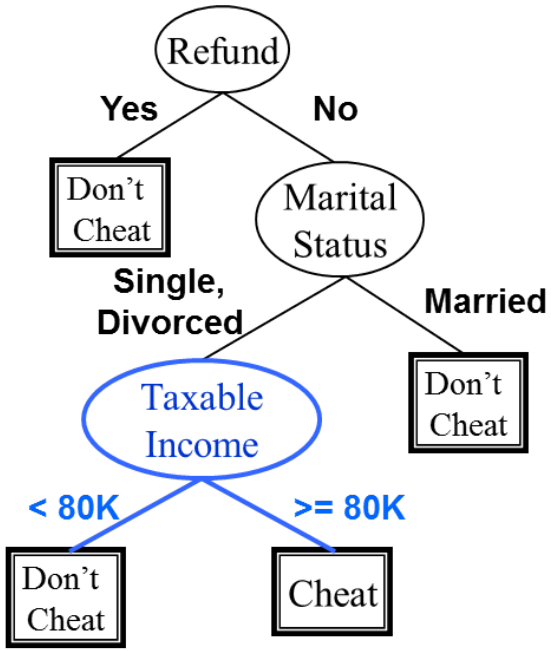
If there is more than one class **then**

separate into small subsets by attribute testing condition

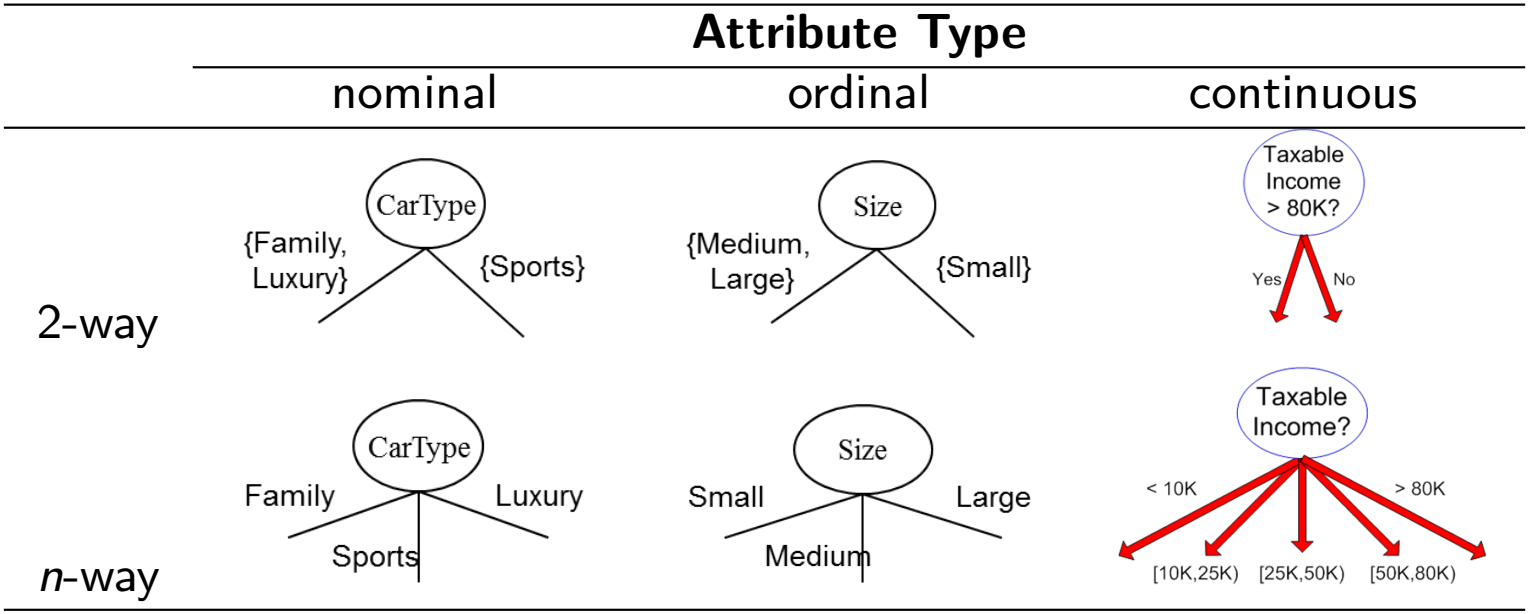
else subset is leaf

back to previous node and repeat alg

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



How to Specify Test Condition?



Example

C0: 5
C1: 5

Non-homogeneous,

High degree of impurity

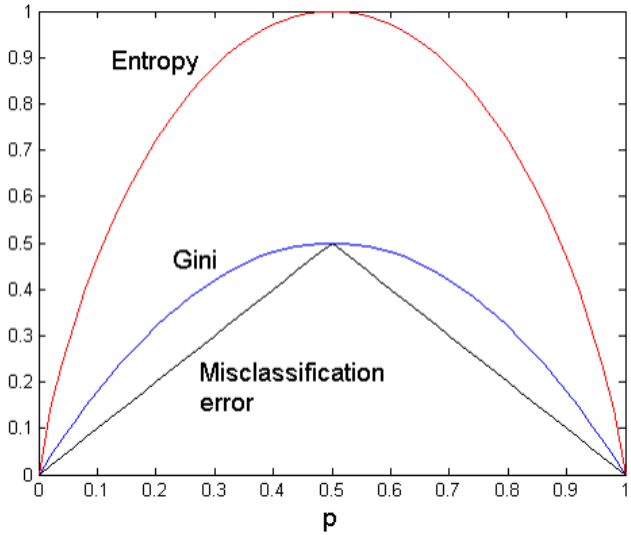
C0: 9
C1: 1

Homogeneous,

Low degree of impurity

Can we quantify the split?

How to measure for best split?



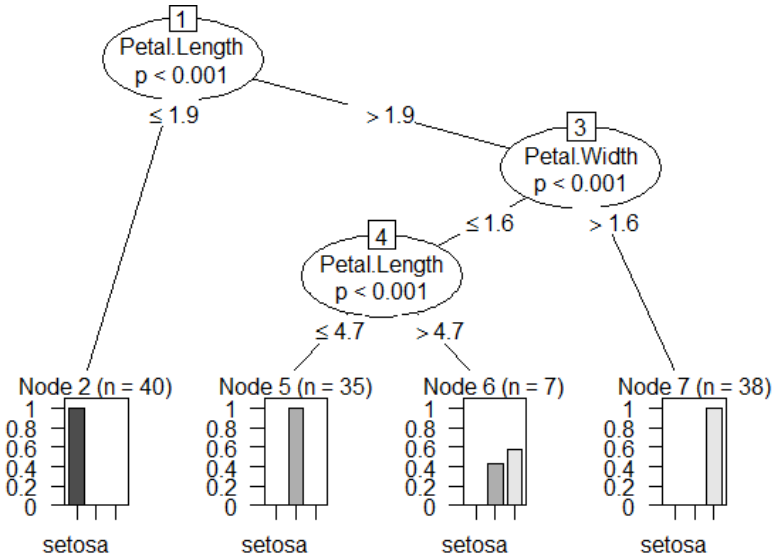
Let p_i is fraction of class i at node t from total c classes

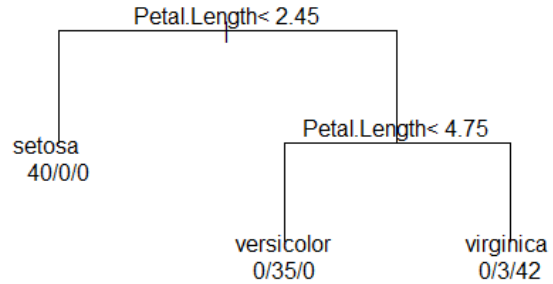
- **Entropy:** $Entro(t) = -\sum_{i=0}^{c-1} p_i \log_2 p_i$
- **Gini:** $Gini(t) = 1 - \sum_{i=0}^{c-1} p_i^2$
- **Classification Error:** $CE(t) = 1 - \max_i p_i$

Example

	Count		Value		
Example	Cls1	Cls2	Gini	Entropy	Error
Ex1	0	6	$1-0^2-1^2$	$-0-0$	$1-\max\{0,1\}$
Ex2	1	5	$1-(1/6)^2-(5/6)^2$	0.65	$1-\max\{1/6,5/6\}$
Ex3	3	3	$1-(3/6)^2-(3/6)^2$	1.00	$1-\max\{3/6,3/6\}$

Iris Classification





ctree(·) in 'party' package rpart(·) in 'rpart' package

Issues

- **data issue:** over-fitting b/c noise or insufficient data; fragment
- **tree issue:** tree replication; require multiple attributes

Iris: Classification I

- **Sampling:**

```
set.seed(937)
row.samp  <- sample(1:nrow(iris),size=30) ## random sampling
iris.trin <- iris[-row.samp,] ; iris.test <- iris[row.samp,]
table(iris.trin$Species)

set.seed(937)
require(splitstackshape)
iris.trin2 <- stratified(iris,"Species",size=0.8,keep.rownames=T) ## stratified sampling
iris.test2[as.numeric(iris.trin2$,rn),]
```

- **k-NN:**

```
require(class)
iris.knn <- knn(iris.trin[,1:4],iris.test[,1:4],cl=iris.trin[,5],k=5)
table(iris.knn,iris.test[,5])
```

- **DecTree:**

```
require(party)
iris.ctree  <- ctree(Species~.,data=iris.trin) ;  plot(iris.ctree)
iris.result <- predict(iris.ctree,newdata=iris.test[,1:4])
table(iris.result,iris.test[,5])

require(rpart)
iris.rpart <- rpart(Species~. ,data=iris.trin
                    ,control=rpart.control(minsplit = 20, xval = 81, cp = 0.01))
iris.rpart$cptable
plotcp(iris.rpart)
require(rpart.plot)
prp(iris.rpart,type=2,extra=104)
```

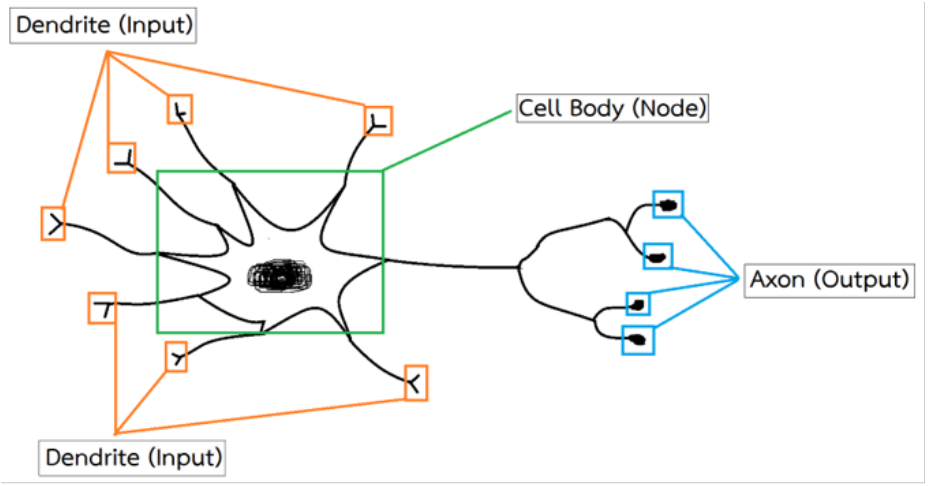
Confusion Matrix

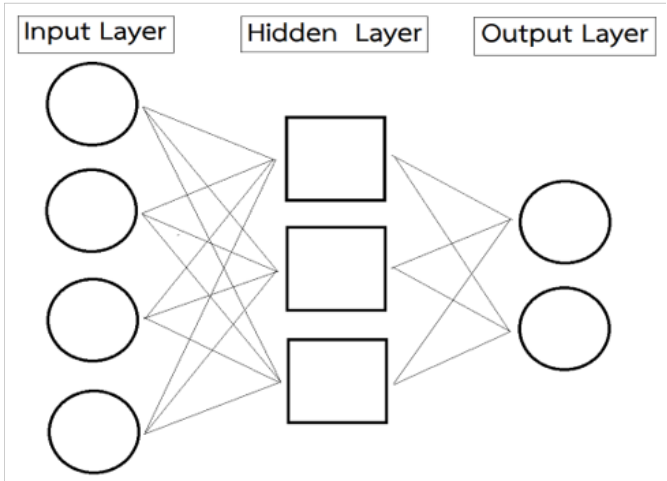
		PREDICTED CLASS	
		Class=Yes	Class=No
ACTUAL CLASS	Class=Yes	a (TP)	b (FN)
	Class=No	c (FP)	d (TN)

Measurement

- **Accuracy:** how often model correct $\frac{TP+TN}{Total}$
- **Misclassification:** how often model wrong $\frac{FP+FN}{Total}$
- **Precision:** how often model say **yes** & correct $\frac{TP}{TP+FP}$
- **Specificity:** how often model say **no** & correct $\frac{TN}{TN+FN}$
- `carat::confusionMatrix()` for calculation, rating, and statistic

Artificial Neural Network

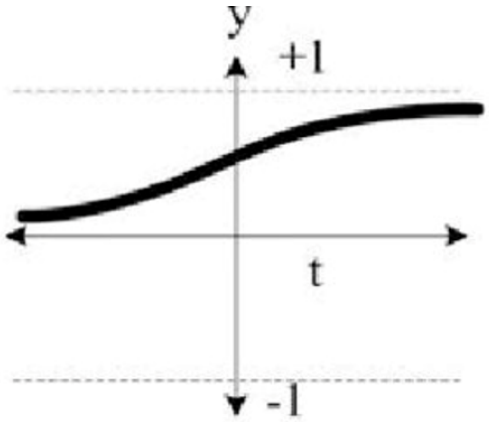




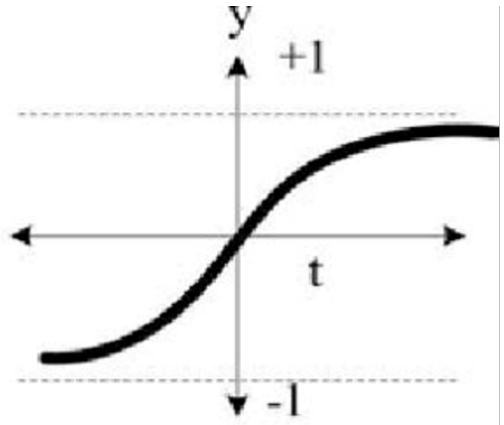
Intro to ANN

- **What:** algorithm inspired by biological systems (human brain)
- **Application:** zip code recognition, speech
- **Issue:** transfer functions, hidden layer, node number

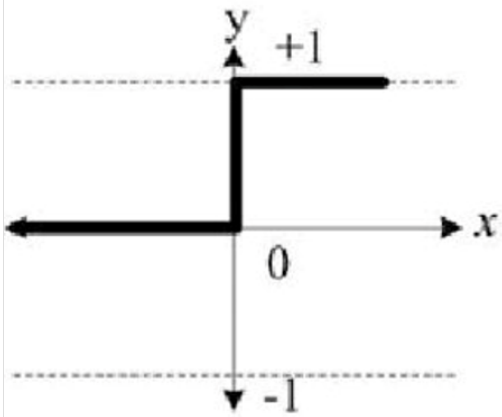
Some Transfers Function in ANN



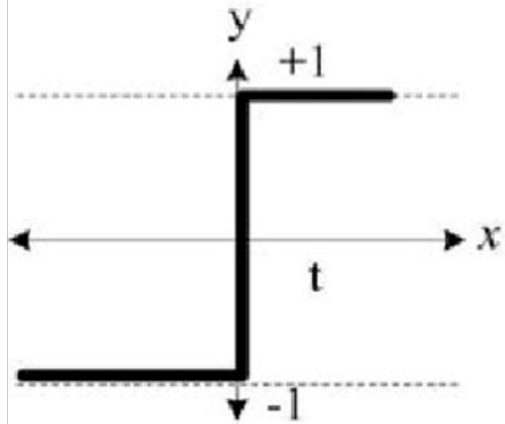
Sigmoid Function



Tansig Function

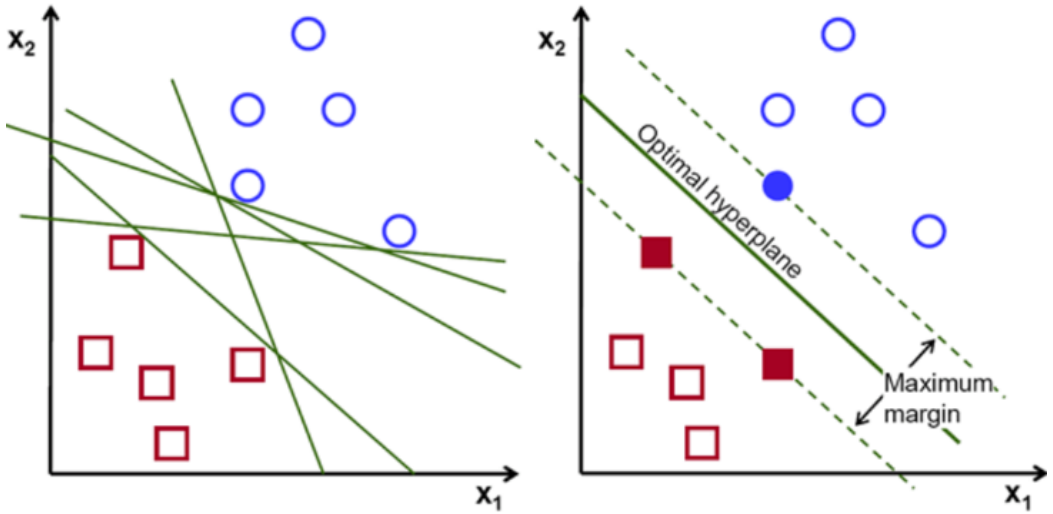


Step Function



Sign Function

Support Vector Machine



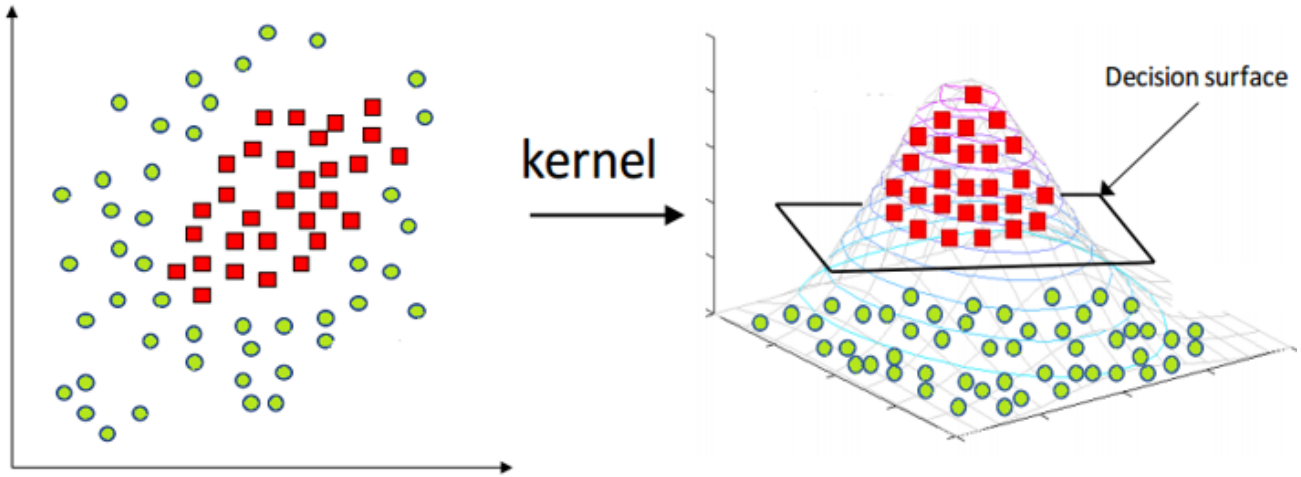
Intro to SVM

- **What:** algorithm inspired by linear separation problem
- **Math Model:** $\mathcal{D} \equiv \{\mathbf{x}_i, y_i\}$, where $y_i \in \{+1, -1\}$

$$\begin{aligned} \min \quad & \|\mathbf{w}\| \\ \text{s.t.} \quad & \\ y_i \left(\mathbf{w}^T \cdot \mathbf{x}_i + b\right) \quad & \geq \quad 1 \quad \forall i \in \mathcal{I} \end{aligned}$$

- **Issue:** scalability, kernel trick

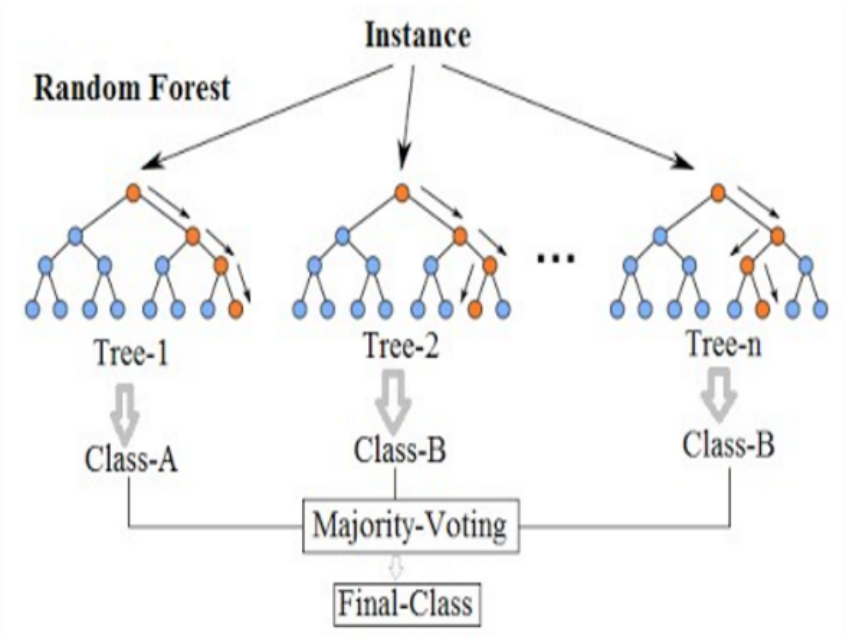
Kernel Trick in SVM



Kernel Trick

- **What:** dimension expansion for clean separation
- **Application:** zip code recognition, speech
- **Example:** polynomial, gaussian, hyperbolic tangent, sigmoid

Random Forest



Intro to Random Forest

- **What:** algorithm inspired by lot of lot decision trees
- **Concept:** sample data to create decision tree
- **Issue:** explanation

Iris: Classification II

- **ANN:**

```
set.seed(937)
require(neuralnet)
nn <- neuralnet(Species~.,data=iris.trin)
iris.nn <- predict(nn,newdata=iris.test[,1:4],hidden=2,act.fct="logistic")
plot(nn)
```

- **SVM:**

```
set.seed(937)
require(e1071)
supVm <- svm(Species~.,data=iris.trin)
iris.svm <- predict(supVm,newdata=iris.test[,1:4])
table(iris.svm,iris.test[,5])
```

- **rForest:**

```
require(randomForest)
rFrst <- randomForest(Species~.,data=iris.trin,importance=T)
iris.rFrst <- predict(rFrst,newdata=iris.test[,1:4])
table(iris.rFrst,iris.test[,5])
importance(rFrst)
```

- **caret :**

```
require(caret)
confusionMatrix(iris.rFrst,iris.test[,5])
confusionMatrix(iris.svm,iris.test[,5])
confusionMatrix(iris.nn,iris.test[,5])
```

Reference

[BG19] Brad Boehmke and Brandon M Greenwell.
Hands-on machine learning with R.
CRC press, 2019.

[NC20] Fred Nwanganga and Mike Chapple.
Practical machine learning in R.
John Wiley & Sons, 2020.

[Pat14] Manas A Pathak.
Beginning data science with R.
Springer, 2014.

[TSK16] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar.
Introduction to data mining.
Pearson Education India, 2016.