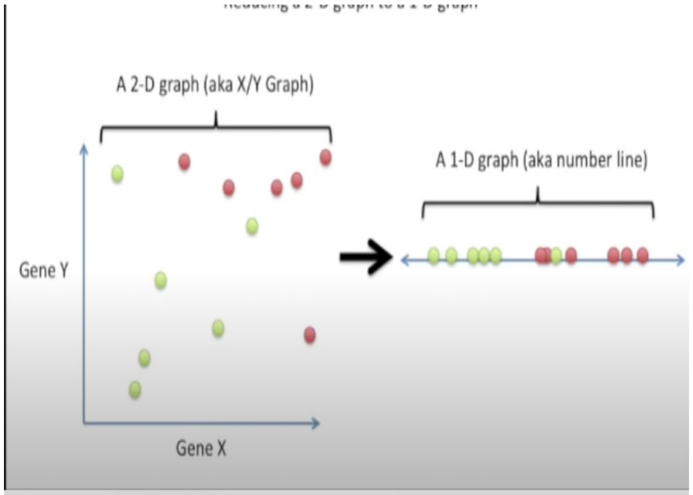
Ch 15: Non-Model Based Classification & Clustering

Introduction

- This chapter focuses on some alternative analyses that are not model-based or have a more general structure
  - Sections 15.1 and 15.2 gives us some non-model-based alternatives to logistic regression for classification:
    - 15.1: Linear discriminant analysis
    - 15.2: Graphical decision trees
  - Section 15.3 discusses ways to group observations into clusters

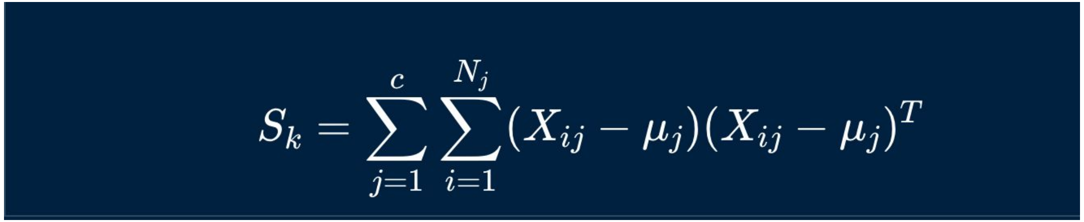
15.1 : Linear Discriminant Analysis (LDA)

- Best known non-model based method is called linear discriminant analysis.
- We want to “discriminate” between classes using observations.
- Goal: **maximize the component axes for class separation.** (*Raschka*)
- Opposed to other methods, LDA searches for the vectors that achieve this goal rather than vectors that best describe the data.
- LDA creates a linear combination of explanatory variable parameters which yields the largest mean differences between the desired classes. (StatQuest)



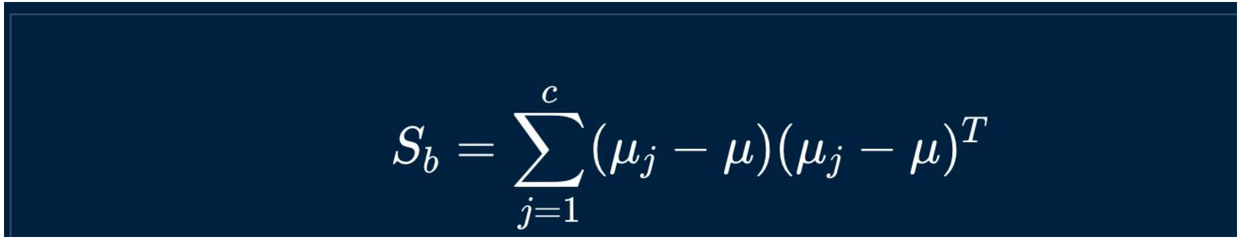
For all the classes (k) we define 2 kinds of variances:

Within-class variance:



$$S_k = \sum_{j=1}^{c} \sum_{i=1}^{N_j} (X_{ij} - \mu_j)(X_{ij} - \mu_j)^T$$

Between class variance:



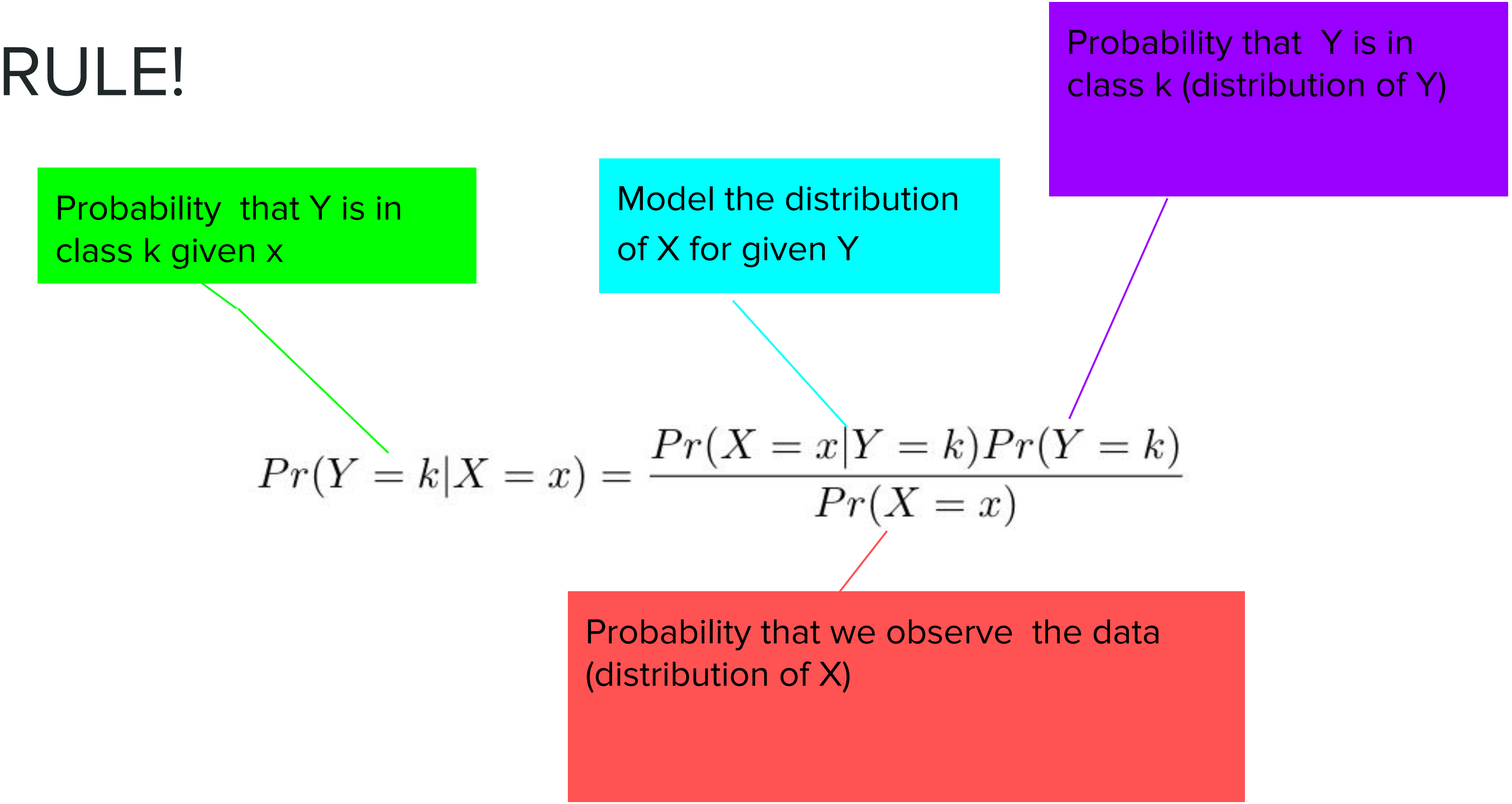
$$S_b = \sum_{j=1}^{c} (\mu_j - \mu)(\mu_j - \mu)^T$$

For LDA We assume uniform variance.

- Linear discriminant analysis makes an assumption about the distribution of X.
- We model the distribution of X for given Y (y = 0 or y = 1) (Gareth James et al., Springer, 138–168.)

$$f_k\left(X\right) \equiv \Pr\left(X = x \mid Y = k\right)$$

BAYES RULE!



Probability that Y is in class k given x

Model the distribution of X for given Y

Probability that Y is in class k (distribution of Y)

$$Pr(Y=k|X=x)=\frac{Pr(X=x|Y=k)Pr(Y=k)}{Pr(X=x)}$$

Probability that we observe the data (distribution of X)

So we want to maximize:

$$Pr(X = x | Y = k) Pr(Y = k)$$

(Gareth James et al., Springer, 138–168.)

- If we assume the distribution of X for given Y is normal (a.k.a. Gaussian), then we can calculate the function:

- “The solution is to maximize that difference between the squared distance between the means of the linear combinations of the parameters for the different classes.” (Agresti)

$$d_j(\mathbf{x}) = (\mathbf{x} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_j), \quad j = 0, 1.$$

- “In practice, we estimate these distances by substituting the sample means xi and x_0 and a pooled covariance estimate S”. (Agresti)

We then have this boundary

Fisher's linear discriminant function.

$$\boxed{(\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \mathbf{S}^{-1}\mathbf{x}} > (\bar{\mathbf{x}}_1 - \bar{\mathbf{x}}_0)^T \mathbf{S}^{-1}(\bar{\mathbf{x}}_1 + \bar{\mathbf{x}}_0)/2 - \text{logit}(\pi_0).$$

LDA vs. Logistic regression

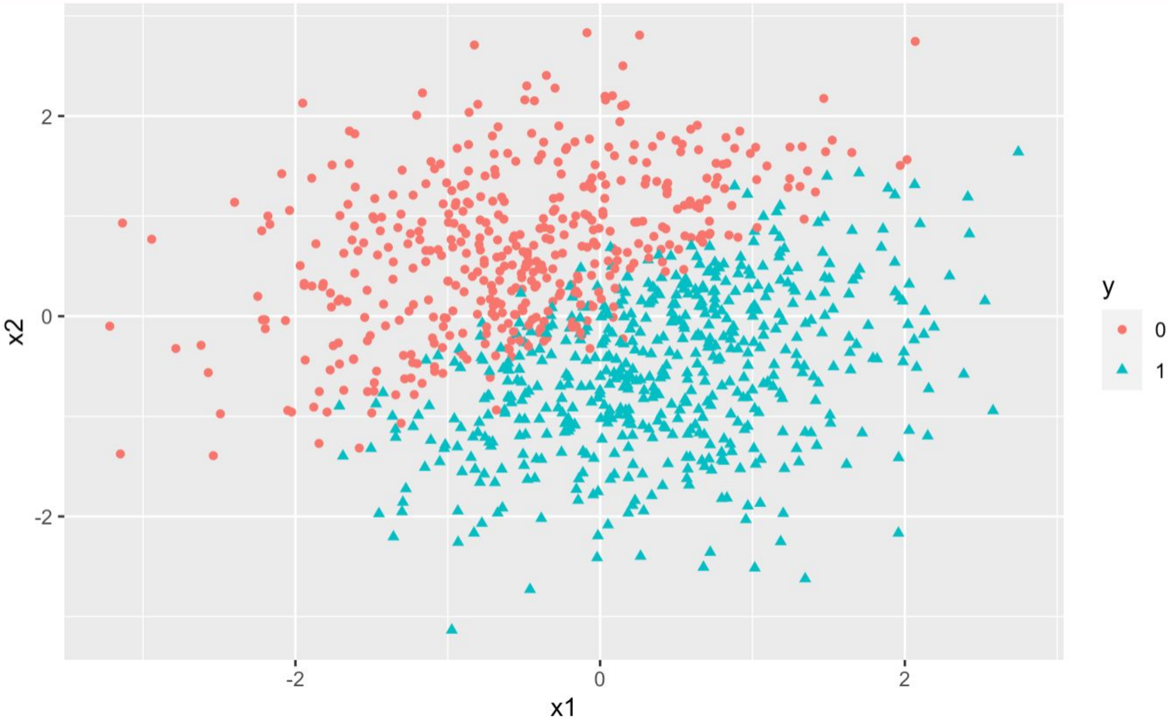
Logistic regression is more robust and has broader scope, as it makes no assumption about a distribution for X and merely assumes a binomial distribution for Y at each value of x.

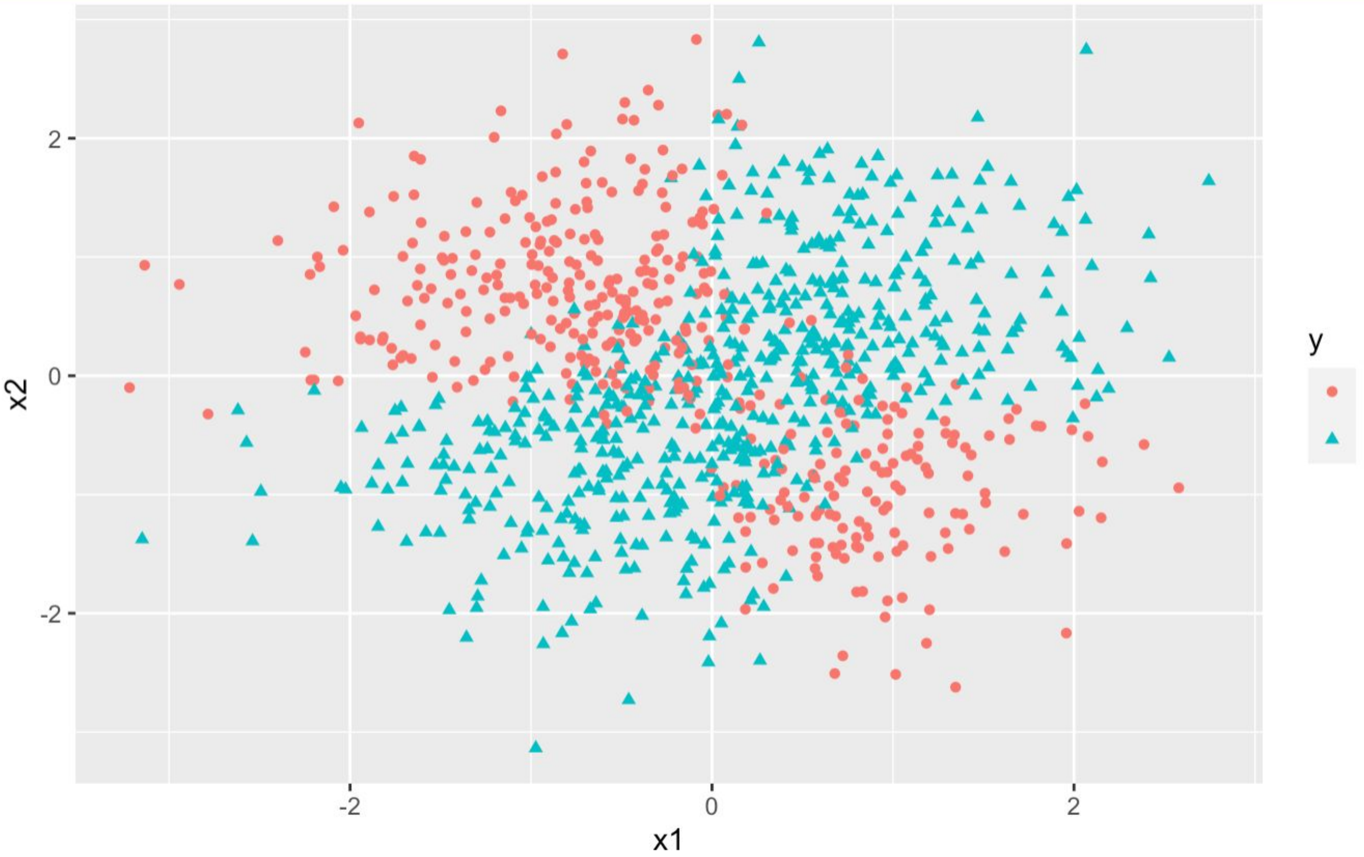
Also, logistic regression has the advantage over discriminant analysis of providing direct ways of summarizing effects of explanatory variables, through odds ratios

LDA out performs logistic regression when the data is normally distributed. (Gareth James et al., Springer, 138–168.)

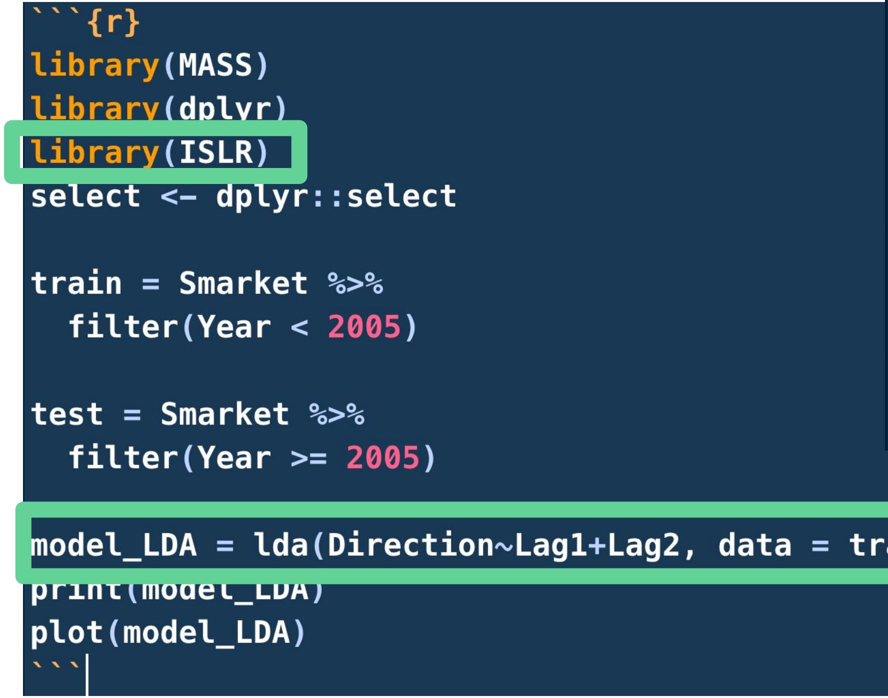
LDA performs better when there is a clear separation of classes

Plot showing linear separation

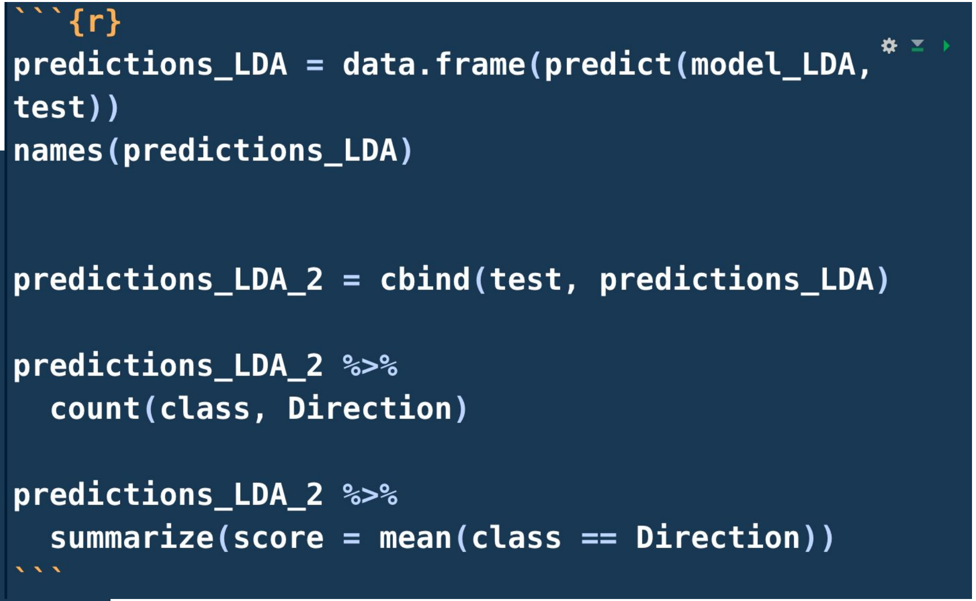



2
0
-2
x2
-2
0
2
x1
y
0
1

Doing LDA in R



~~~
```{r}
library(MASS)
library(dplyr)
library(ISLR)
select <- dplyr::select

train = Smarket %>%
  filter(Year < 2005)

test = Smarket %>%
  filter(Year >= 2005)

model_LDA = lda(Direction~Lag1+Lag2, data = train)
print(model_LDA)
plot(model_LDA)
```
~~~

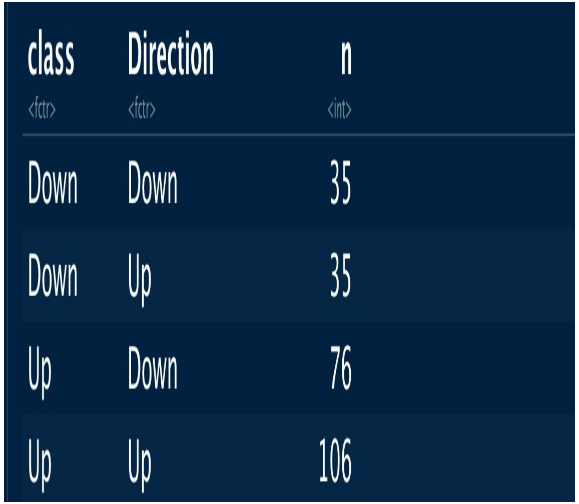


~~~
```{r}
predictions_LDA = data.frame(predict(model_LDA,
test))
names(predictions_LDA)


predictions_LDA_2 = cbind(test, predictions_LDA)

predictions_LDA_2 %>%
  count(class, Direction)

predictions_LDA_2 %>%
  summarize(score = mean(class == Direction))
```
~~~

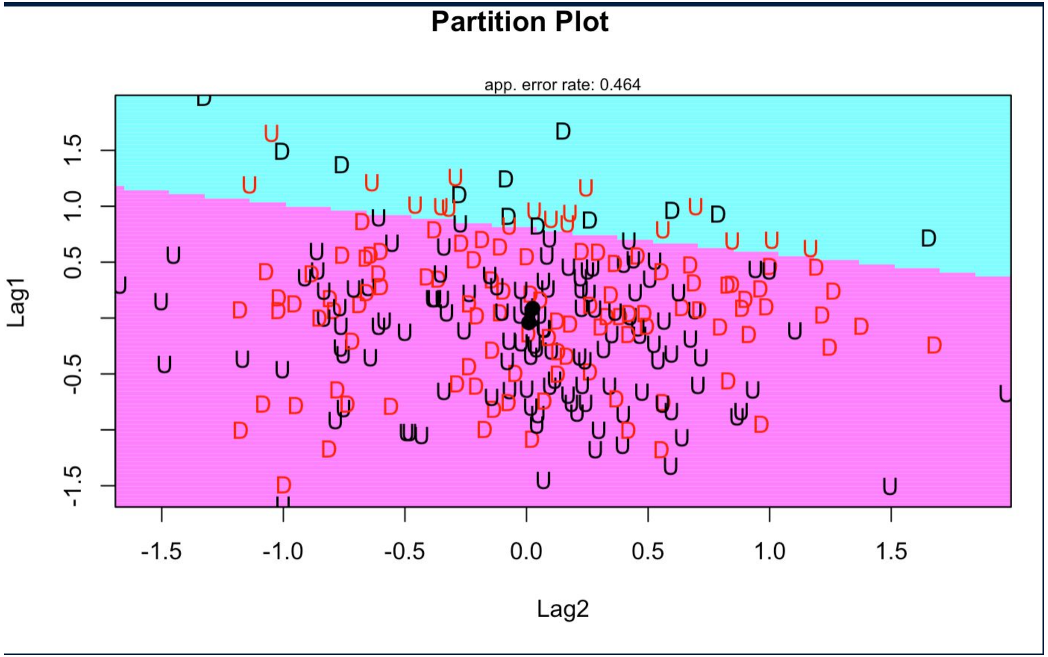
The syntax for the lda() function is identical to that of lm()

This comes from p. 161-163 of "Introduction to Statistical Learning with Applications in R" (Gareth James et al., Springer, 161-163.)

LDA results



class <fctr>	Direction <fctr>	n <int>
Down	Down	35
Down	Up	35
Up	Down	76
Up	Up	106

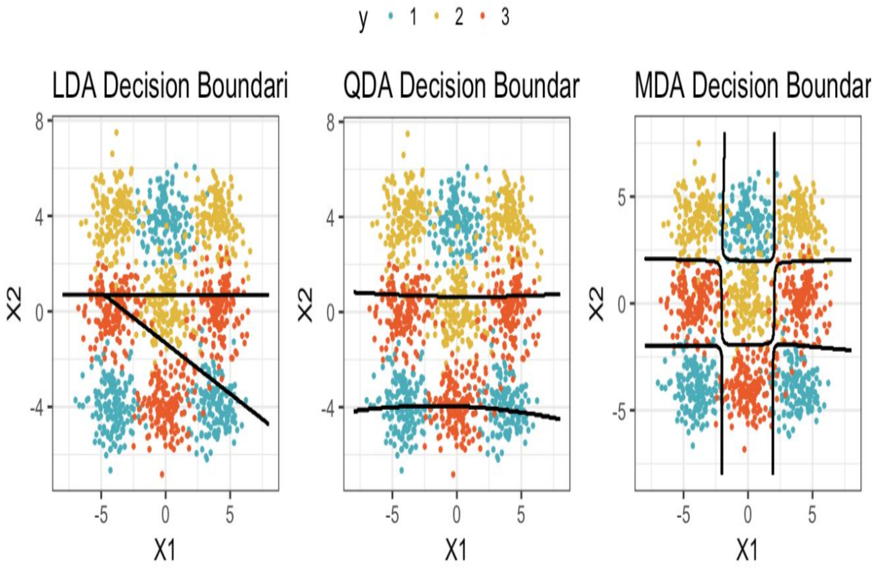


Other discriminant methods

Mixture discriminant analysis-

- The mixture of normals is used to obtain a density estimation for each class.
- Each data point has a probability of belonging to each class.
- Equality of covariance matrix, among classes, is assumed. (Charles)

Quadratic discriminant analysis- QDA is more flexible than LDA and it is where the we relax the assumption of uniform variance. (Charles)

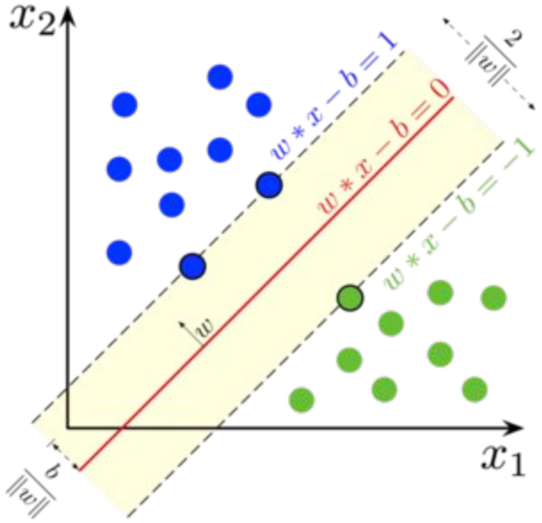


15.2.6: Support Vector Machines

Goal: To create a line or a hyperplane which separates the data into classes.

SVM algorithm:

- We find the points closest to the line from both classes.
- These points are called support vectors.
- Now, we compute the distance between the line and the support vectors.
- This distance is called the margin.
- Our goal is to maximize the margin. (nlp.stanford.edu)

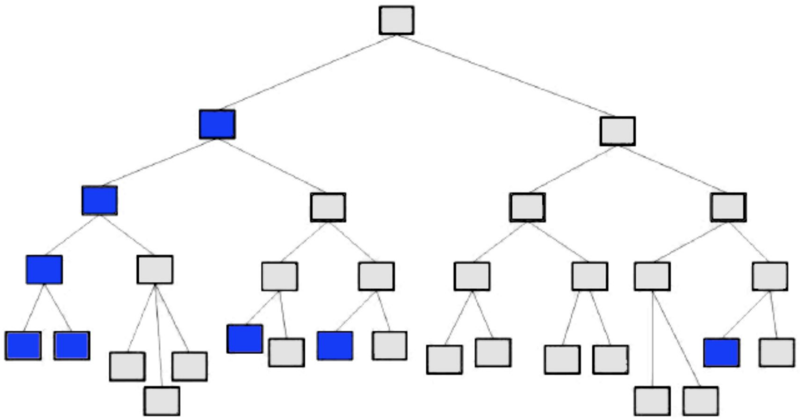


15.2: Classification Tree Based Prediction

- Tree based algorithms allow for some stability and ease of interpretation.

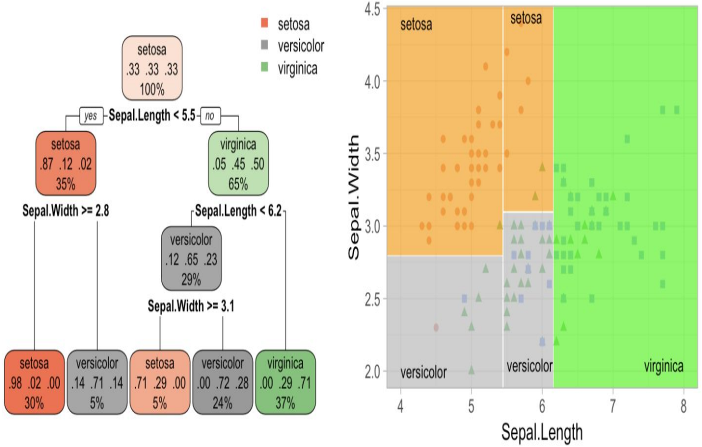
- They map nonlinear relationships quite well.

- An adaptable method of problem solving (classification or regression). (Vidhya)



Decision trees

- The **classification tree** method is a process that uses a sequential set of questions about x values to classify a prediction on y.
- These methods involve stratifying or segmenting the predictor space (x-values) into a number of simple regions.
- Set of X values are which y-hat = 1 has a simple form consisting of a set of rectangular regions.
- In order to predict a given observation, we typically use **the mean of the mode for the training observations in the given class**



()

- We use recursive binary splitting to grow a classification tree.
- One criterion for making the binary splits is the classification error rate
- We assign an observation in a region in to the most commonly occurring class
- **The classification error rate** is the fraction of the training observations in that region that do not belong to the most common class

$$E = 1 - max_k(\hat{p}_{mk})$$

- P_mk represents the proportion of number of observations in the mth region given for a kth class

(Gareth James et al., Springer, 303–332)

Minimizing the classification error

The usual way we would minimize the classification error rate is not sensitive enough to build a useful tree so we use these two methods instead

Gini index :

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}),$$

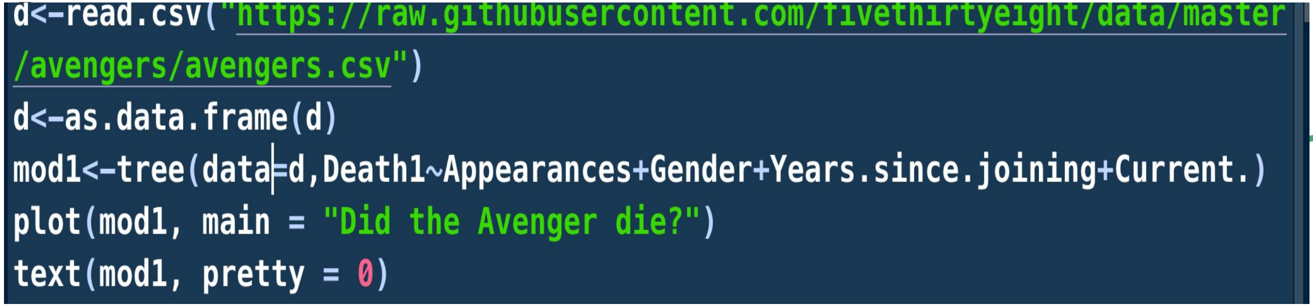
Entropy:

$$D = - \sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk}.$$

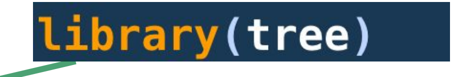
These methods measure purity, whether the node contains mostly observations from the same class

Example: Avengers and Death

- To illustrate components of the classification tree method, we will examine data about “the deaths of Marvel comic book characters between the time they joined the Avengers” to determine the probability of that character dying. (https://github.com/fivethirtyeight/data/tree/master/avengers)
- We are going to use these variables (Current?, Appearances, Gender, Years since joining) to predict whether the Avenger died.



```r
d<-read.csv("https://raw.githubusercontent.com/fivethirtyeight/data/master
/avengers/avengers.csv")
d<-as.data.frame(d)
mod1<-tree(data=d,Death1~Appearances+Gender+Years.since.joining+Current.)
plot(mod1, main = "Did the Avenger die?")
text(mod1, pretty = 0)
```

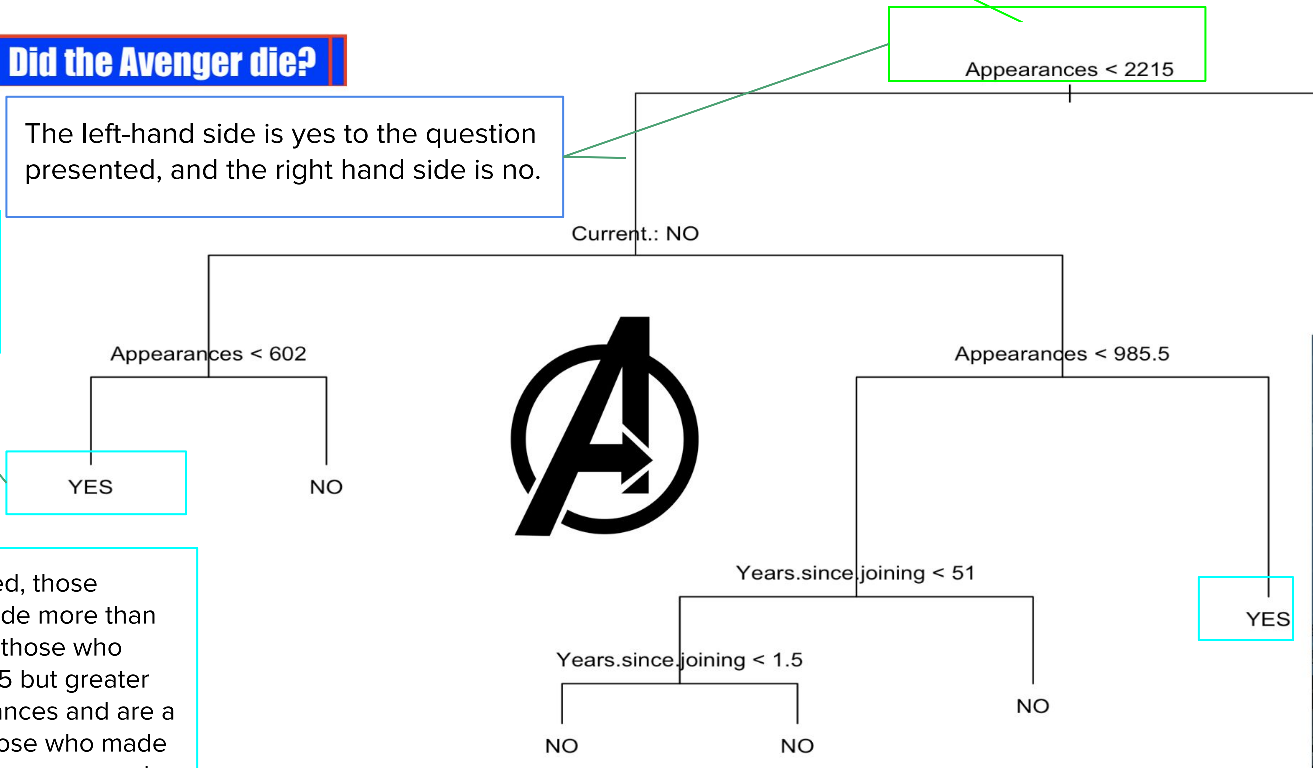


```r
library(tree)
```

Example: **Avengers and Death**

Did the Avenger die?

Most important indicator of death appears to be whether the number of appearances < 2215



The left-hand side is yes to the question presented, and the right hand side is no.

Terminal nodes are the predicted values of y in the model

As the tree predicted, those predicted to die made more than 2215 appearances, those who made less than 2215 but greater than 985.5 appearances and are a current Avenger, those who made less than 2215 appearances and are not a current Avenger and made less than 602 appearances

Pruning

$$\sum_{t} p(t)c(t) + [\lambda \times (\text{number of terminal nodes})],$$

- It turns out that trees tend to overfit to the data. This is because a tree can continue to grow until there are as many nodes as there are distinct observations in the data, a stopping rule is put into place to allow for some misclassification.
- So one way to not overfit is to use the pruning method.
- **Big Idea:** Build a big tree to start and cut the branches that are not adding to the performance of the tree. (Similar to feature selection in regression.)
- We can't start with a small tree because a not so useful branch at the start might lead to a more fruitful branch. Don't cut too early! (Gareth James et al., Springer, 303–332)

Tree based methods vs Logistic regression

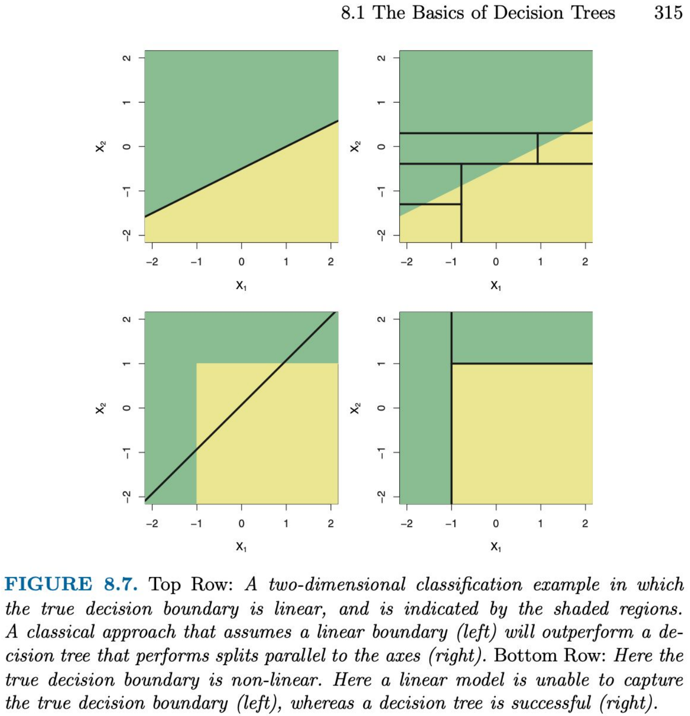
Decision trees perform better when there is a clear separation of classes.

A disadvantage of a classification tree compared with logistic regression modeling is the lack of smoothness

Decision Trees also require larger amounts of data to be useful

Decision Trees perform worse when the number of dimensions start outnumbering the number of data points

The classification tree method has low bias but high variance. Hard to generalize.



8.1 The Basics of Decision Trees 315

FIGURE 8.7. Top Row: *A two-dimensional classification example in which the true decision boundary is linear, and is indicated by the shaded regions. A classical approach that assumes a linear boundary (left) will outperform a decision tree that performs splits parallel to the axes (right).* Bottom Row: *Here the true decision boundary is non-linear. Here a linear model is unable to capture the true decision boundary (left), whereas a decision tree is successful (right).*

(Gareth James et al., Springer, 303–332)

Pros and cons of Trees

Pros:

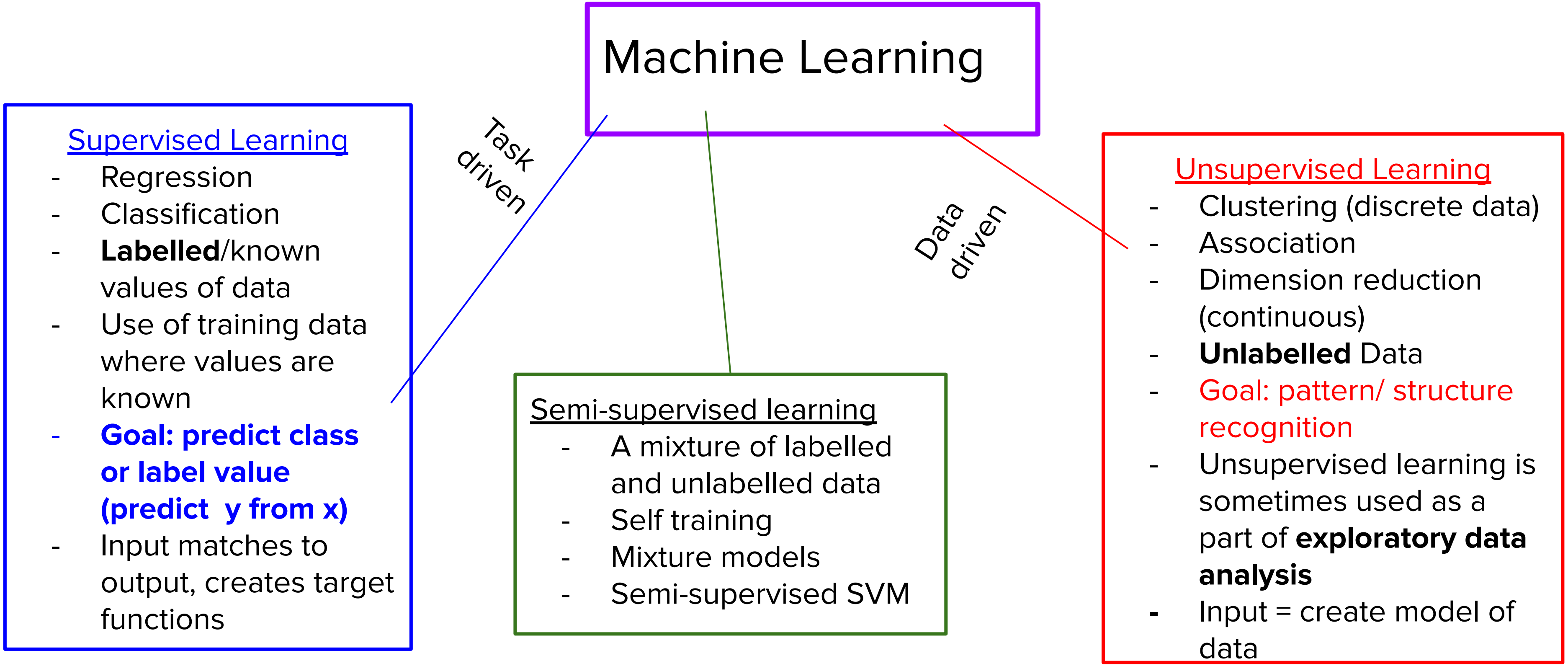
- Easy to understand and intuitive.
- Some argue that the decision trees more closely mirror human decision-making than than other classification approaches.
- Trees can be displayed graphically, and are easily interpreted (even better when small)
- Trees easily work with qualitative predictors without having to create dummy variables.
- Feature selection happens automatically.

Cons:

- Trees are not robust. This means a small change in data cause a large change in the final estimated tree.
- Trees tend to not have as great accuracy as other methods
- Trees tend to overfit to the data that they are fed.

(Gareth James et al., Springer, 303–332)

Ch 15.3.2 : Supervised vs. Unsupervised Learning



Machine Learning

Supervised Learning

- Regression
- Classification
- **Labelled**/known values of data
- Use of training data where values are known
- **Goal: predict class or label value (predict y from x)**
- Input matches to output, creates target functions

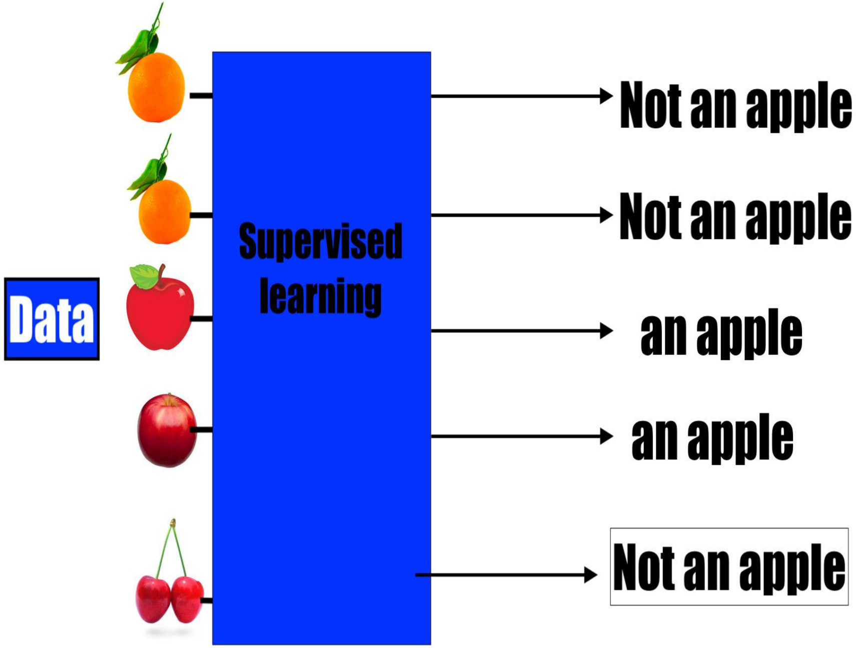
Semi-supervised learning

- A mixture of labelled and unlabelled data
- Self training
- Mixture models
- Semi-supervised SVM

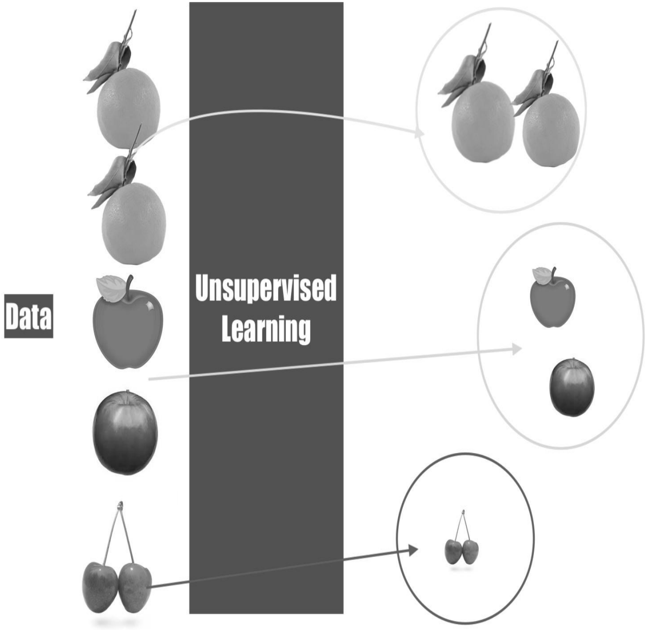
Unsupervised Learning

- Clustering (discrete data)
- Association
- Dimension reduction (continuous)
- **Unlabelled** Data
- Goal: pattern/ structure recognition
- Unsupervised learning is sometimes used as a part of **exploratory data analysis**
- Input = create model of data

Supervised learning



Unsupervised learning



Clustering

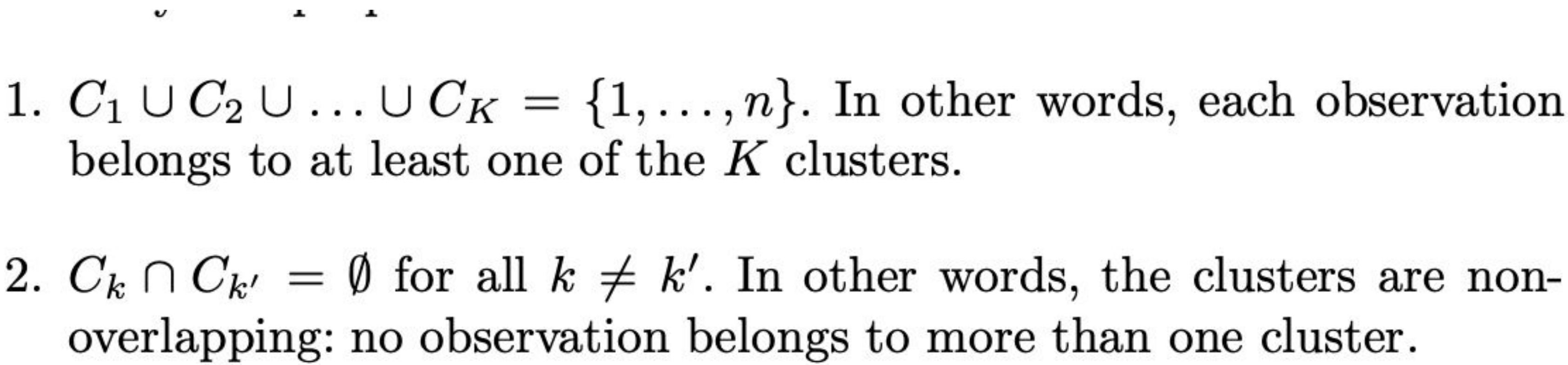
A **cluster** is a collection of data values that are grouped together because of selection of similarities that also make them dissimilar from data values in other groups.

Big Idea: Divide/partition the data values into distinct subgroups.

Observations within each group should be **similar** to each other and observations in **different** groups are different from each other!

We need to find a way to define similar and different.

Clustering Notation



1. $C_1 \cup C_2 \cup \ldots \cup C_K = \{1, \ldots, n\}$. In other words, each observation belongs to at least one of the K clusters.

2. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$. In other words, the clusters are non-overlapping: no observation belongs to more than one cluster.

Measuring dissimilarity

Clustering wants observations in one cluster to have low dissimilarity and observations from different clusters to have high dissimilarity.

The dissimilarity measure gives the proportion where the outcomes differ.

Using Jaccard Index is one method to measure dissimilarity. The jaccard index is defined as the size of intersection divided by the size of the union.

Dissimilarity index:

$$\frac{(b+c)}{(a+b+c+d)}$$

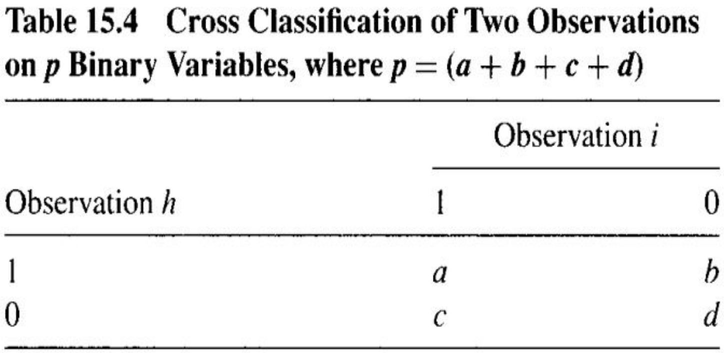


Table 15.4 Cross Classification of Two Observations on *p* Binary Variables, where *p* = (*a* + *b* + *c* + *d*)

	Observation *i*	
Observation *h*	1	0
1	*a*	*b*
0	*c*	*d*

Linkage: Pairwise dissimilarity

Linkage: Computes pairwise dissimilarities between observations in a cluster R and the elements in cluster S.

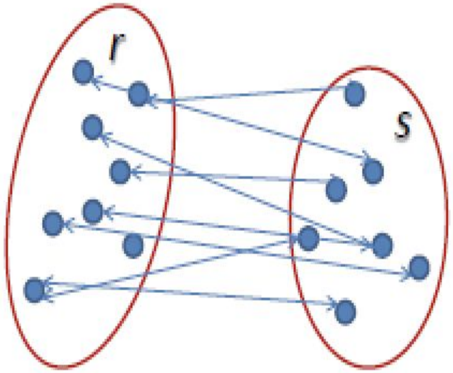
Average Linkage: The distance between two clusters is defined as the average distance between pairs of points in one cluster to points in another cluster.

Single Linkage: Distance between two clusters is defined as the the minimal distance between pairs of observations in one cluster to every point in the other cluster (inter-cluster distance).

Complete Linkage: Maximal inter-cluster distance.

Centroid: The distance between cluster means.

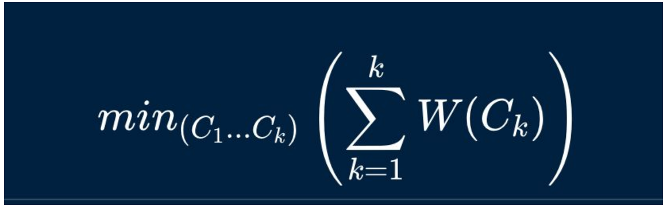
(Gareth James et al., 385–399.)



K-means Clustering: Partitions

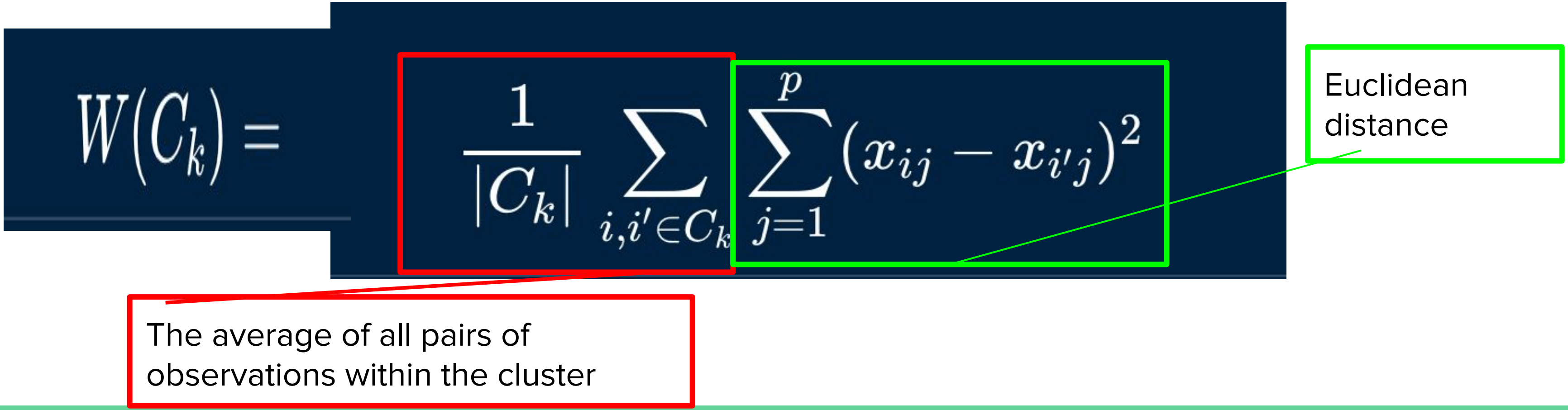
Goal: We want to partition the data into a predetermined number of subgroups (clusters).

To do this we want small within cluster variation:



$$min_{(C_1 ... C_k)} \left(\sum_{k=1}^{k} W(C_k) \right)$$

To solve this problem we have to calculate average Euclidean distances between the observations in the kth cluster (Gareth James et al.,385–399.):



$$W(C_k) = \frac{1}{|C_k|} \sum_{i,i' \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{i'j})^2$$

Euclidean distance

The average of all pairs of observations within the cluster

K-means algorithm

There are K^n ways to partition n observations into k-clusters. If k or n are large this becomes really hard.

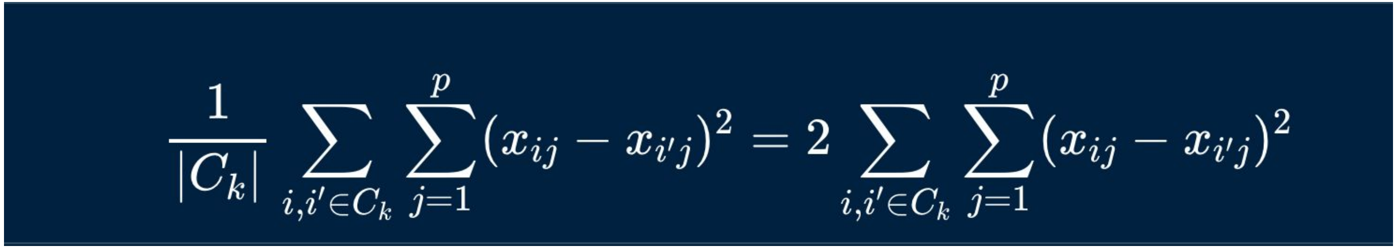
So we use the k-means algorithm to find the local optimum.

1. Randomly assign each observation to a cluster
2. Iterate until the cluster values stop changing
   a. For each of the k clusters, compute the vector of the p feature means for the observations in the k-th cluster (this is called the cluster centroid).
   b. Assign each observation to the cluster whose centroid is the closest to the observation’s value (using euclidean distance).

(Gareth James et al.,385–399.)

How does it work?

- The k-means Algorithm guarantees to decrease the value of the within cluster variation.
- Using this identity:



$$\frac{1}{|C_k|}\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2 = 2\sum_{i,i'\in C_k}\sum_{j=1}^{p}(x_{ij}-x_{i'j})^2$$

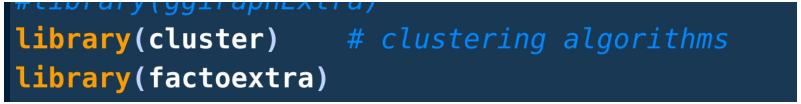
In Step 2.a the cluster means for each feature are the constants that minimize the sum-of-squared deviations.

In Step 2.b moving the observations based on Euclidean distance can only improve W(C_k).

This means while algorithm runs, the clustering obtained will continually improve until the result no longer changes. Therefore the within class variance will never increase.

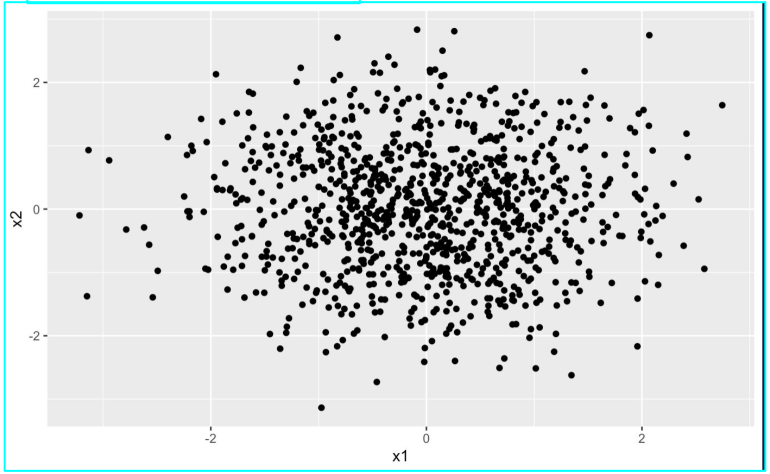
(Gareth James et al.,385–399.)

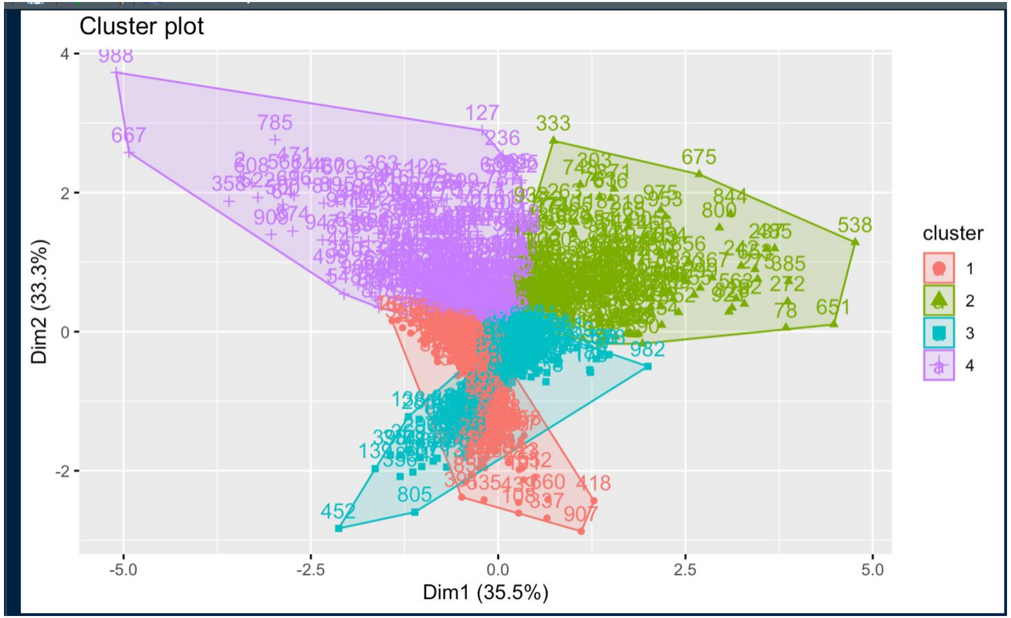
Example in R!

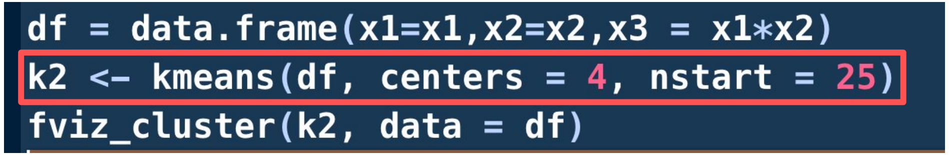


```r
#library(ggiraphExtra)
library(cluster)    # clustering algorithms
library(factoextra)
```

Data







```r
df = data.frame(x1=x1,x2=x2,x3 = x1*x2)
k2 <- kmeans(df, centers = 4, nstart = 25)
fviz_cluster(k2, data = df)
```

Drawbacks to K-means Algorithm

- K means finds a local optimum and not a global optimum.
- The results depend on the initial random assignment.
- Important: the algorithm is run multiple times to avoid being stuck.
- We also have to figure out what is the right predetermined number of clusters to get meaningful results.

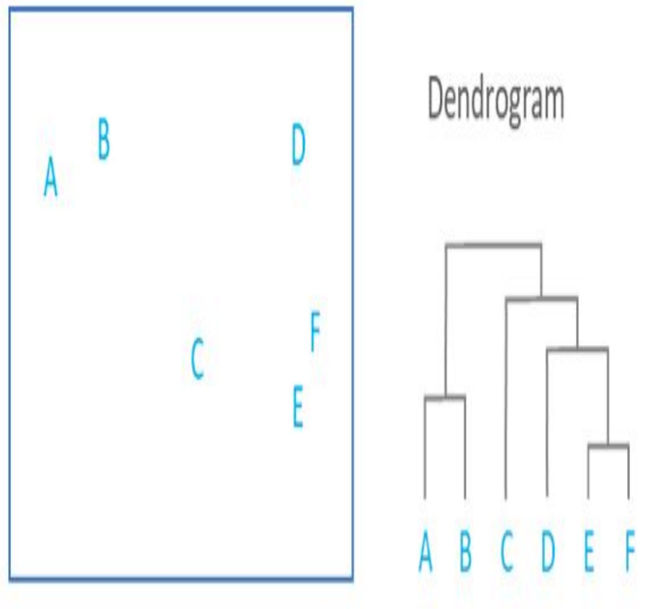
Hierarchical clustering

Goal: Adapt tree-based methods while no longer having to predetermine the number of clusters.

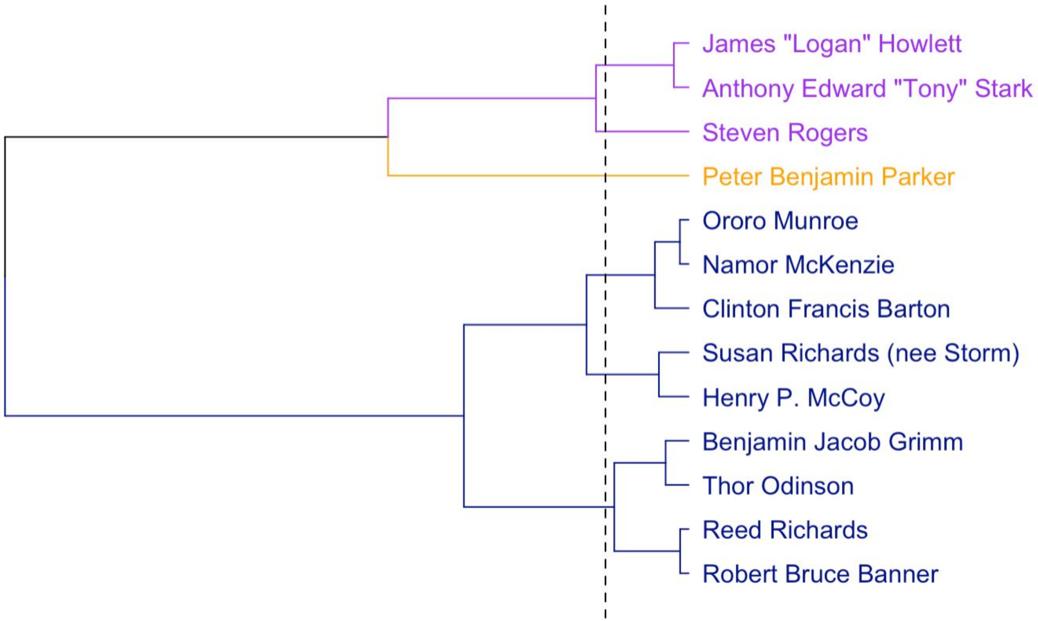
The clusters at a particular level of the hierarchy result from merging clusters at the next level (bottom up approach).

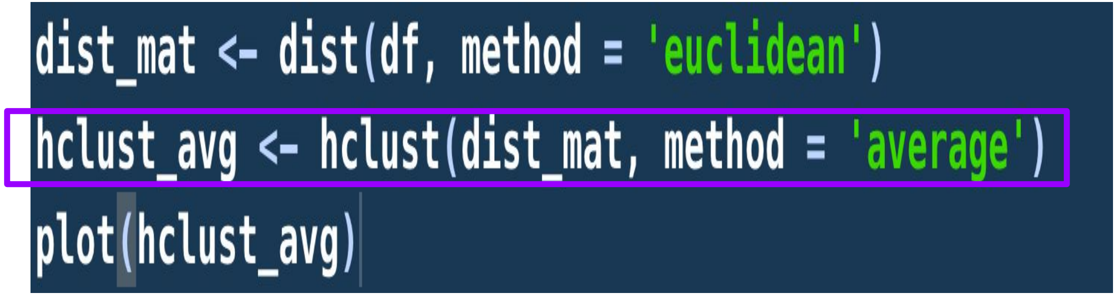
At each step we combine two clusters that contain the closest pair of elements not yet belonging to the same cluster as each other. (Gareth James et al.,385–399.)

Drawback: elements of the same cluster have small distances, but elements at opposite ends of a cluster may be much farther from each other than two elements of other clusters



Hierarchical clustering





```r
dist_mat <- dist(df, method = 'euclidean')
hclust_avg <- hclust(dist_mat, method = 'average')
plot(hclust_avg)
```

References

“Ch:10.3 Clustering Method.” An Introduction to Statistical Learning: with Applications in R, by Gareth James et al., Springer, 2017, pp. 385–399.

“Ch: 4.4: Linear Discriminant Analysis.” An Introduction to Statistical Learning: with Applications in R, by Gareth James et al., Springer, 2017, pp. 138–168.

“Ch 8: Tree-Based Methods.” An Introduction to Statistical Learning: with Applications in R, by Gareth James et al., Springer, 2017, pp. 303–332.

“Ch 15: Non-Model Based Classification &amp; Clustering .” Categorical Data Analysis, by Alan Agresti, Wiley, 2014.

Charles. “Discriminant Analysis Essentials in R.” STHDA, 11 Mar. 2018, www.sthda.com/english/articles/36-classification-methods-essentials/146-discriminant-analysis-essentials-in-r/.

Gao, Chao, et al. “Model-Based and Model-Free Machine Learning Techniques for Diagnostic Prediction and Classification of Clinical Outcomes in Parkinson's Disease.” Nature News, Nature Publishing Group, 8 May 2018, www.nature.com/articles/s41598-018-24783-4.

References

“Linear Discriminant Analysis.” Dr. Sebastian Raschka, 3 Aug. 2014, sebastianraschka.com/Articles/2014_python_lda.html.

Madeleine, et al. “StatQuest: Linear Discriminant Analysis (LDA), Clearly Explained.” StatQuest!!!, 10 July 2016, statquest.org/statquest-linear-discriminant-analysis-lda-clearly-explained/.

Support Vector Machines: The Linearly Separable Case. nlp.stanford.edu/IR-book/html/htmledition/support-vector-machines-the-linearly-separable-case-1.html.

Vidhya, and Analytics Vidhya. “Tree Based Algorithms : A Complete Tutorial from Scratch (in R &amp; Python).” Analytics Vidhya, 27 Mar. 2020, www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/.

“What Is a Dendrogram? How to Use Dendrograms.” Displayr, 20 Apr. 2020, www.displayr.com/what-is-dendrogram/.

THANK YOU!
ANY QUESTIONS?
makeameme.org