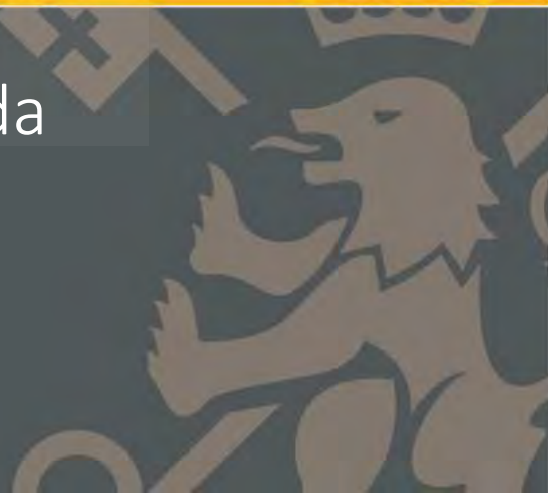
COMILLAS
UNIVERSIDAD PONTIFICIA
ICAI

ICAI – GITI/GITT

Classification
Estadística II

Eugenio Sánchez Úbeda

January 2024



comillas.edu

Bibliography

- G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor, *An Introduction to Statistical Learning with Applications in Python*, Springer, 2023 (https://www.statlearning.com)

1. Introduction
2. Model complexity vs. generalization error
3. Direct approach: Classification trees
4. Probabilistic approach: Linear Discriminant Analysis
5. Quiz
6. Real examples

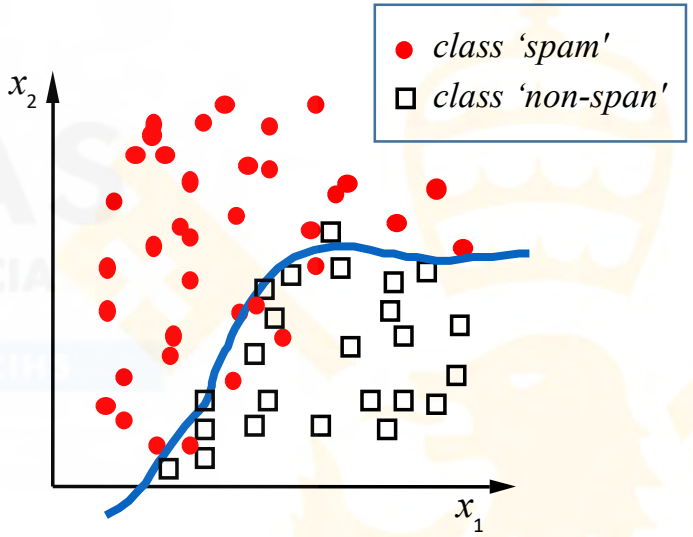
1

Introduction

Classification problems
Example

- **Email Spam Filtering** involves classifying an email as spam or non-spam based on the email contents
  - Output: spam or non-spam
  - Input variables or features: information obtained from the email

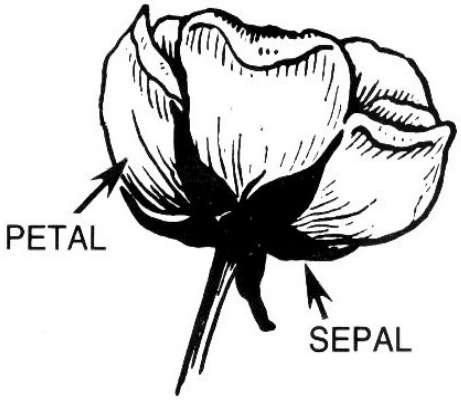
http://www.kdnuggets.com/

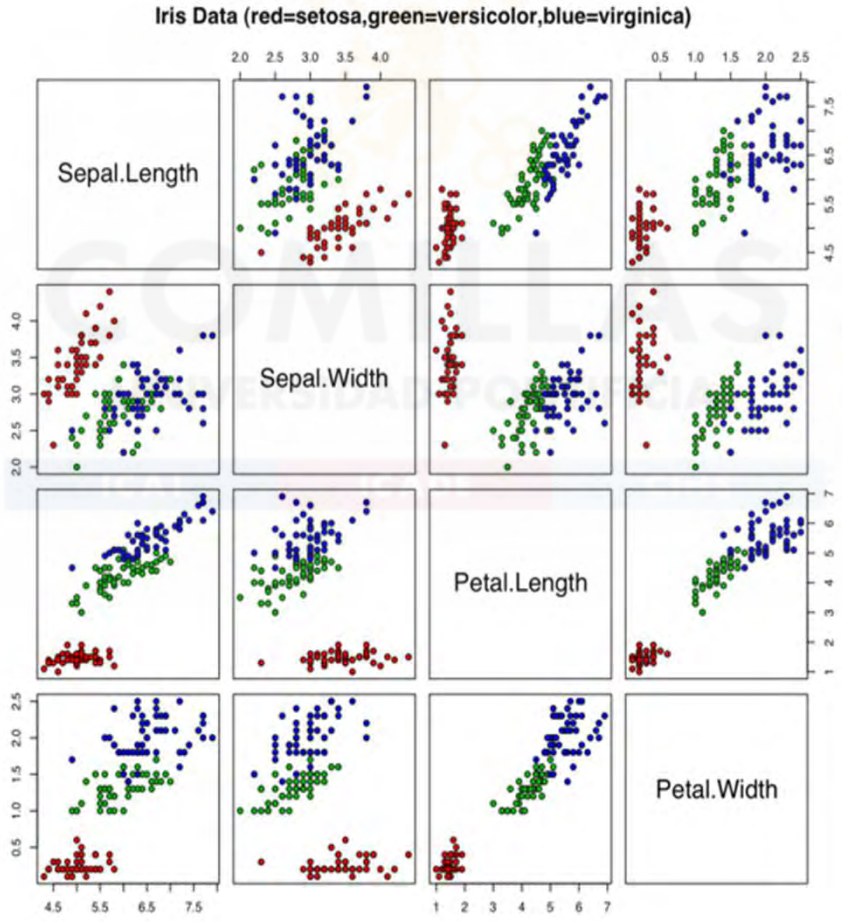


The filter estimates the output from the inputs (i.e., given an email, it is spam or not)

Classification problems
Iris data

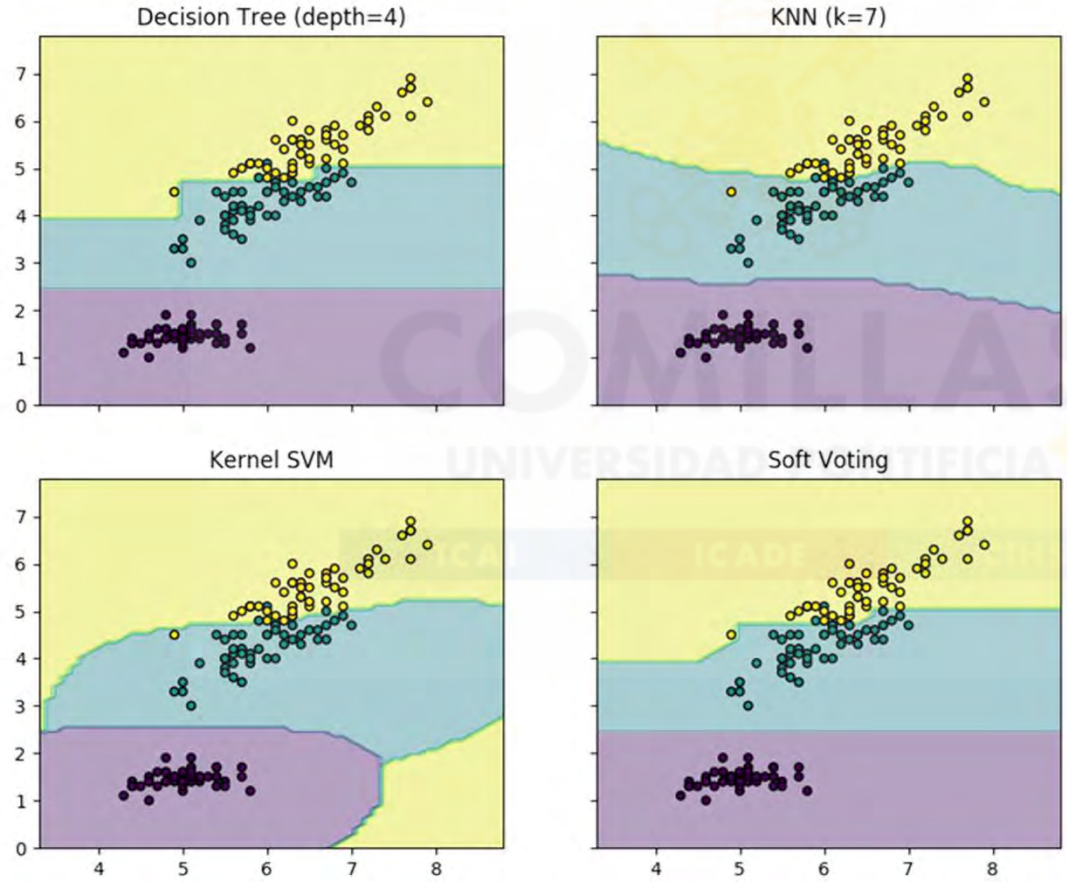
- 150 observations
  - 4 inputs: length and width of petals and sepals
  - Output: type of Iris (setosa, versicolor, virginica)





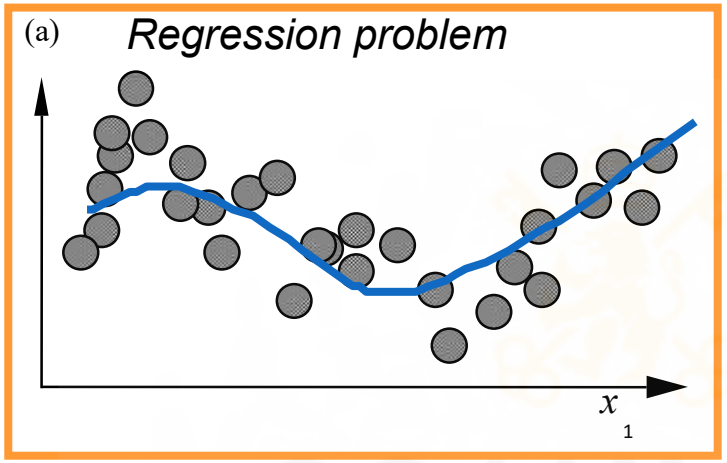
Classification problems
Iris data

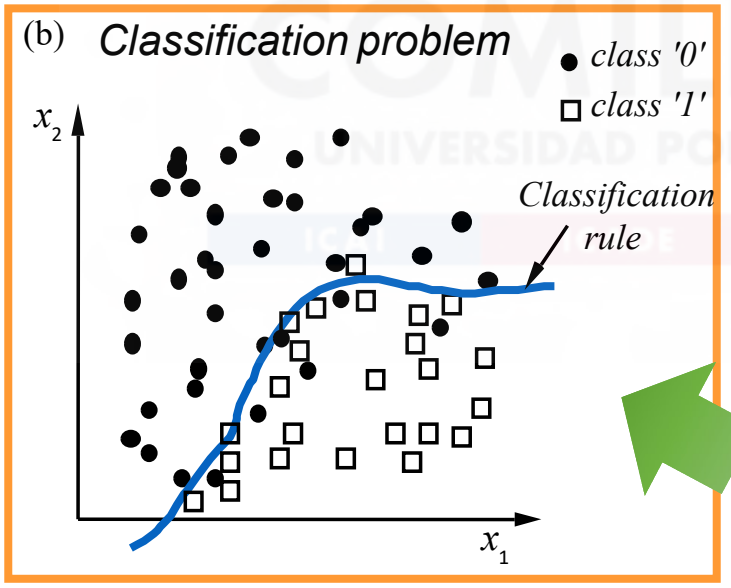
- Estimated output by four different classifiers (using two inputs)
  - For each pair of input values, the color represents the estimated output

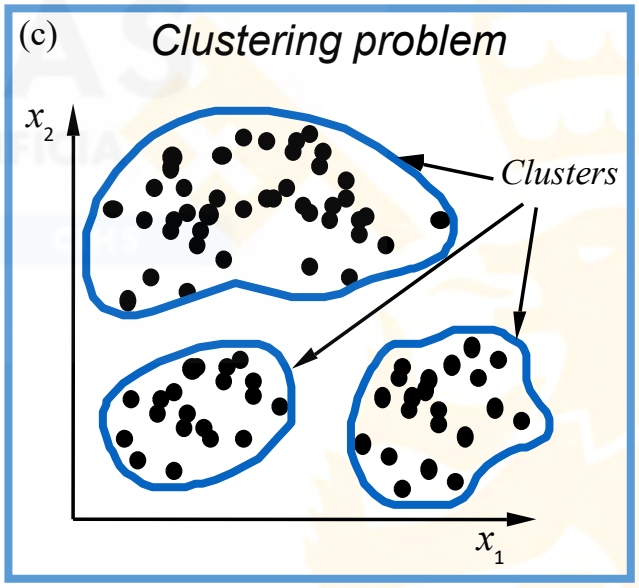


- There exist many classification techniques
- Accuracy, interpretability, and learning complexity of the estimation process are key for selecting the appropriate one

Main types of problems
The classification problem







Supervised learning

Unsupervised learning

Many of the concepts we have found in regression are transferred to the classification setting with only some modifications because *the output is no longer numerical*.

Classification Approach

- The theoretical model consists of two main terms:

Deterministic Term

Random Term (classification error)

$$Y = f(X) + \epsilon$$

The qualitative output variable Y *can take* K possible distinct and unordered values (K different classes or categories)

Usually, in classification, the deterministic term has no mathematical compact form; it includes some logical classification rule

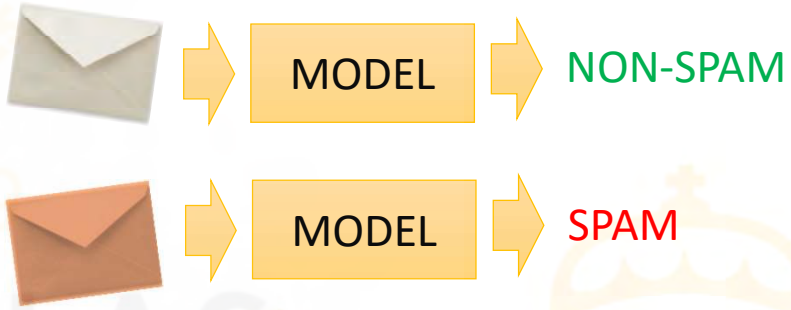
Classification
Main approaches (idea)

$$Y = f(X) + \epsilon$$

- **Direct approach**
  - Focus on the **conditional expected value** of the output (the average class)
  - Estimates directly the output value

Classification trees

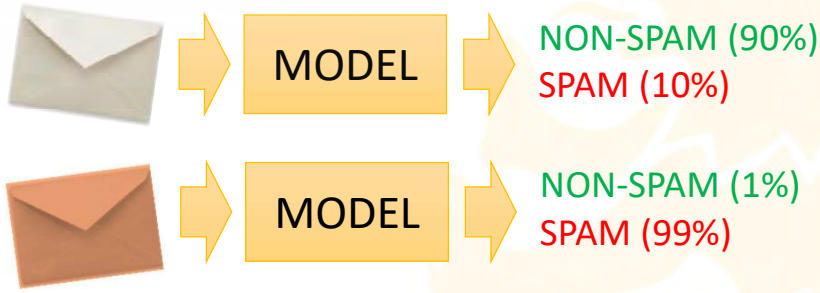
$$\mathrm{E}(Y|X = x)$$



- **Probabilistic approach**
  - Focus on the **conditional probability of each class**
  - Estimate the K probabilities to provide an **enhanced output**

Linear discriminant

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



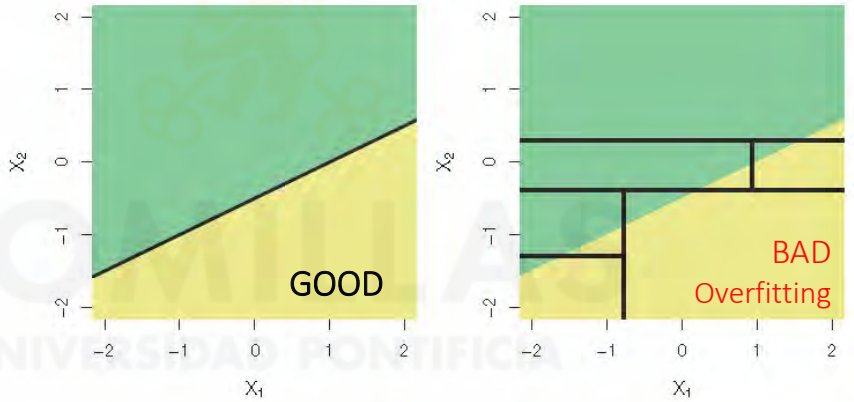
Classification
Classification tree vs. linear discriminant

- Two bidimensional classification illustrative examples

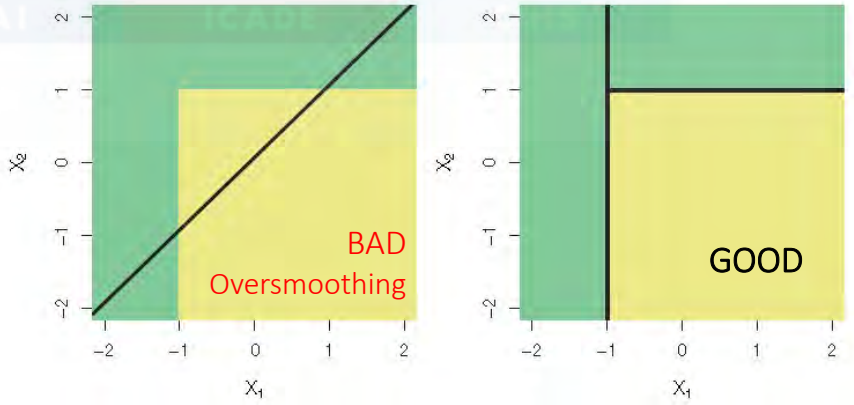
Classical
Linear discriminant

Classification tree

PROBLEM A: the true decision boundary is linear



PROBLEM B: The true decision boundary is nonlinear



Classification problem
Classification vs. regression

- The main goal is to estimate, from the input values, the output value (as in regression)
- Now the output is qualitative (categorical)
  - Finite set of possible values
  - Unordered values

New ways of assessing the model accuracy

Classification
Assessing model accuracy

- The most common approach for quantifying the accuracy of the classifier is the error rate, the proportion of mistakes that are made if we apply our estimate to the observations
  - 1 means 100% of misclassification
  - 0 means perfect classification

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

Dataset for computing the error rate

$(x_1, y_1), \ldots, (x_n, y_n)$

Indicator variable

I(true)=1 I(false) = 0

$$y_i = \hat{y}_i \rightarrow I(y_i \neq \hat{y}_i) = 0$$

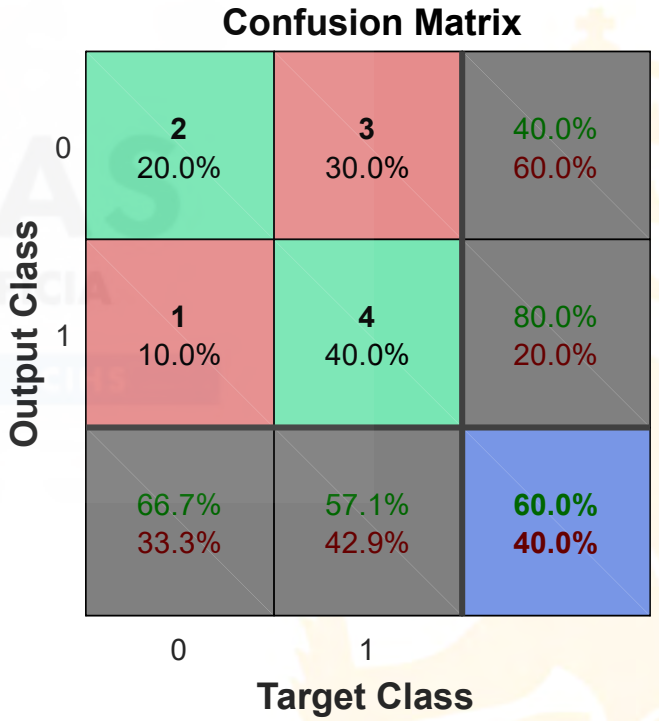
Classification
Assessing model accuracy

- The classification confusion matrix allows us to understand how the classifier performed in each class, the types of mistakes
- Example

```
y    = [1 1 1 1 1 1 1 0 0 0];  %true, target
yest = [1 0 1 1 0 0 1 0 0 1];  %estimated
plotconfusion(y,yest);
```

y	yest
1	1
1	0
1	1
1	1
1	0
1	0
1	1
0	0
0	0
0	1

Elements on the diagonal of the matrix represent observations whose output were correctly predicted, while off-diagonal elements represent cases that were misclassified



Overall error rate = 40.0%

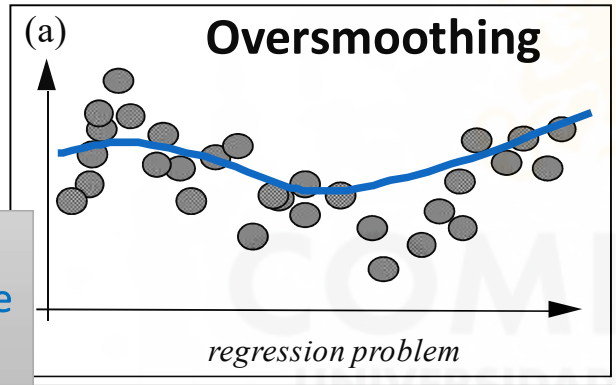
2

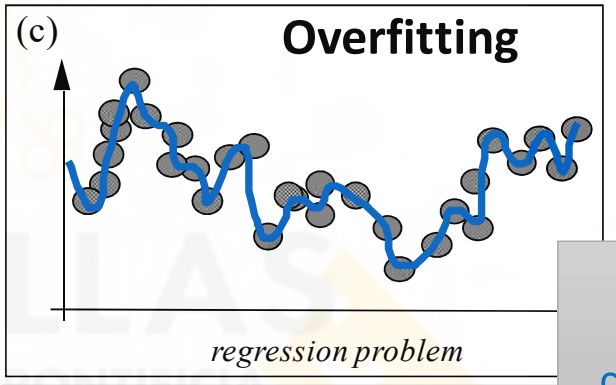
Model complexity vs. generalization error

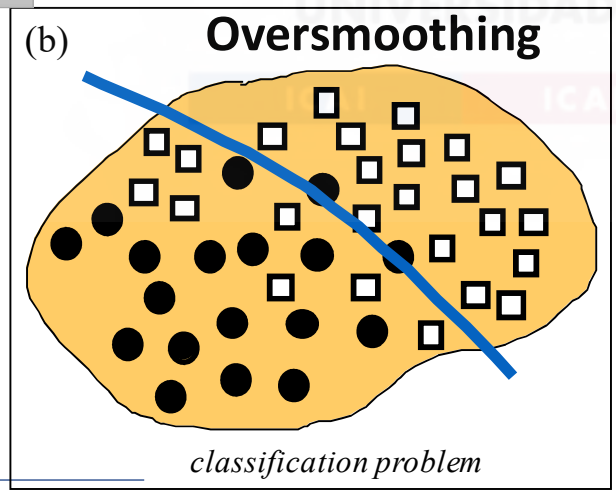
Main difficulties
Overfitting and oversmoothing

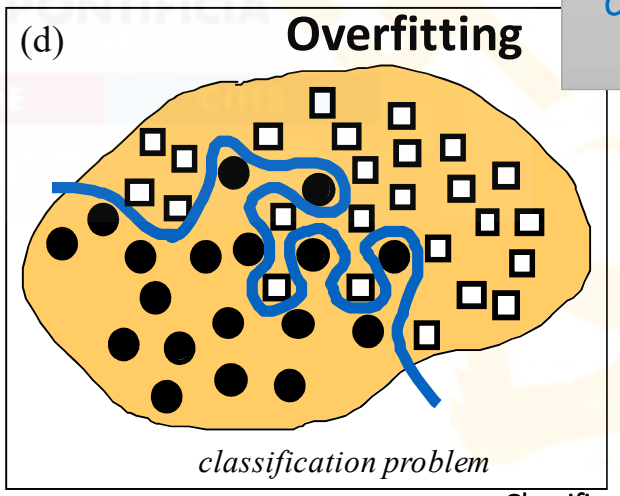
- More complex models (such as classification trees) can lead to a phenomenon known as overfitting the data, which essentially means they follow the errors, or noise, too closely

Oversmoothing is typical if very simple models are used





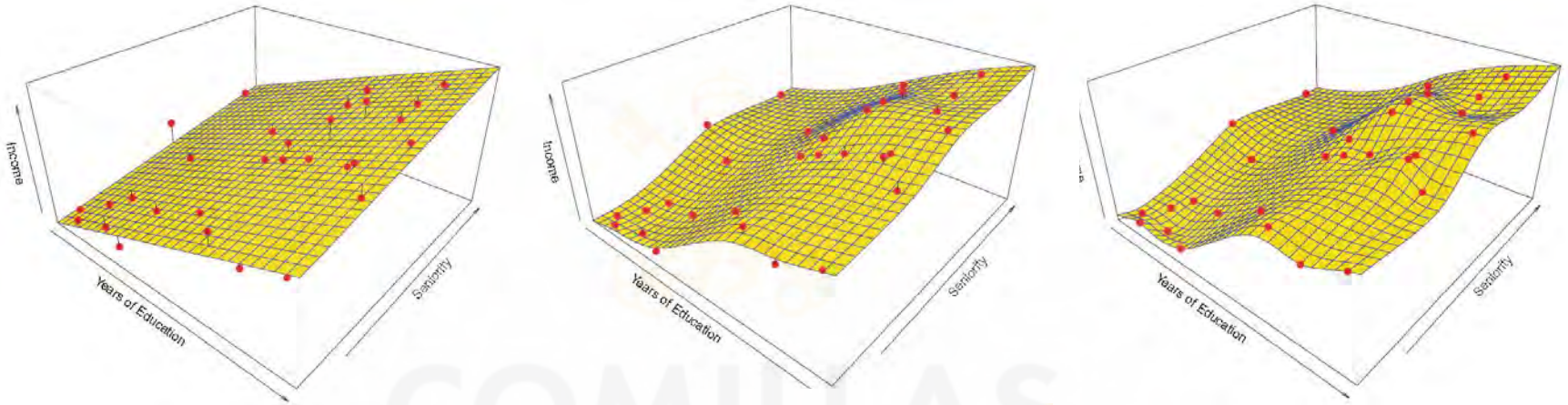




Overfitting is typical if very complex models are used

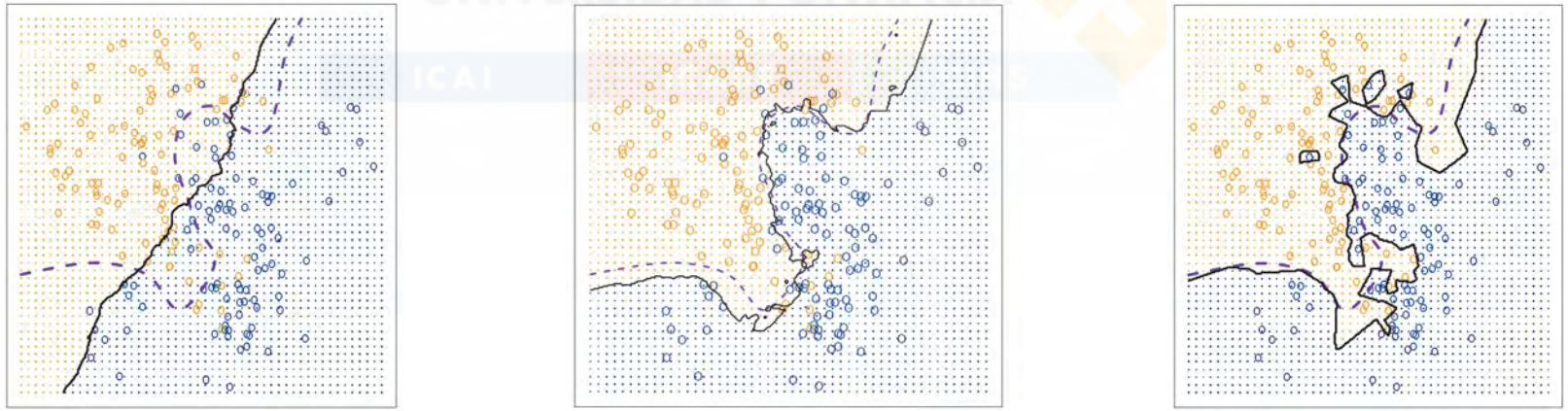
Main difficulties
Overfitting and oversmoothing

- Example (bidimensional regression problem)



Complexity increases

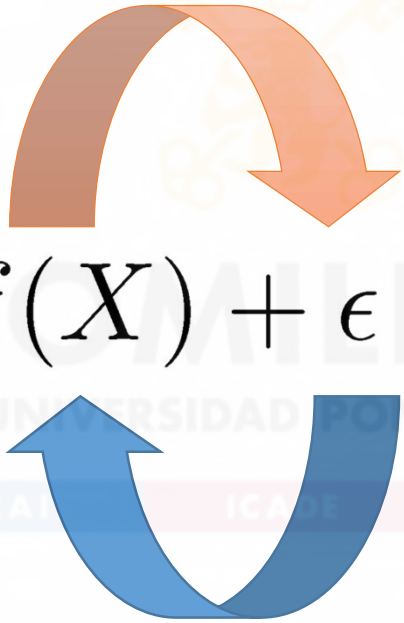
- Example (classification problem)



Complexity increases

Main difficulties
Overfitting and oversmoothing

- Trade-off between the deterministic and the random component of the model



$$Y = f(X) + \epsilon$$

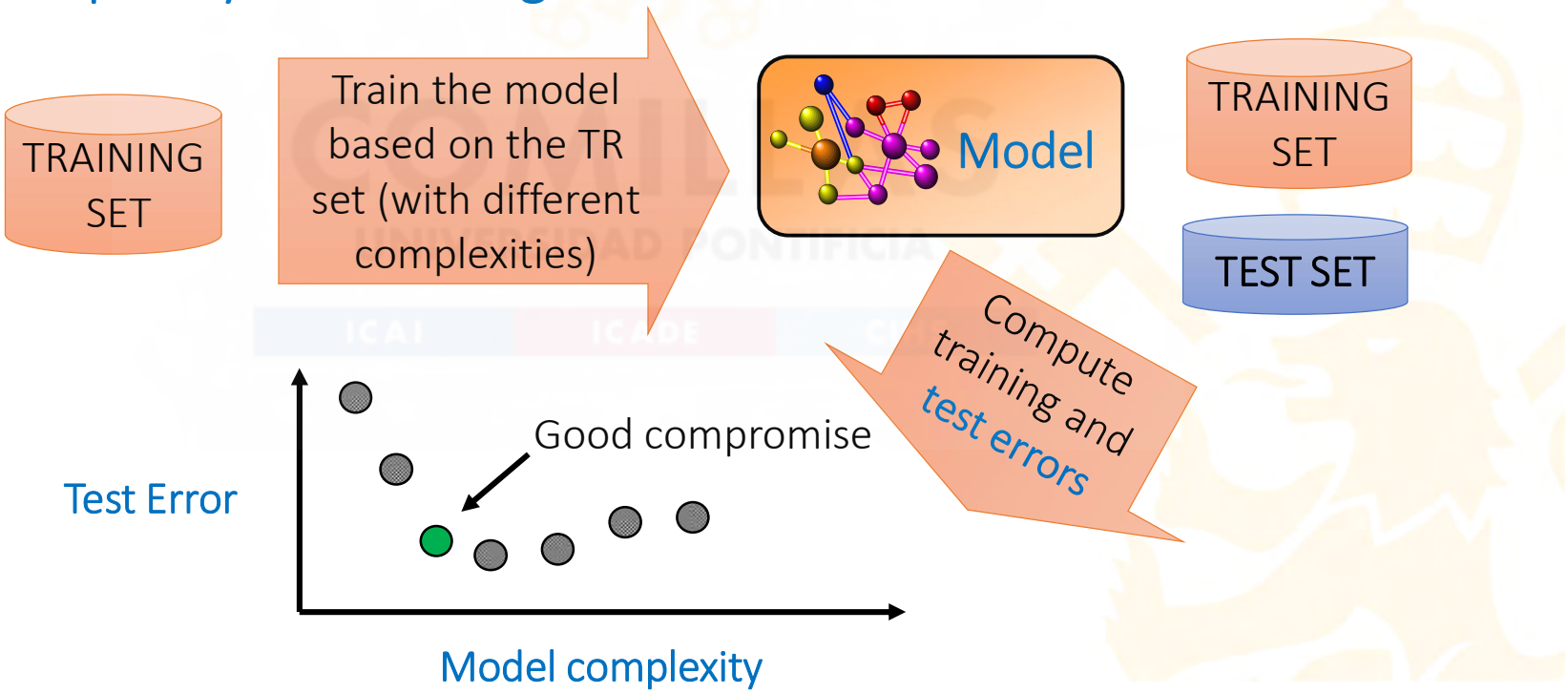
In the over-smoothing case, part of the deterministic component is considered noise

In the overfitting case, part of the noise is modeled as deterministic

It is a question of estimating the right complexity of the model

Model complexity
Training and test sets

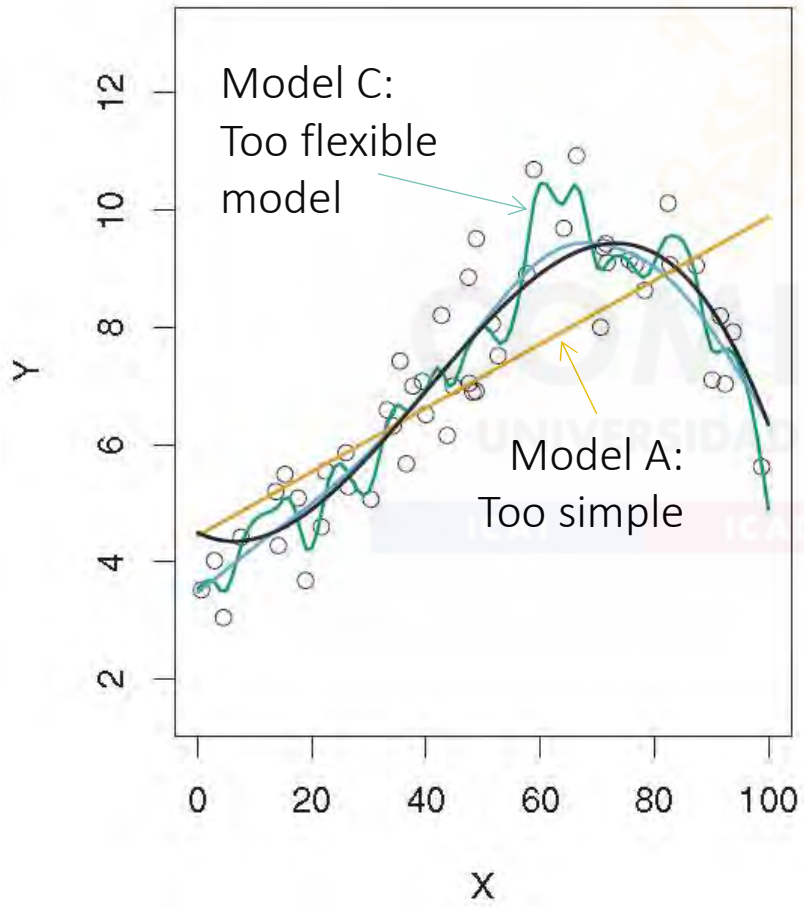
- The test set is used to estimate the future prediction error, consisting of unseen observations not used to train the statistical learning method
- The test set allows fixing the trade-off between model complexity and training error

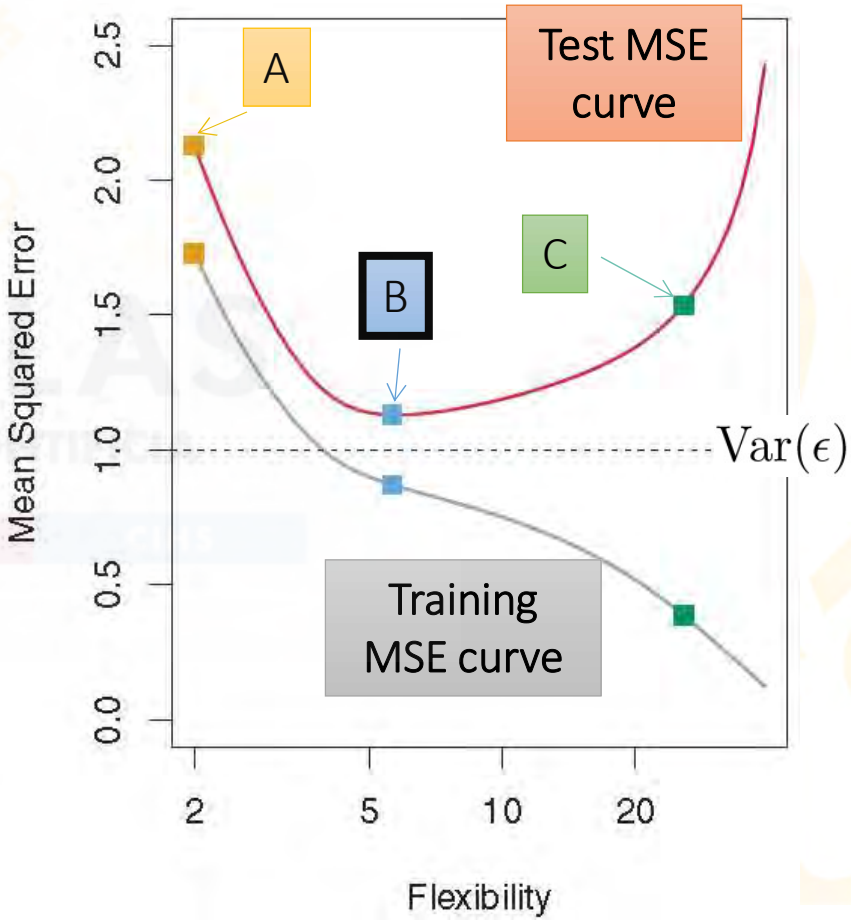


Model complexity
Training and test sets

- Example (regression problem)

The U-shape in the test MSE curve holds regardless of the data set and the statistical method being used



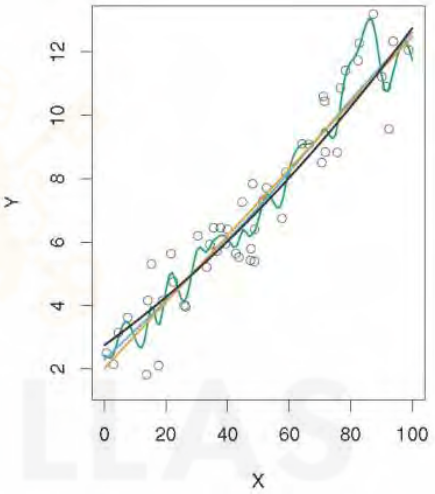


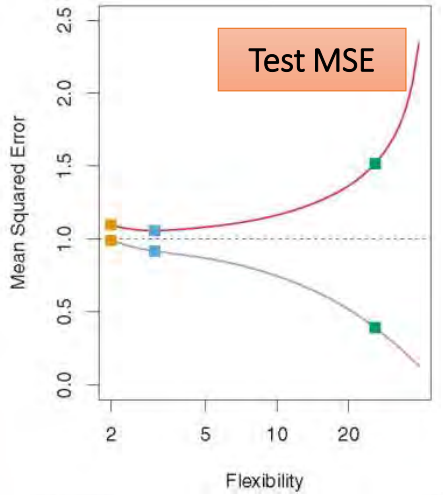
Model complexity
Training and test sets

U-shape in the test MSE holds

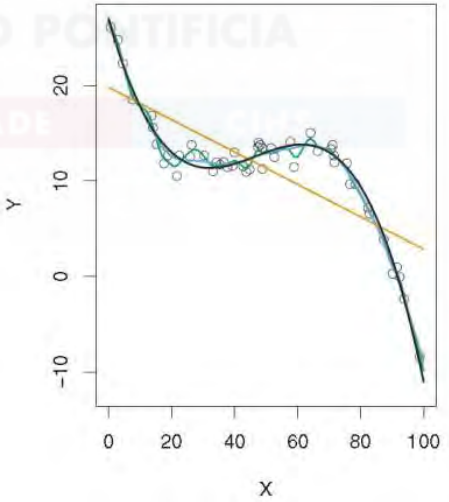
- Example (regression problem)

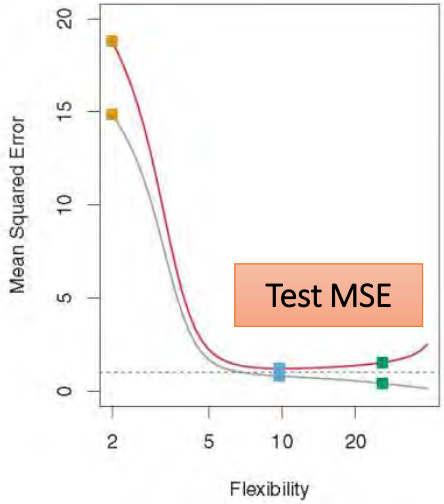
The training MSE decreases monotonically. However, because the truth is close to linear, the test MSE only decreases slightly before increasing again





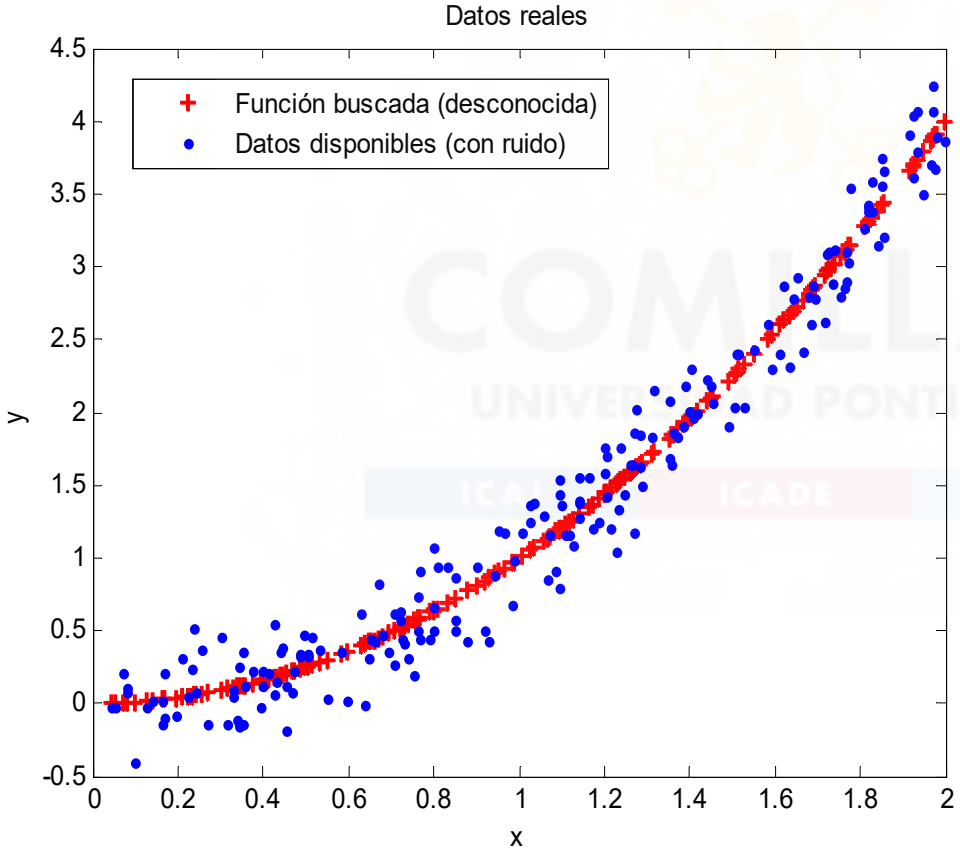
The training and test MSE curves still exhibit the same general patterns, but now there is a rapid decrease in both curves before the test MSE starts to increase slowly





Model complexity
Training and test sets

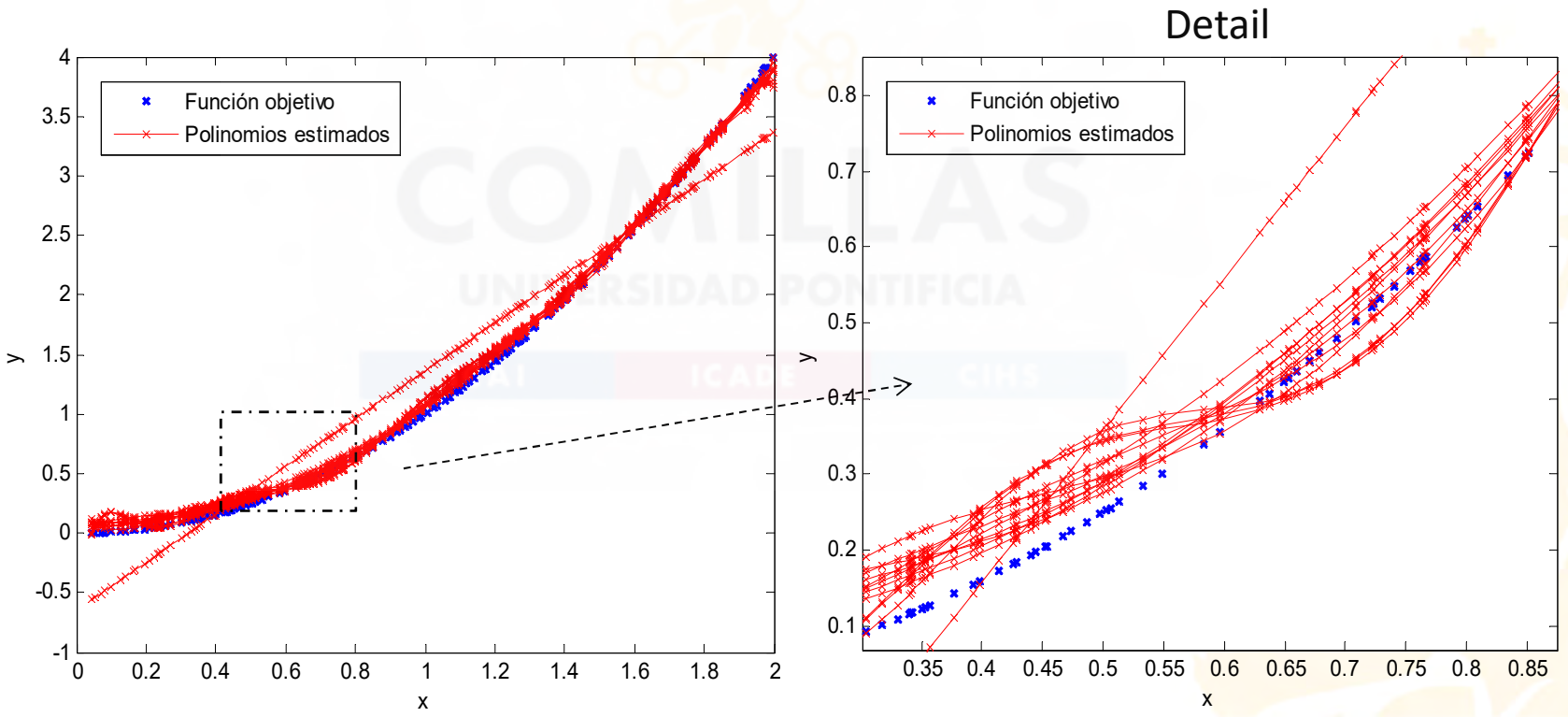
- Regression example
  - Estimate the polynomial (complexity and parameters) from the available dataset



- 200 observations
- There is noise

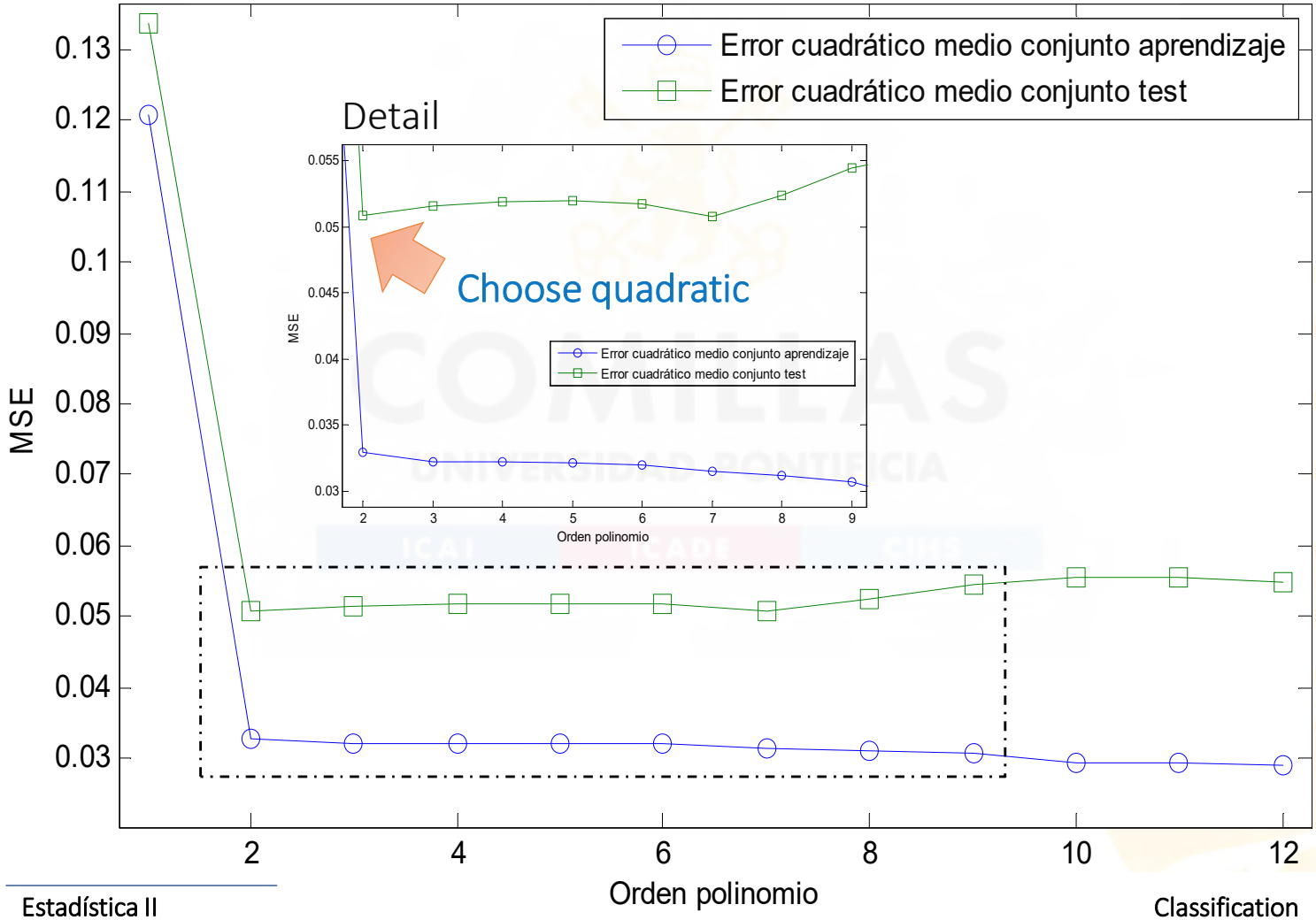
Model complexity
Training and test sets

- Fit 12 polynomials with increasing complexity between 1 and 12
- Use 80% of the data set as training data and 20% for testing



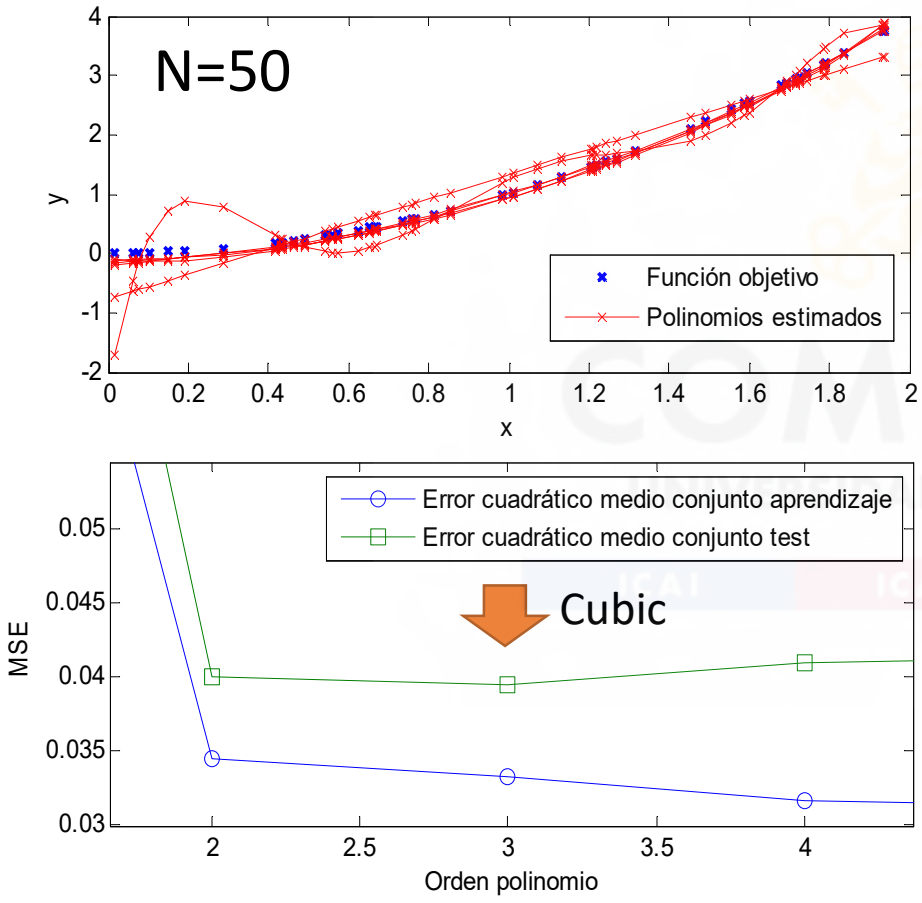
Model complexity
Training and test sets

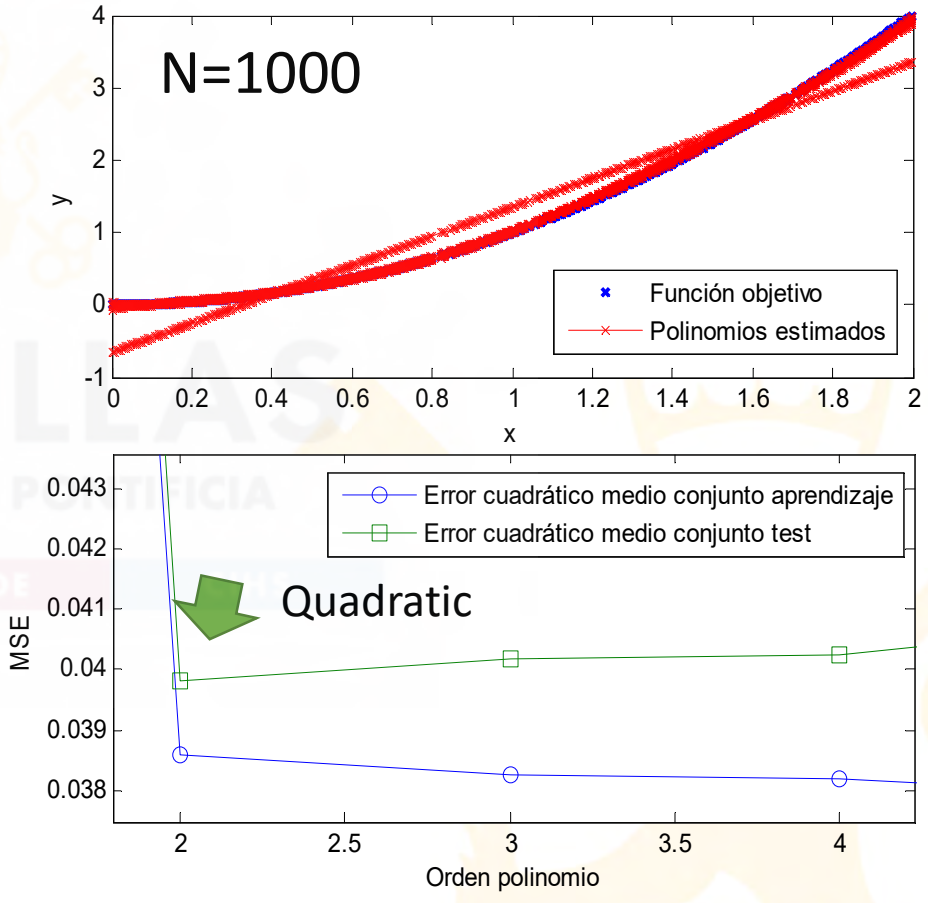
- Trade-off between test error and model complexity



Model complexity
Training and test sets

- More data allows better estimation of the complexity





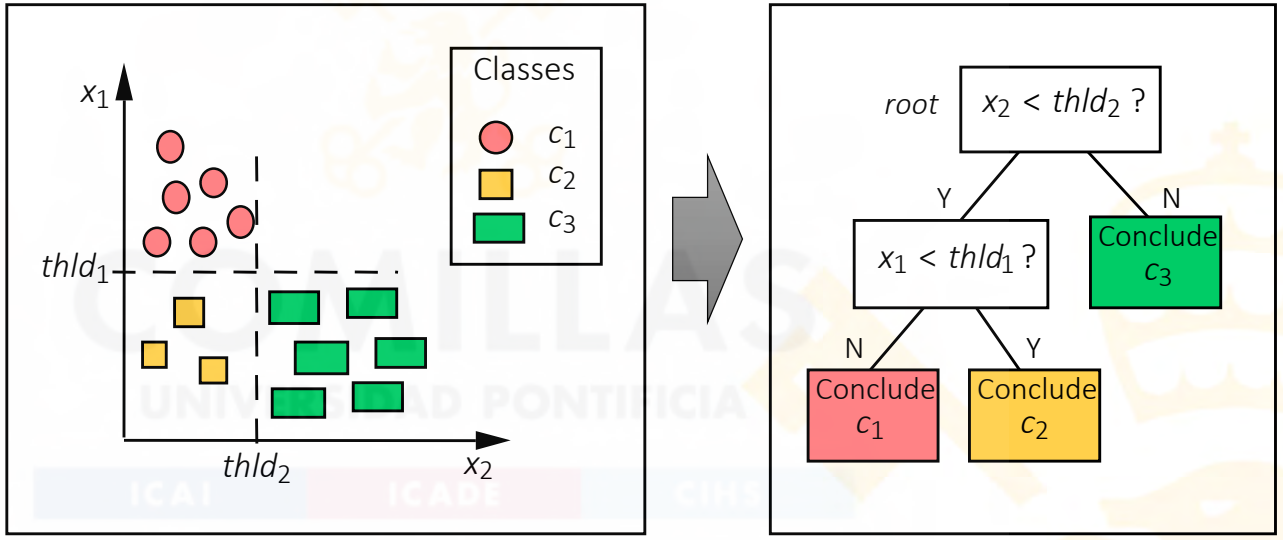
1. Introduction
2. Model complexity vs. generalization error
3. Direct approach: Classification trees
4. Probabilistic approach: Linear Discriminant Analysis
5. Quiz
6. Real examples

3

Direct approach: Classification trees

Classification trees
Overview

- Characteristics:
  - Classify classes of the output by splitting the input space
  - The resulting white-box model is hierarchical
- Example:

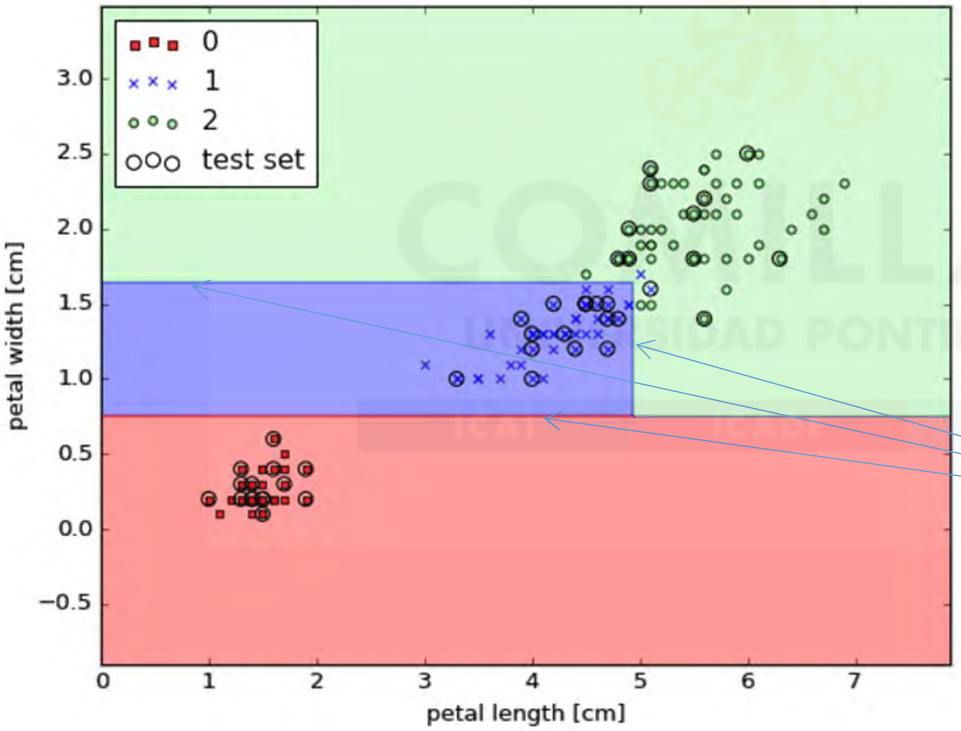


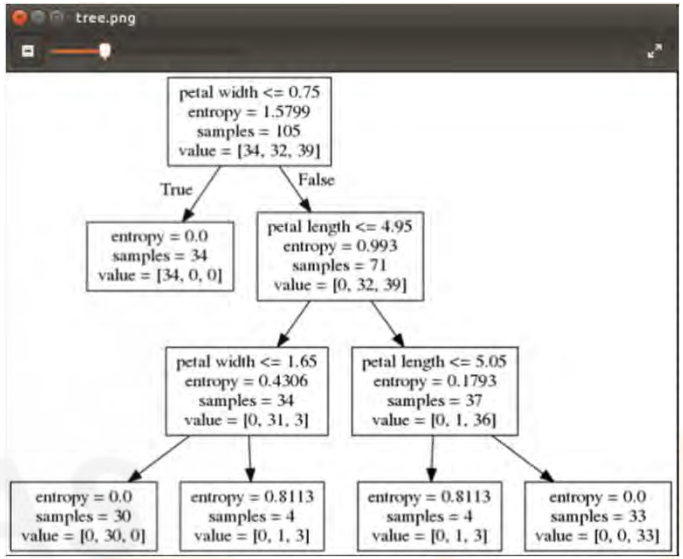
- Types of nodes: test and terminal
- Types of separators:
  - For categorical input variables: Value of X?
  - For continuous input variables: Value of $X <$ threshold?

- Nonlinear models
- Universal approximators

Classification trees
Partition of the input space

- Iris problem
  - The hierarchy of the tree imposes the input space partitioning





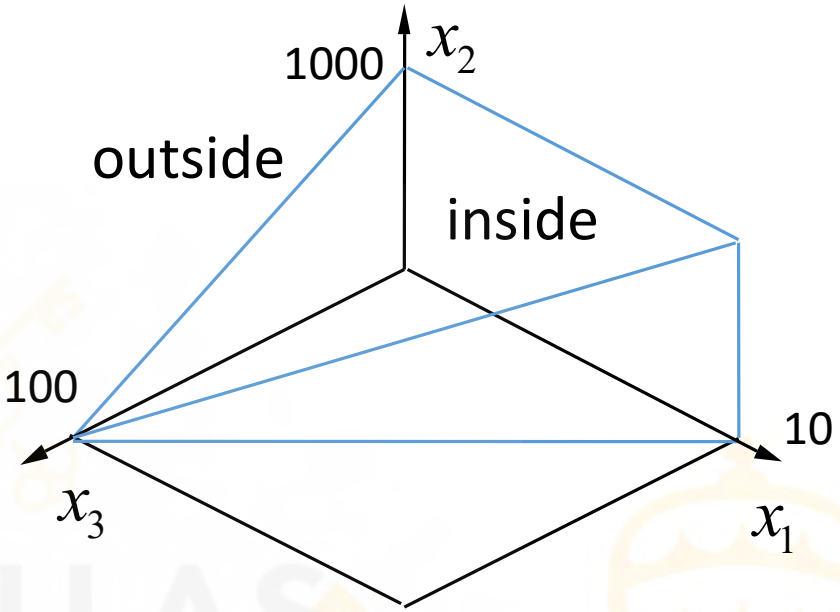
Splits are orthogonal to the axes

Classification trees
Illustrative example

- 2 classes: inside/outside
- 4 inputs (attributes)

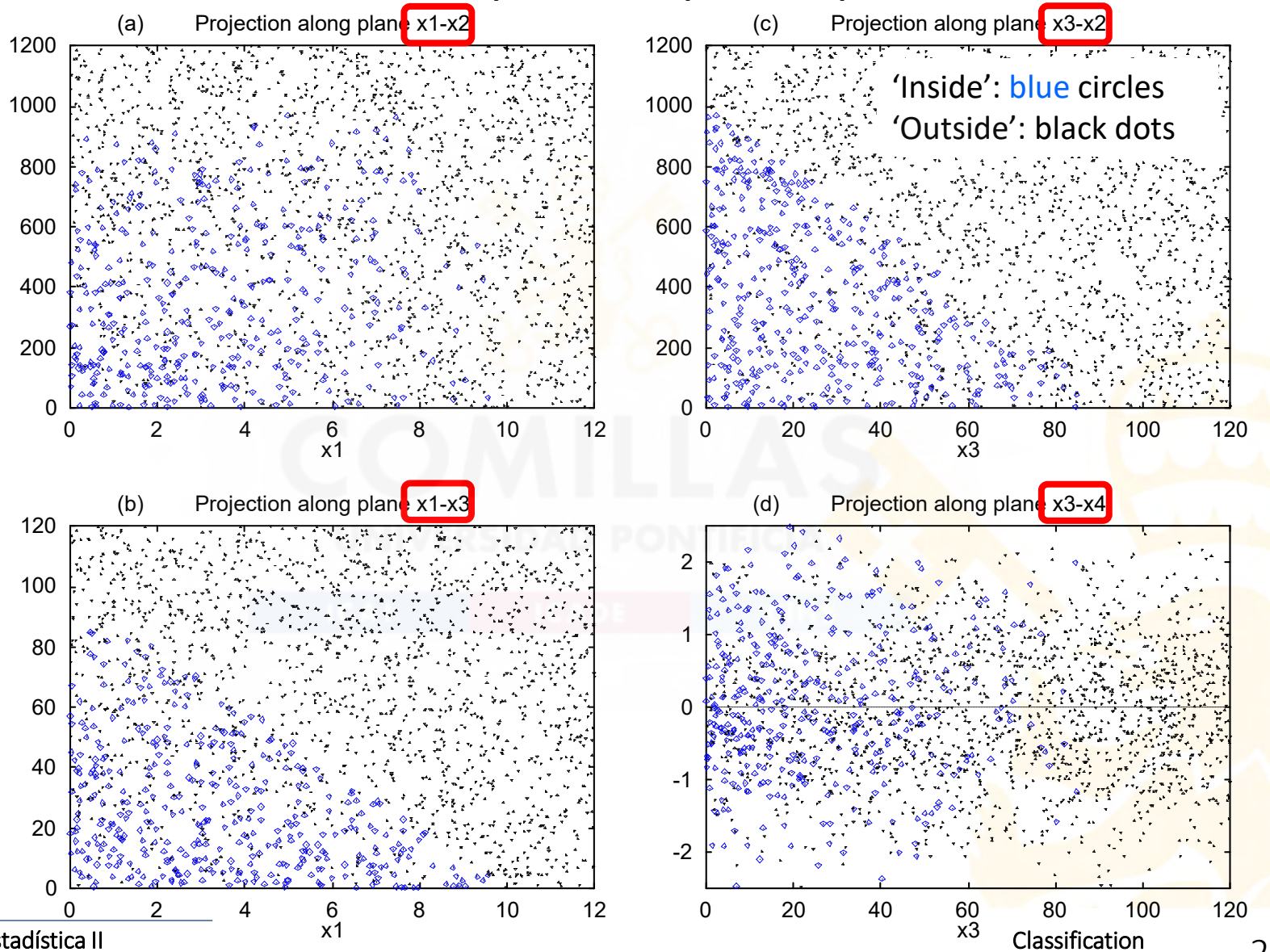
$$(x_1, x_2, x_3, x_4)$$

The fourth input is noise



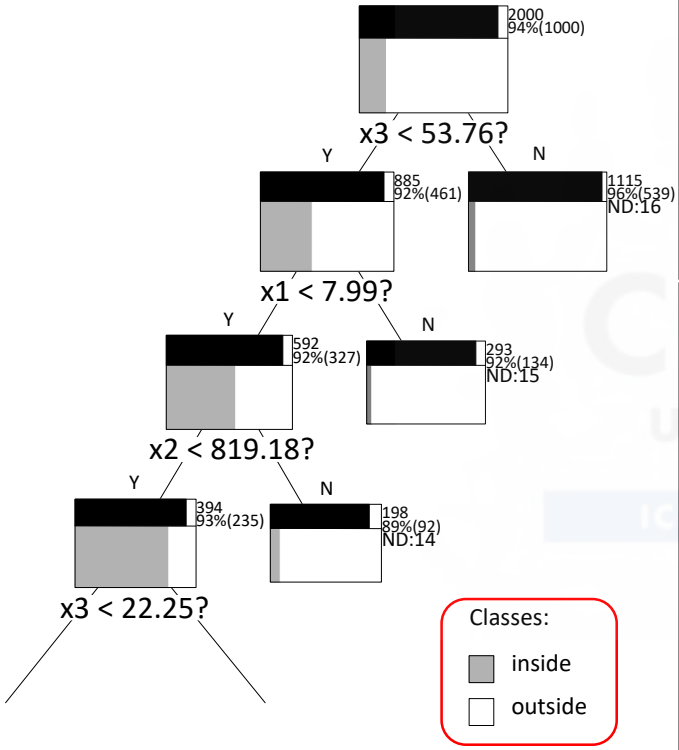

```
if (x3>100 OR x3<0)
    y = 0;  /* outside */
else{
  if ((x3<10*(10-x1)) AND (x3<(1000-x2)/10))
    y = 1 ; /* inside  */
  else
    y = 0 ; /* outside */
}
```

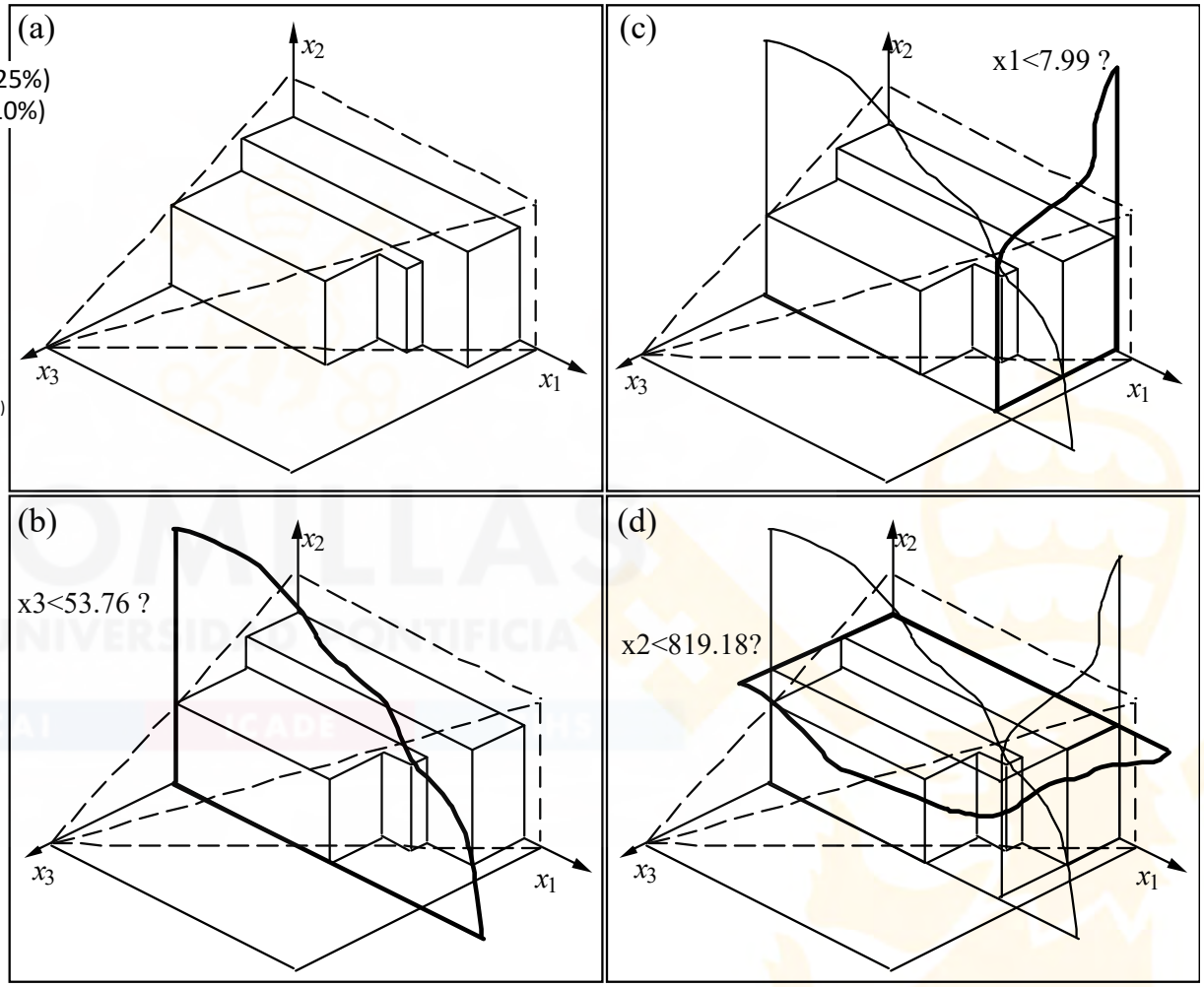
Classification trees
Illustrative example: input space



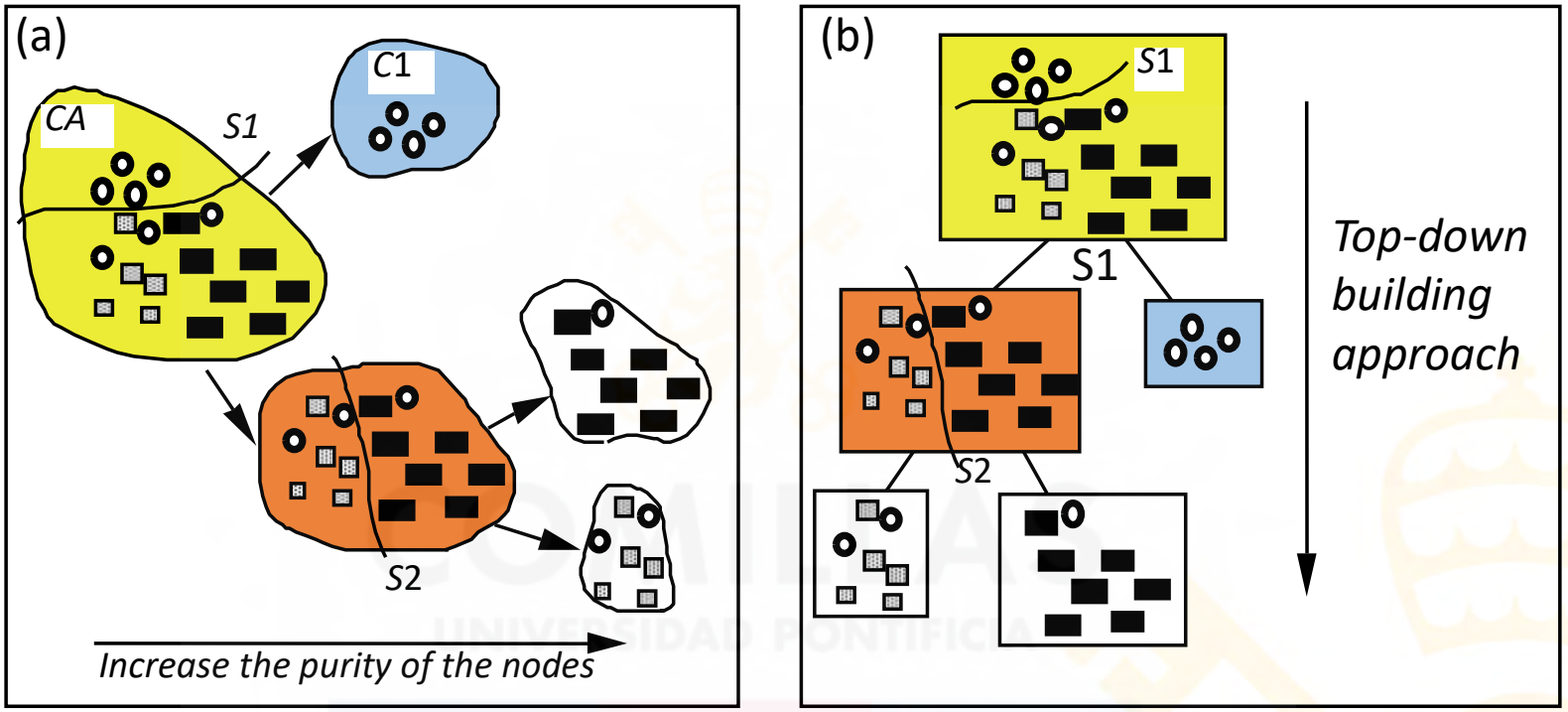
Classification trees
Illustrative example: splits

Learning set: data.tr N=2000 Correct_classif.= 1905 (95.25%)
Test set: data.tst N=1000 Correct_classif.= 941 (94.10%)
Algorithm: ID3 Hmin: 0.52 Number_of_nodes= 17





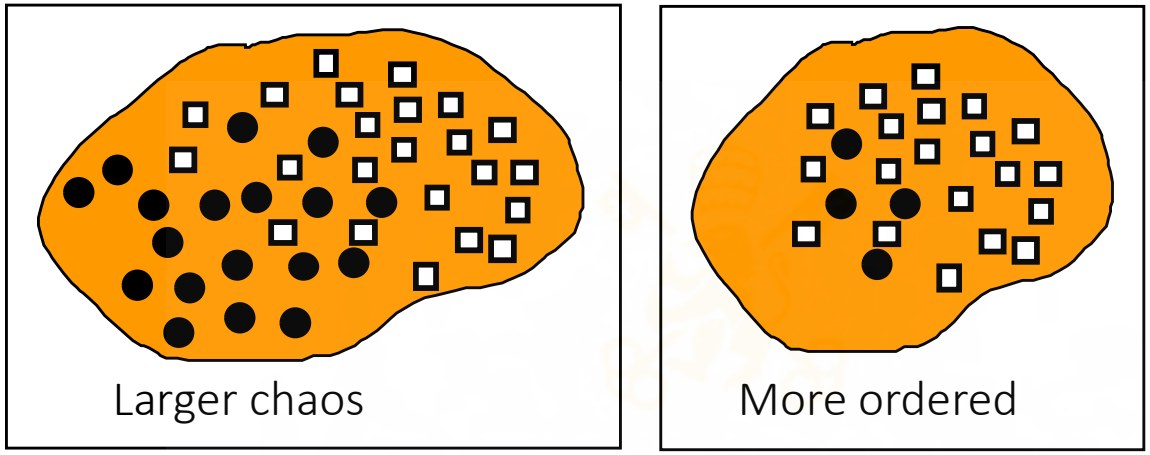
Classification trees
Growing learning algorithm



- Idea:
  - Split the input space recursively until the terminal nodes are pure enough

How do we assess the purity of a set?

Classification trees
Diversity



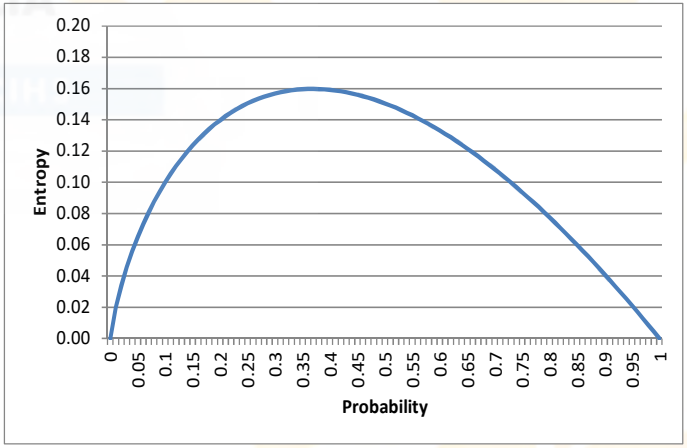
Chaos

Mixture of classes

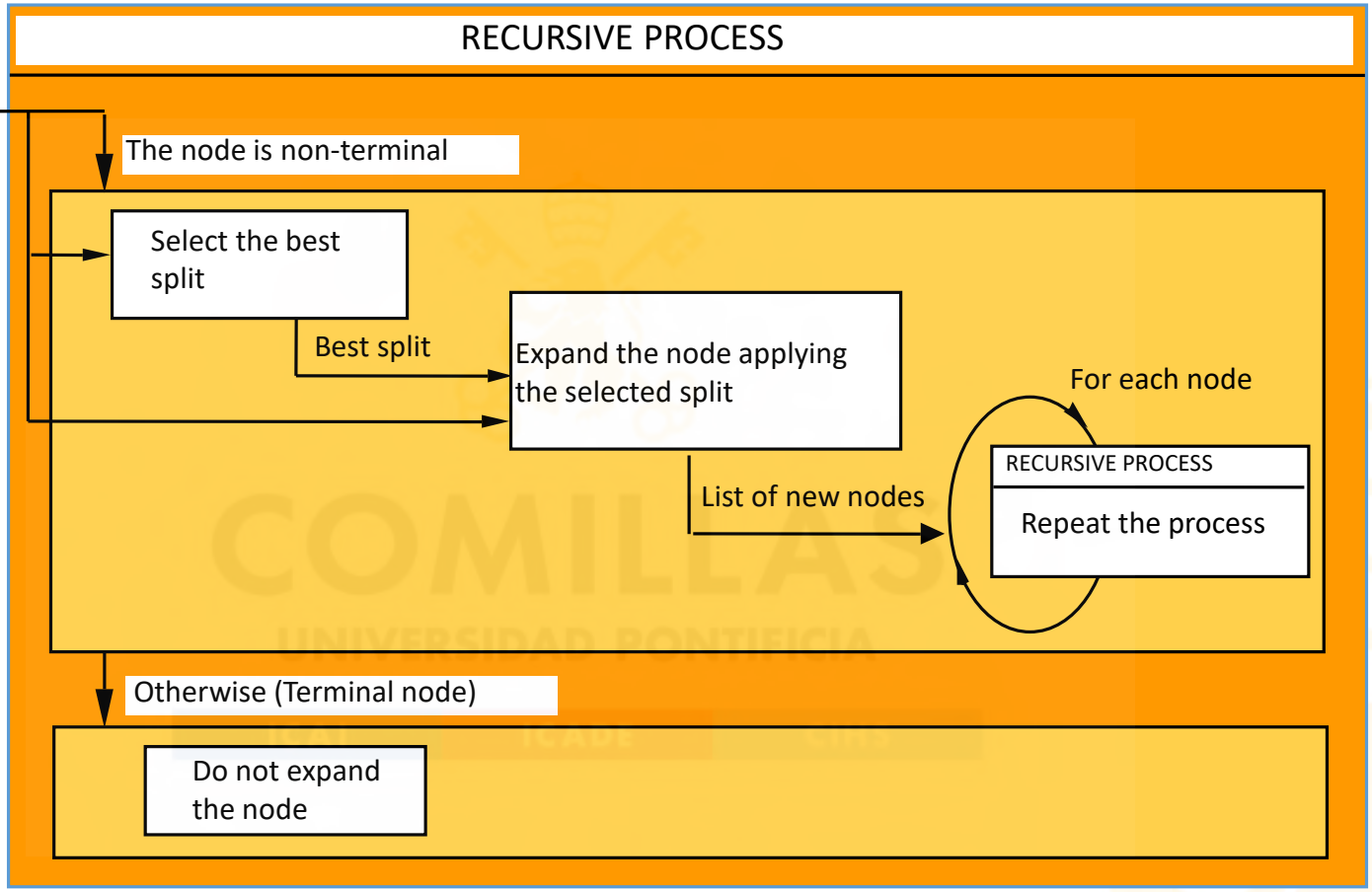
- Entropy: a standard way of assessing the chaos in the data set

$$H\big(LS(n)\big) = -\sum_{i=1,N_C} p(n,c_i)\log_2[p(n,c_i)]$$

Proportion of points belonging to class c_i



Classification trees
Growing learning algorithm

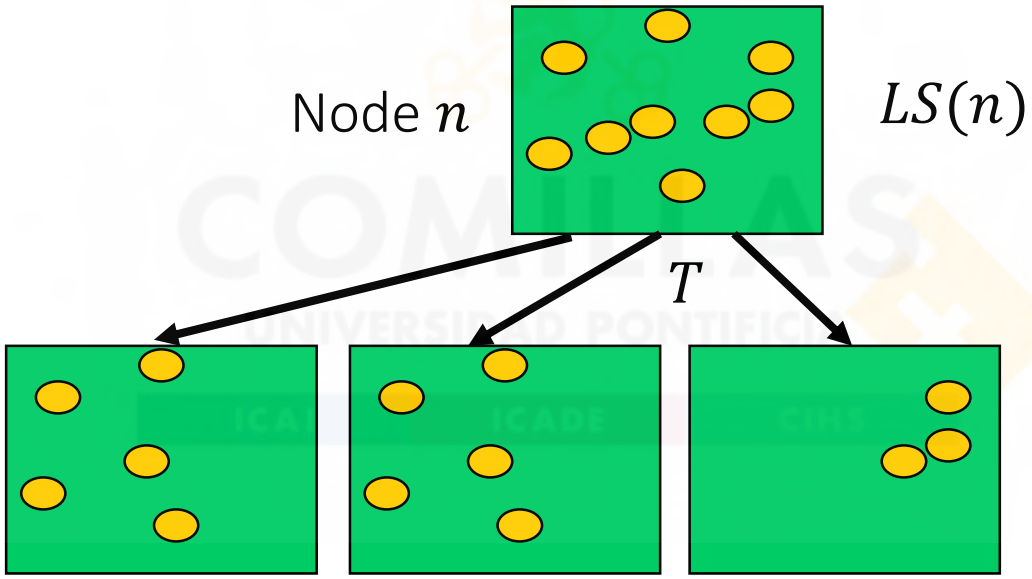


- Key points
  - Splitting criterion: selection of the best split
  - Stopping criterion

Classification trees
Growing Learning algorithm: Splitting criterion

- Select the best split T that produces the largest decrease of the chaos in the $LS(n)$

$$Index(n,T) = H\big(LS(n)\big) - \sum_{s=1,N_S} p(n_s)H\big(LS(n_s)\big)$$



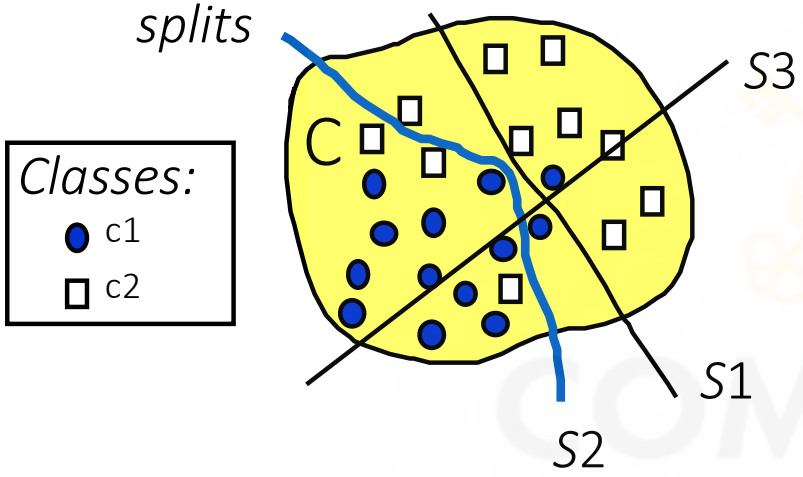
Types of splits (T):

X is categorical: T = value X?

X is continuous: T = value X < threshold?

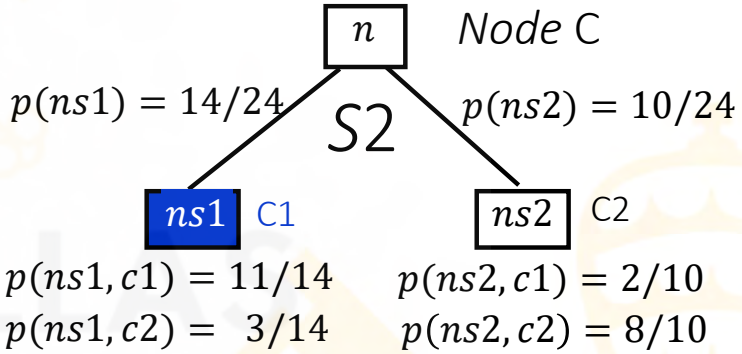
Classification trees
Growing learning algorithm: Splitting criterion

Data set and candidate splits



24 observations (13 of class c1 and 11 of class c2)

Evaluation of split S2



$$I_{ID3}(n,T) = H\big(LS(n)\big) - \sum_{s=1,N_S} p(n_s) H\big(LS(n_s)\big)$$

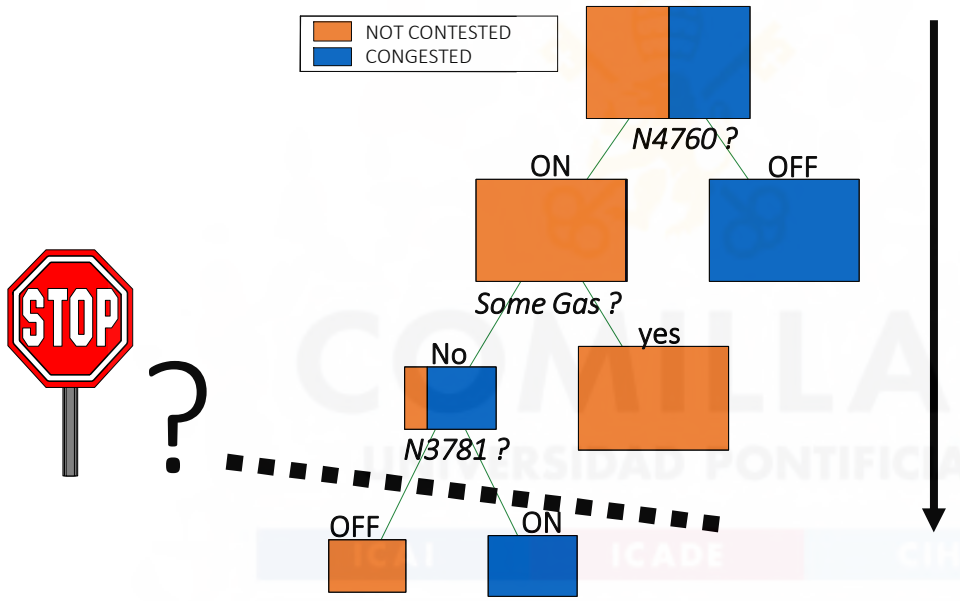
Entropy of the original data set

Entropy of the split data set

$$H\big(LS(n)\big) = -\sum_{i=1,N_C} p(n,c_i)\log_2[p(n,c_i)]$$

Classification trees
Growing learning algorithm: Stopping criterion

- Controls the number of nodes (related to the complexity of the tree)
- There exist different stopping criteria (more or less complex)



Growing stage

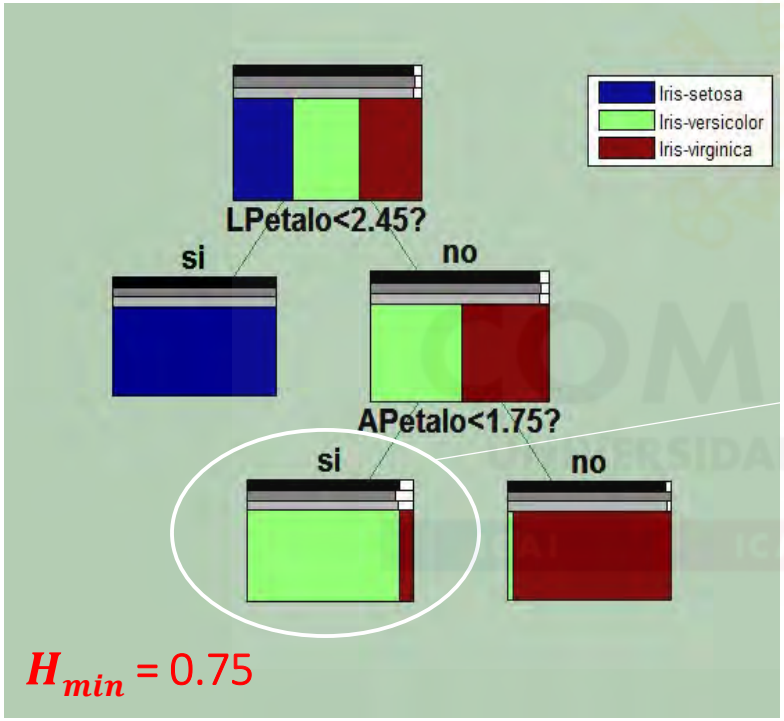
$$H(LS(n)) \leq H_{min}$$

H_{min}: Minimum entropy

- Stopping criterion based on entropy:
  - Stop splitting the node n if the entropy is small enough

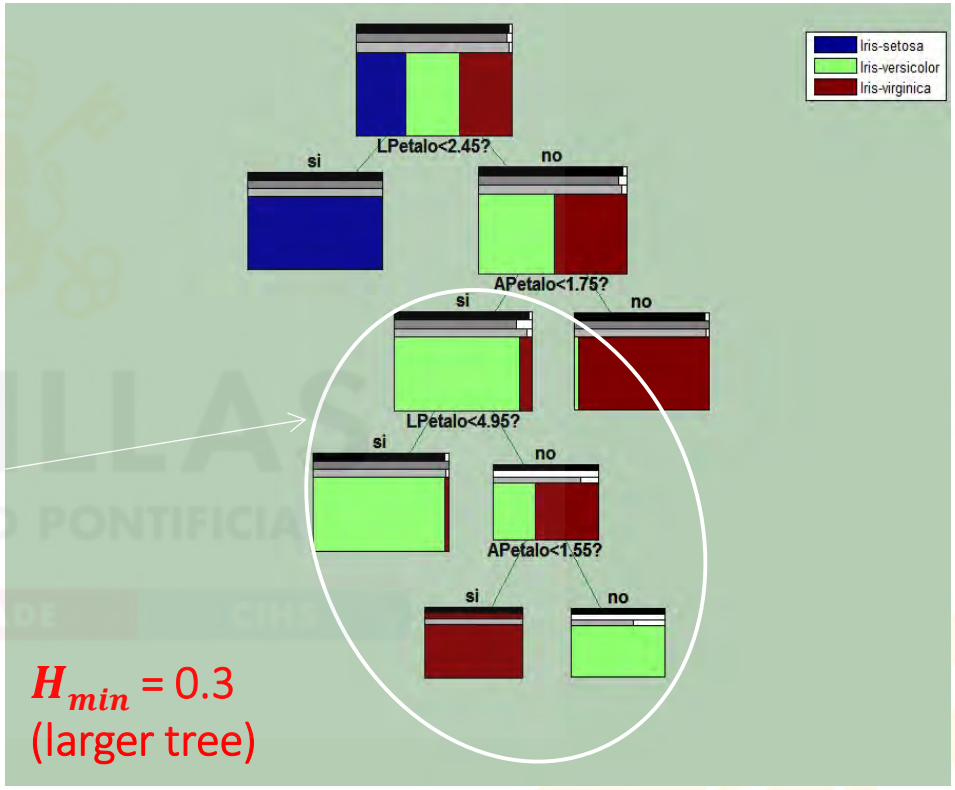
Classification trees
Real problem: Iris classification

96% of training observations are correctly classified
(97% of the test set)



H_{min} = 0.75

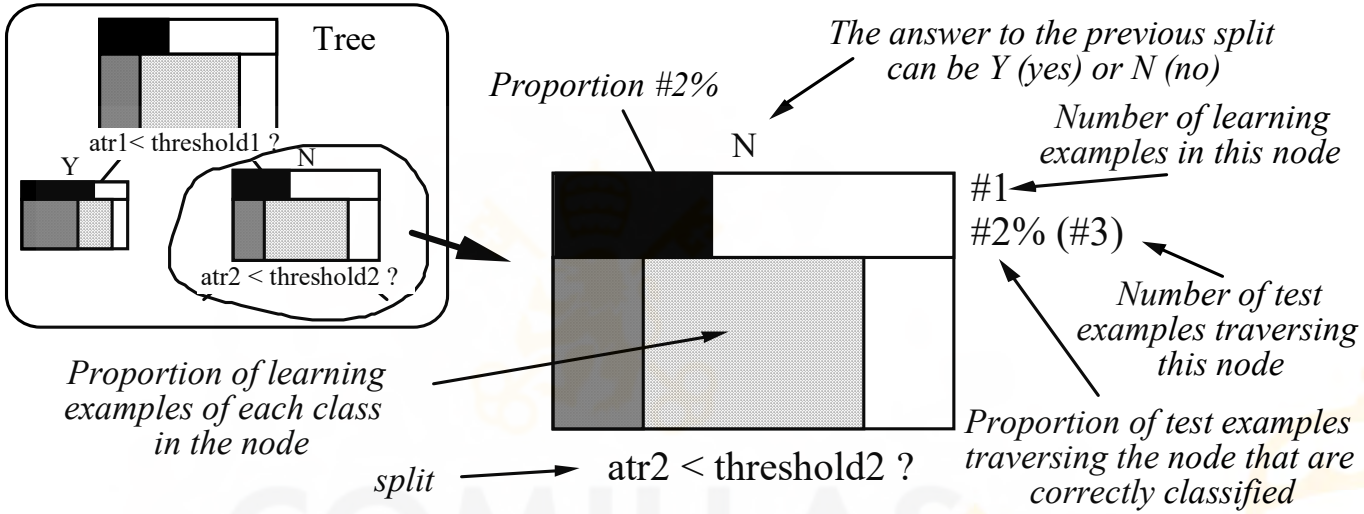
98% of training observations are correctly classified
(97% of the test set)

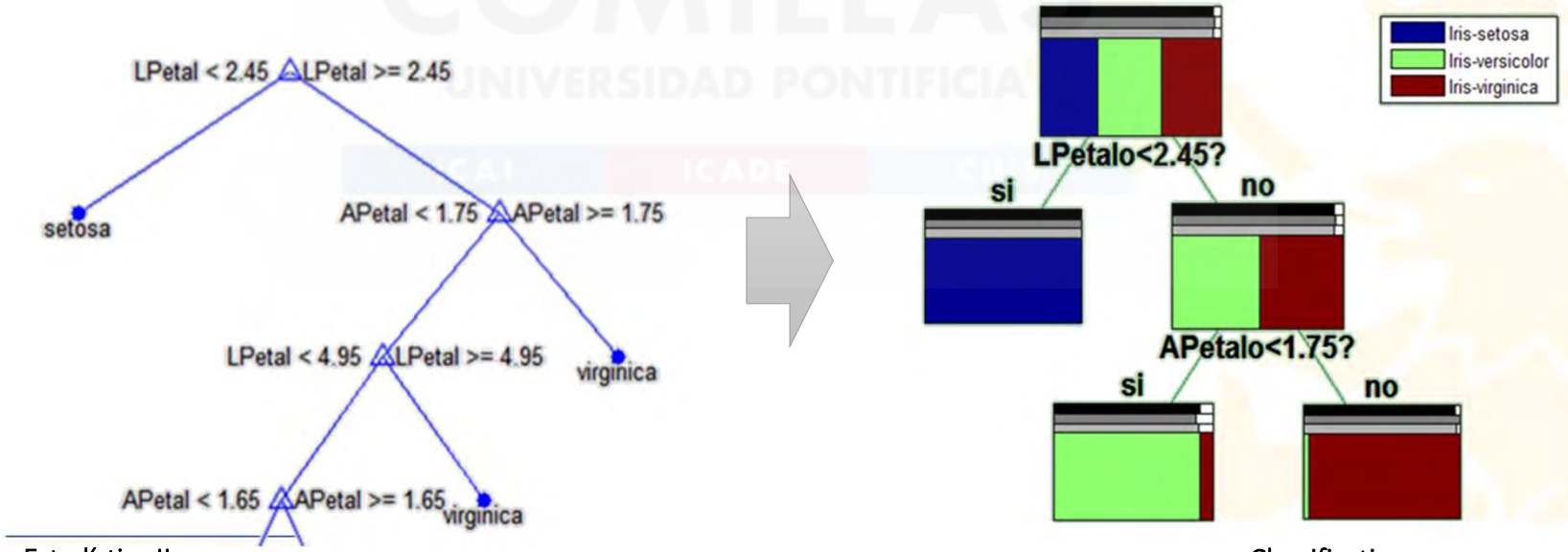


H_{min} = 0.3
(larger tree)

Both models have the same test error rate.
One should choose the simplest one

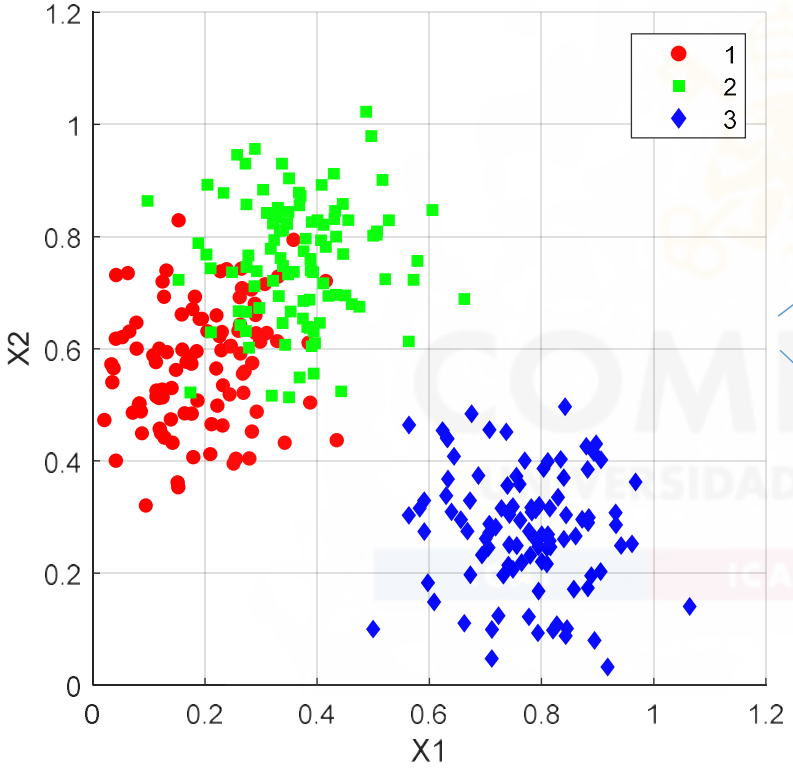
Classification trees
Wehenkel representation





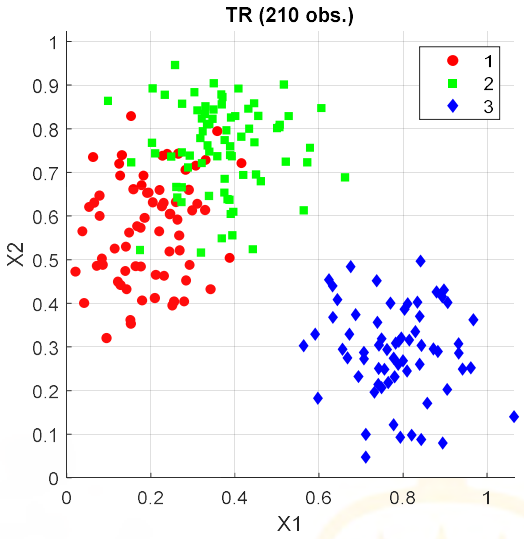
Classification trees
Illustrative synthetic cases

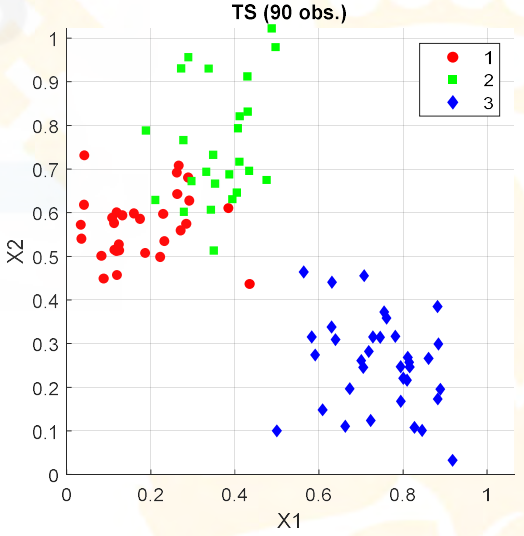
- C1: 3 classes with few data



70% for training

30% for testing



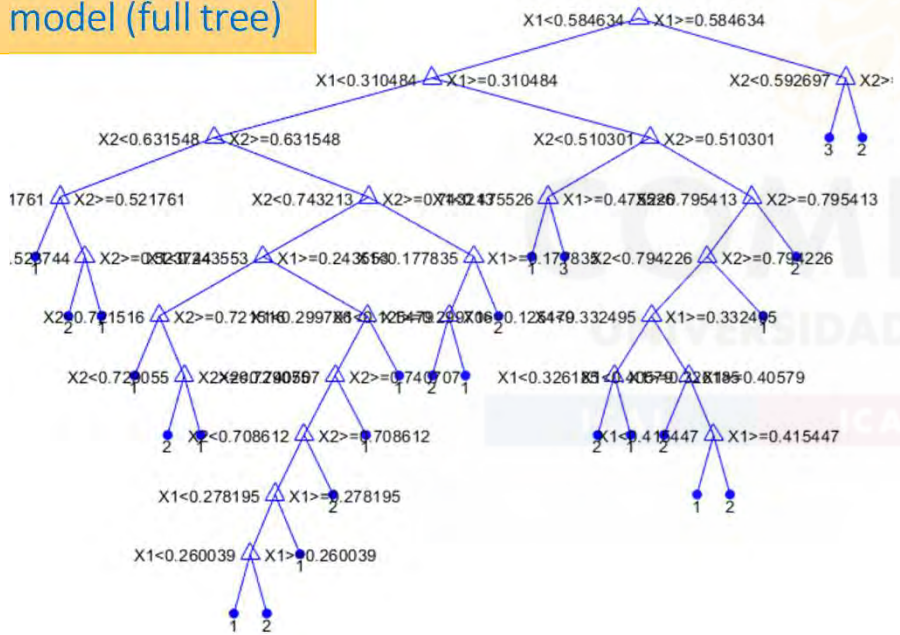


- Class 3 (blue) is easily split from the other two
- The border between classes 1 and 2 is not very clear

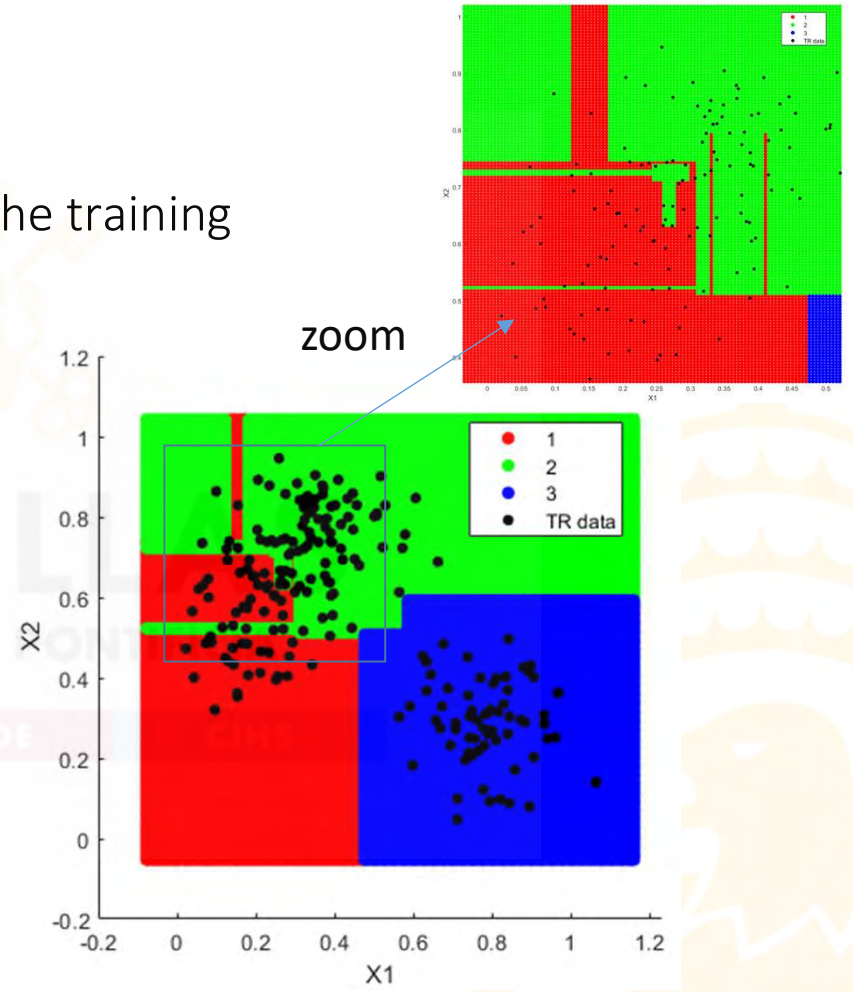
Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Fit a full classification tree
    - Built to fit perfectly into the training data

Very complex model (full tree)



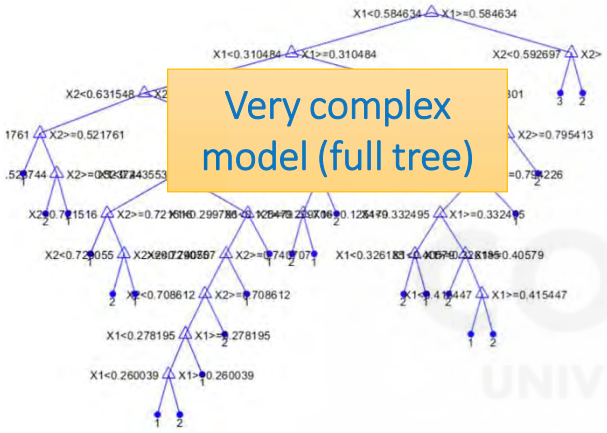
zoom



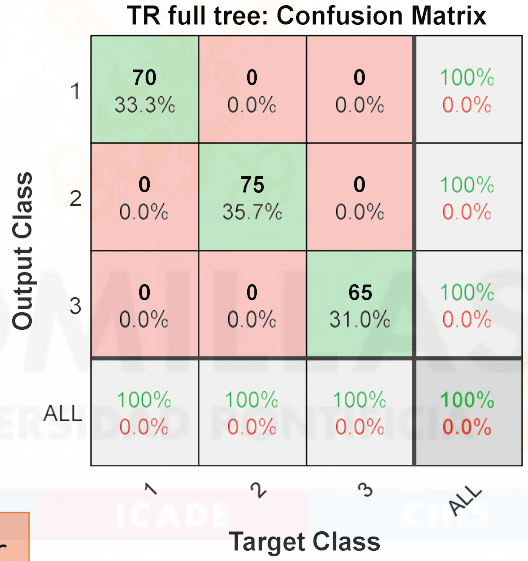

```
treeIni = fitctree(tr,'Y~X1+X2','SplitCriterion','deviance','MinParentSize',1);
view(treeIni,'mode','graph'); % show the tree
```

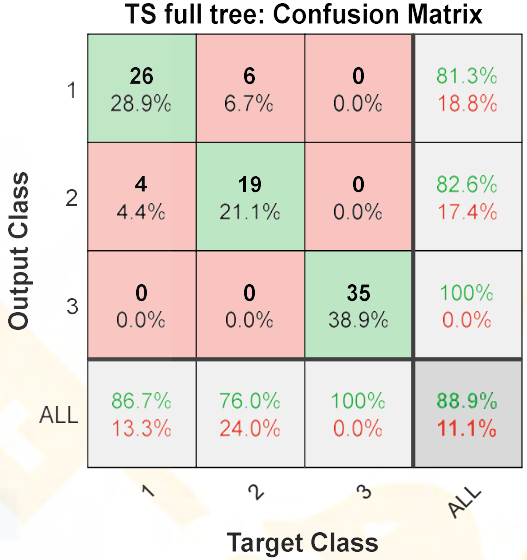
Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Confusion matrix (training and test sets)

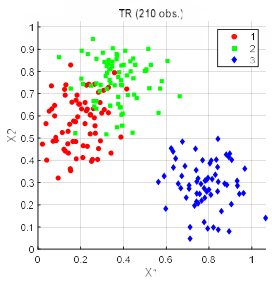


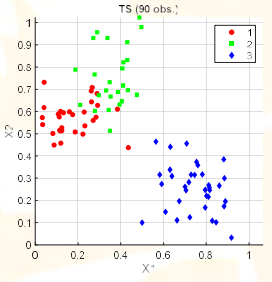
Very complex model (full tree)





Overfitted model: the error rate goes from 0% (TR) to 11.1% (TS)

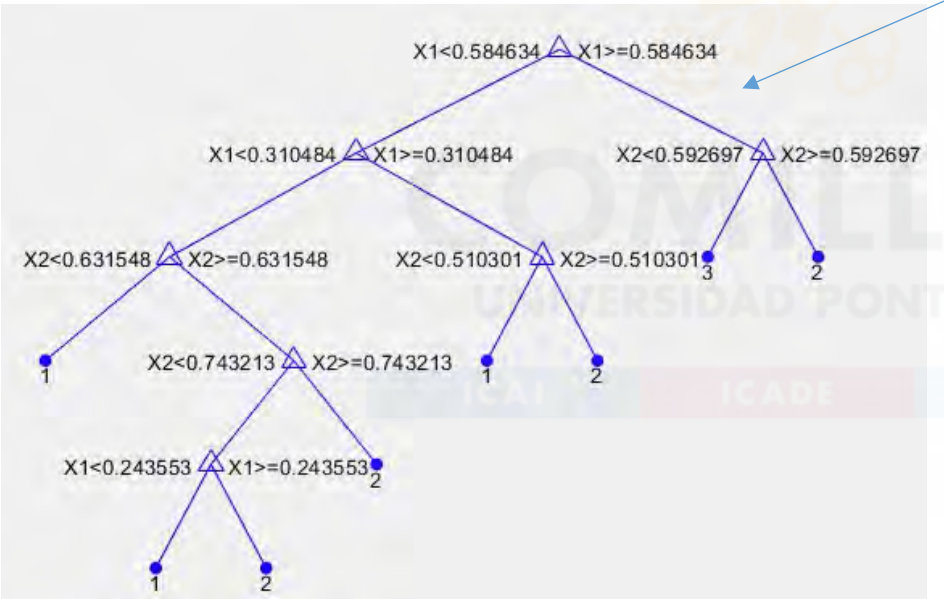


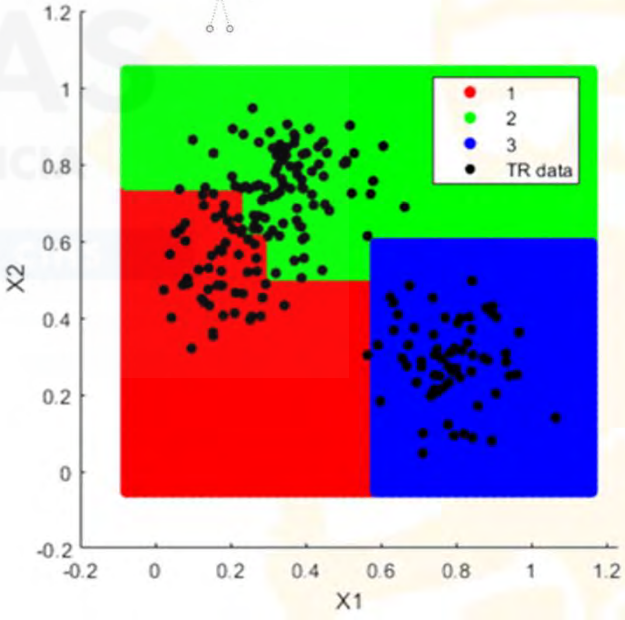


Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Fit a simpler classification tree by pruning the full tree

Simpler model
(tree with 15 nodes)

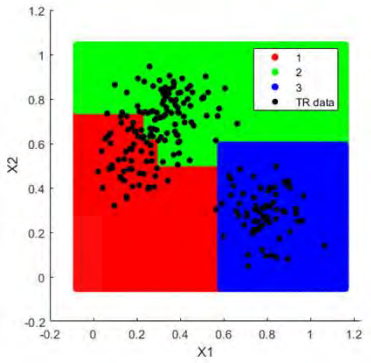


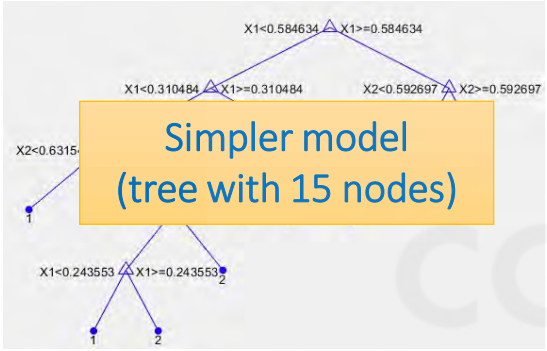



```
PrunLevel = 3;
tree1 = prune(treeIni,'Level', PrunLevel);
view(tree1,'mode','graph');
```

Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Confusion matrix (training and test sets)





The error rate goes from 6.2% (TR) to 11.1% (TS)

Overtrained model?

TR simple tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	60 28.6%	2 1.0%	1 0.5%	95.2% 4.8%
2	10 4.8%	73 34.8%	0 0.0%	88.0% 12.0%
3	0 0.0%	0 0.0%	64 30.5%	100% 0.0%
ALL	85.7% 14.3%	97.3% 2.7%	98.5% 1.5%	**93.8%** **6.2%**

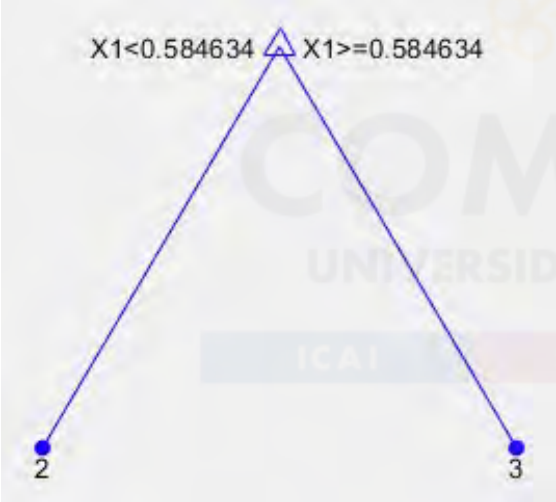
TS simple tree: Confusion Matrix

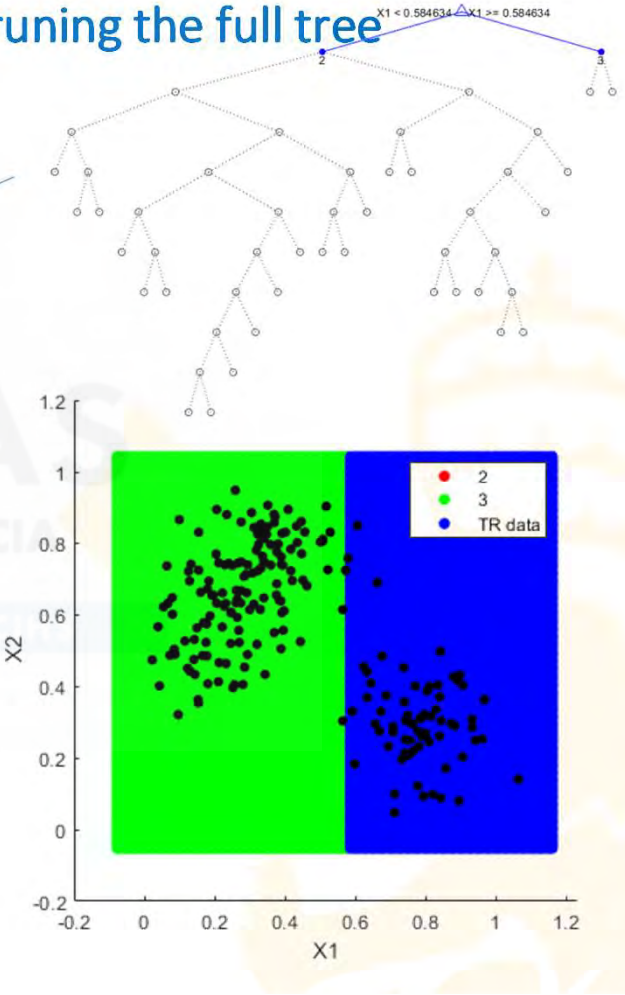
Output Class \ Target Class	1	2	3	ALL
1	25 27.8%	2 2.2%	3 3.3%	83.3% 16.7%
2	5 5.6%	23 25.6%	0 0.0%	82.1% 17.9%
3	0 0.0%	0 0.0%	32 35.6%	100% 0.0%
ALL	83.3% 16.7%	92.0% 8.0%	91.4% 8.6%	**88.9%** **11.1%**

Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Fit a very simple classification tree by pruning the full tree

Very simple model
(tree with 3 nodes)

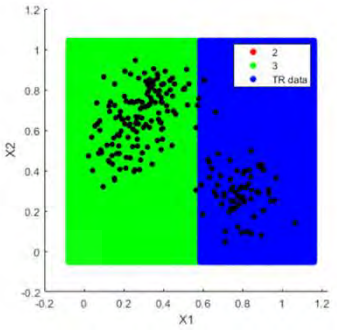


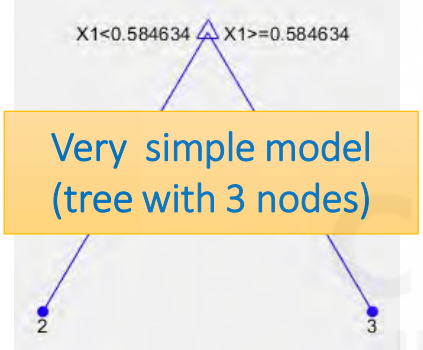



```
PrunLevel = 6;
tree2 = prune(treeIni,'Level', PrunLevel);
view(tree2,'mode','graph');
```

Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Confusion matrix (training and test sets)





Very simple model (tree with 3 nodes)

The error rate goes from 34.8% (TR) to 36.7% (TS)

A good model?

TR Very simple tree: Confusion Matrix

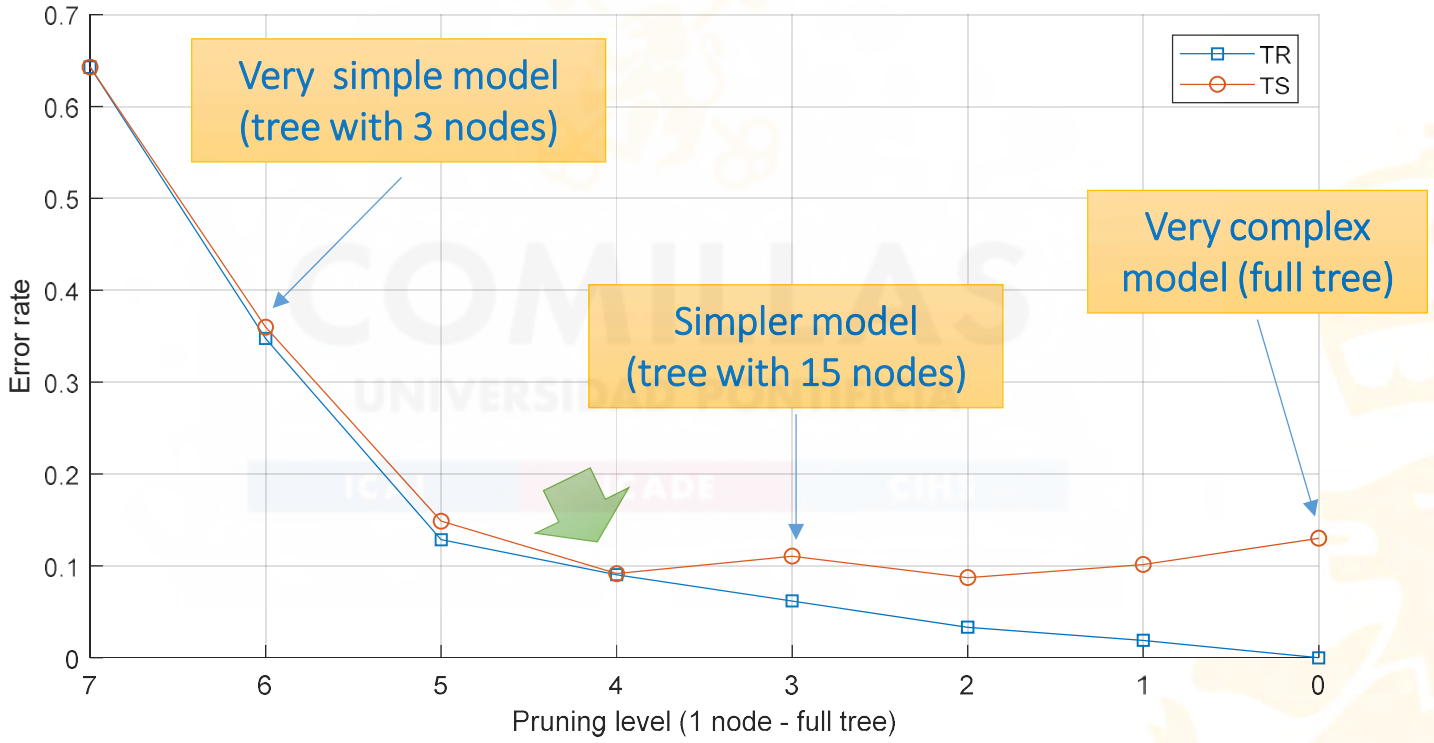
Output Class \ Target Class	1	2	3	ALL
1	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	70 33.3%	73 34.8%	1 0.5%	50.7% 49.3%
3	0 0.0%	2 1.0%	64 30.5%	97.0% 3.0%
ALL	0.0% 100%	97.3% 2.7%	98.5% 1.5%	**65.2%** **34.8%**

TS Very simple tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	30 33.3%	25 27.8%	3 3.3%	43.1% 56.9%
3	0 0.0%	0 0.0%	32 35.6%	100% 0.0%
ALL	0.0% 100%	100% 0.0%	91.4% 8.6%	**63.3%** **36.7%**

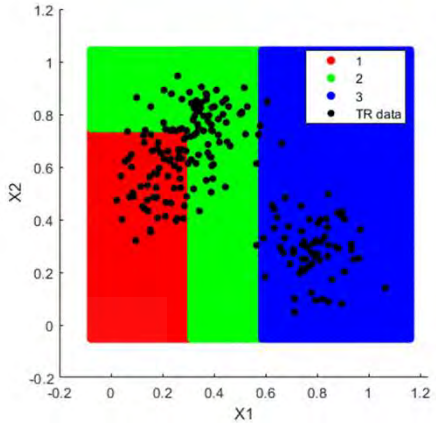
Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Select the correct complexity by pruning the initial model according to the TS error rate

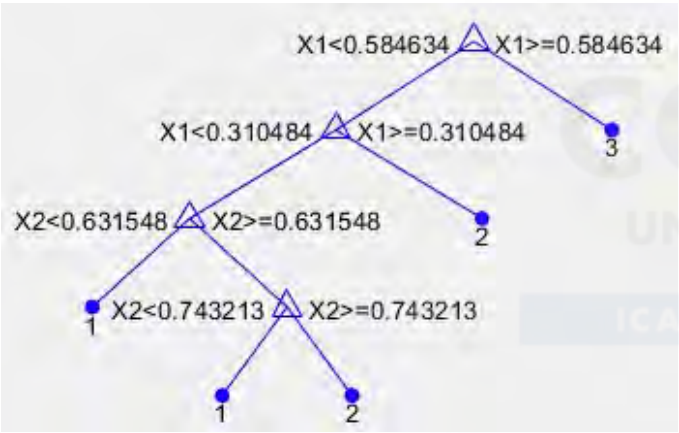


Classification trees
Illustrative synthetic cases

- C1: 3 classes with few data
  - Optimal tree: Good compromise between TS error and complexity



Optimal tree
(tree with 9 nodes)



TR Opt tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	63 30.0%	9 4.3%	0 0.0%	87.5% 12.5%
2	7 3.3%	64 30.5%	1 0.5%	88.9% 11.1%
3	0 0.0%	2 1.0%	64 30.5%	97.0% 3.0%
ALL	90.0% 10.0%	85.3% 14.7%	98.5% 1.5%	**91.0%** **9.0%**

TS Opt tree: Confusion Matrix

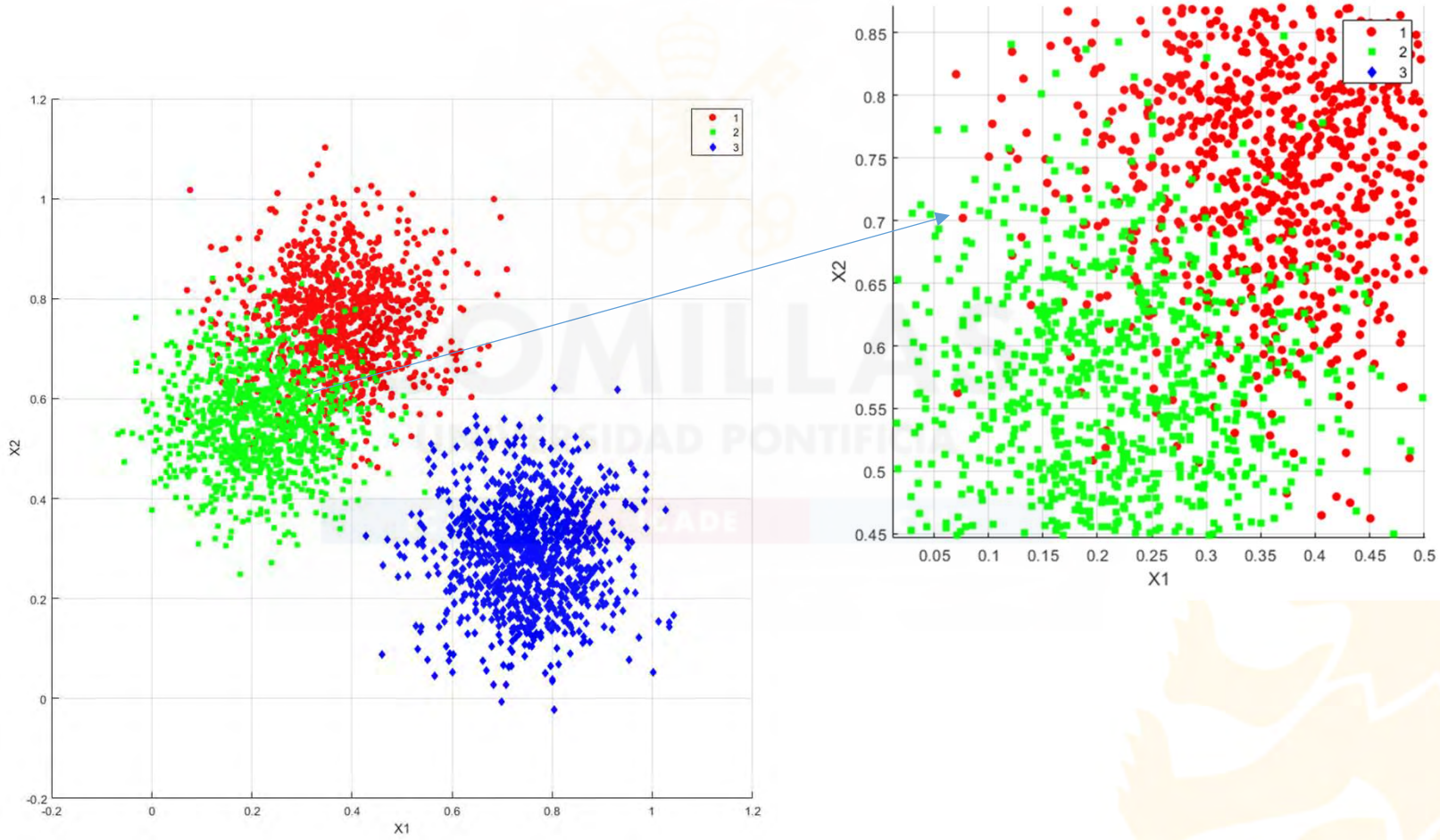
Output Class \ Target Class	1	2	3	ALL
1	28 31.1%	3 3.3%	0 0.0%	90.3% 9.7%
2	2 2.2%	22 24.4%	3 3.3%	81.5% 18.5%
3	0 0.0%	0 0.0%	32 35.6%	100% 0.0%
ALL	93.3% 6.7%	88.0% 12.0%	91.4% 8.6%	**91.1%** **8.9%**

```
PrunLevel = 4;
treeOpt = prune(treeIni,'Level', PrunLevel);
view(treeOpt,'mode','graph');
```

The error rate goes from 9.0% (TR) to 8.9% (TS)

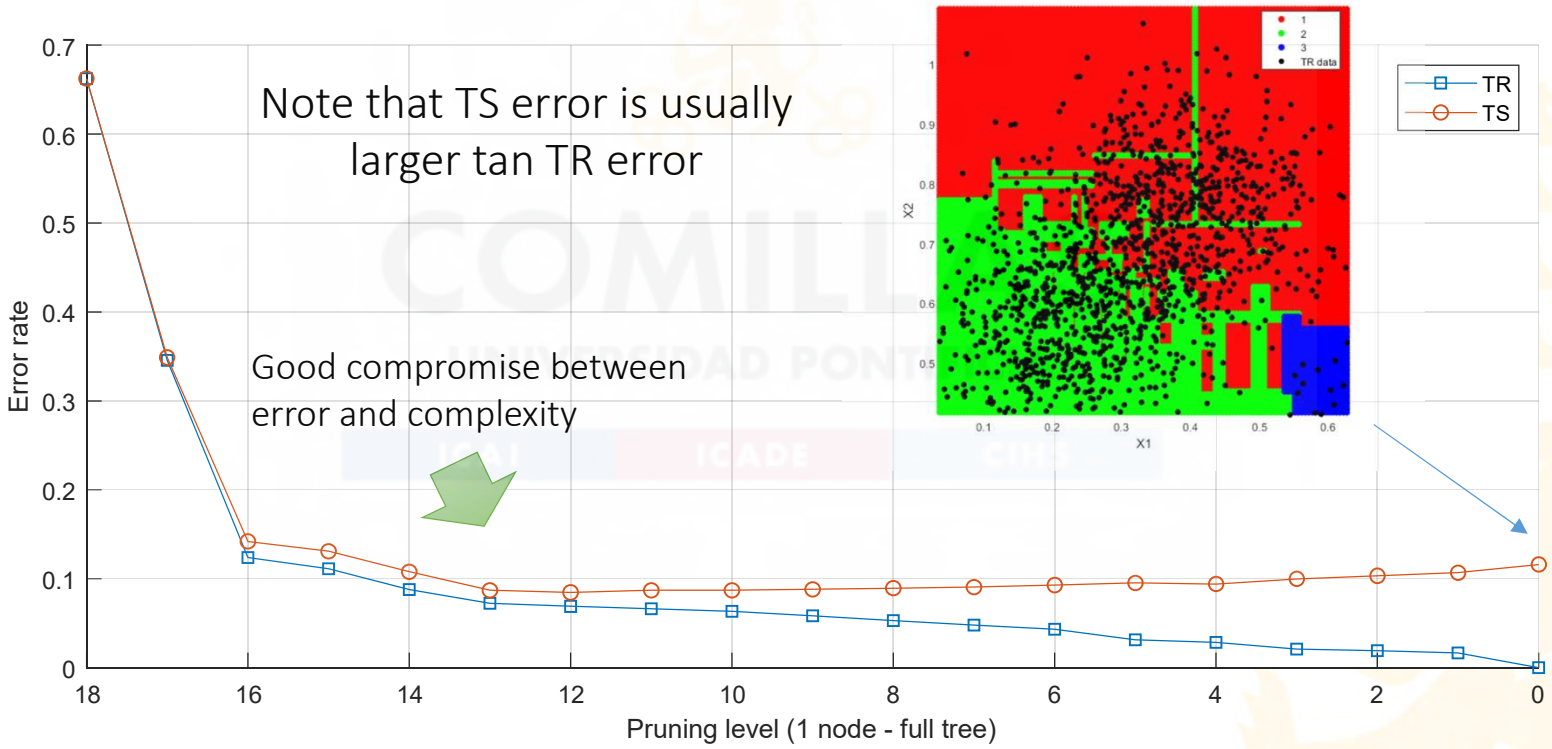
Classification trees
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)



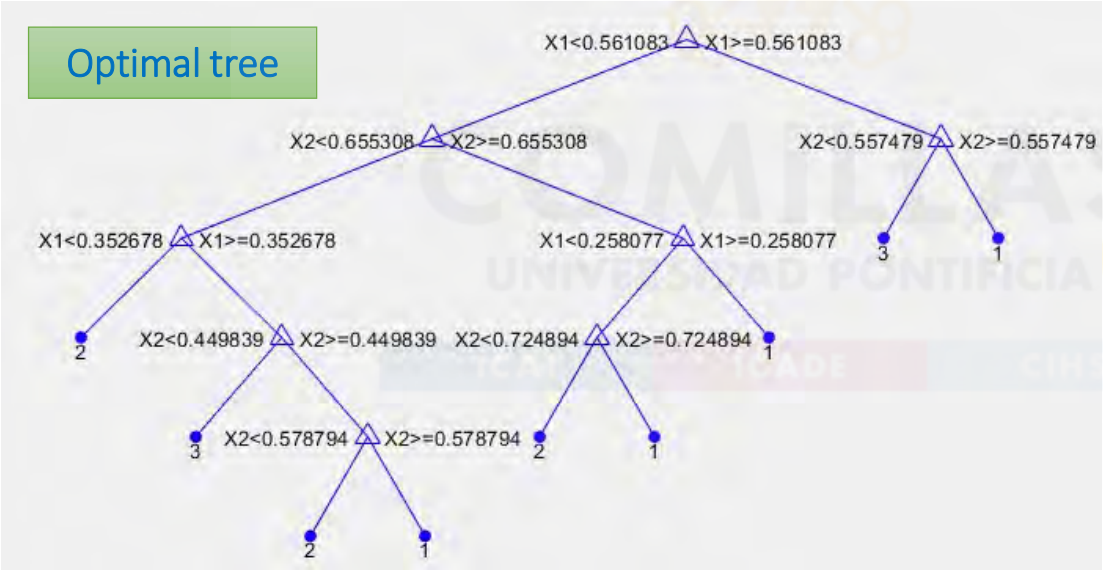
Classification trees
Illustrative synthetic cases

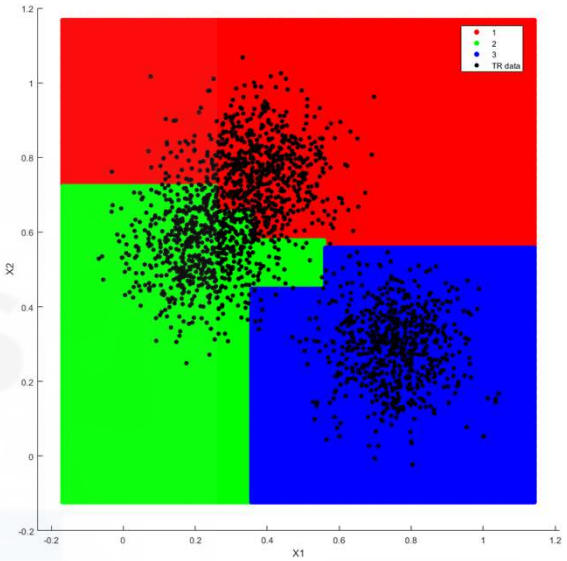
- C2: 3 classes with many data (3 x 1000)
  - Select the correct complexity by pruning the initial model according to the TS error rate



Classification trees
Illustrative synthetic cases

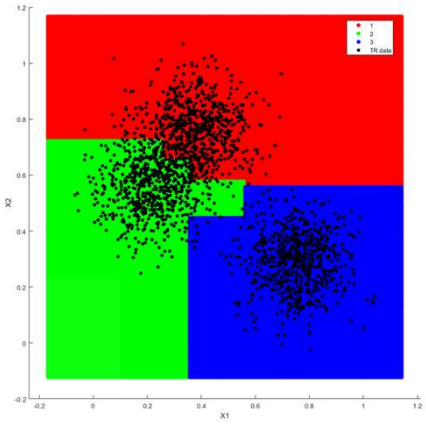
- C2: 3 classes with many data (3 x 1000)
  - Optimal tree: Good compromise between TS error and complexity






```
PrunLevel = 13;
treeOpt = prune(treeIni,'Level', PrunLevel);
view(treeOpt,'mode','graph');
```

Classification trees
Illustrative synthetic cases



- C2: 3 classes with many data (3 x 1000)
  - Optimal tree: Good compromise between TS error and complexity

Optimal tree

TR Optimal tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	643 30.6%	77 3.7%	0 0.0%	89.3% 10.7%
2	66 3.1%	623 29.7%	1 0.0%	90.3% 9.7%
3	0 0.0%	8 0.4%	682 32.5%	98.8% 1.2%
ALL	90.7% 9.3%	88.0% 12.0%	99.9% 0.1%	**92.8%** **7.2%**

TS Optimal tree: Confusion Matrix

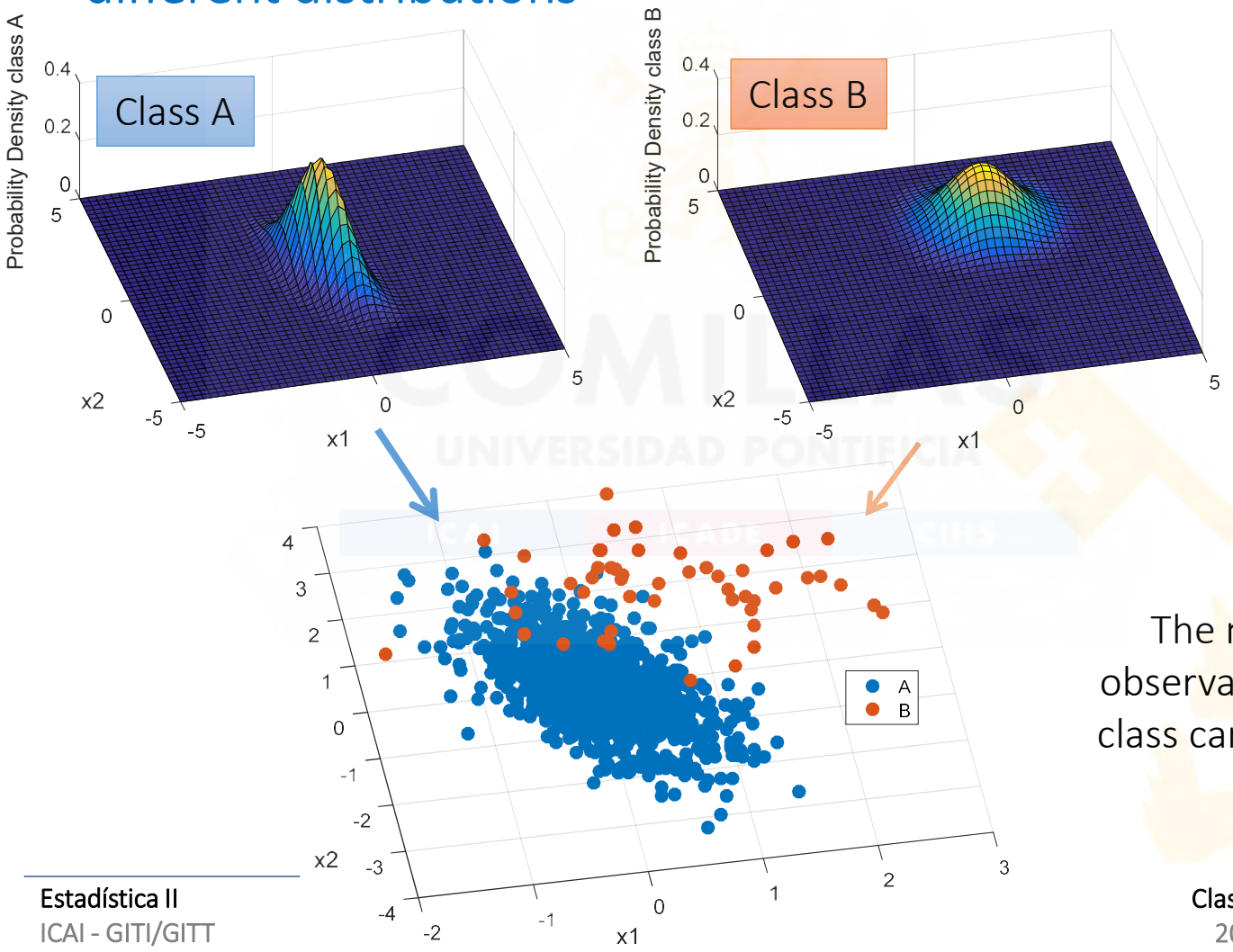
Output Class \ Target Class	1	2	3	ALL
1	261 29.0%	37 4.1%	5 0.6%	86.1% 13.9%
2	30 3.3%	252 28.0%	1 0.1%	89.0% 11.0%
3	0 0.0%	3 0.3%	311 34.6%	99.0% 1.0%
ALL	89.7% 10.3%	86.3% 13.7%	98.1% 1.9%	**91.6%** **8.4%**

4

Probabilistic approach: Linear Discriminant Analysis

Discriminant Analysis
Idea

- Assume that different output classes generate data based on different distributions



The number of observations of each class can be different

Discriminant Analysis
Overview

- This method first uses the multivariate Gaussian distribution to model, for each output class $\boldsymbol{k}$, the joint probability distribution of the inputs

$$f_k(X) \equiv \Pr(X = x|Y = k)$$

- Then, it uses Bayes' theorem to obtain the posterior probabilities

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

- Finally, it uses the Bayes classifier optimal rule to decide the estimated class from the previous posterior probabilities

Discriminant Analysis
Application case

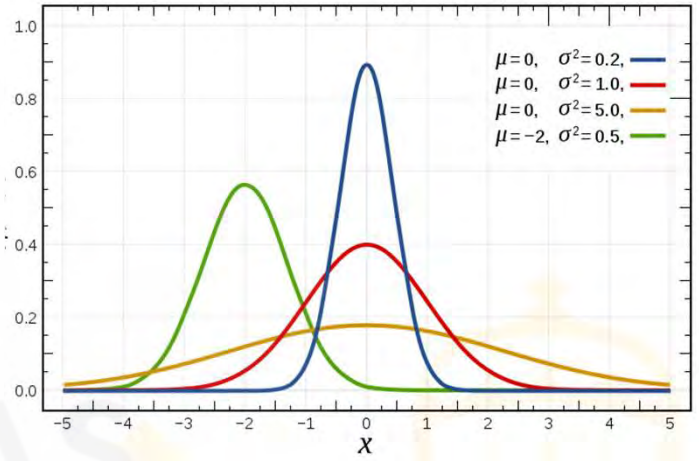
- k: class. Class A: male students. Class B: female students
- $x1, x2$: standardized grades for Algebra and Calculus
- $f_k(X)$ probability of having a grade of $x1$ and $x2$ given that you are male/female
- $\Pr(Y = k|X = x)$ probability of being male given that you have obtained $x1$ and $x2$ in your Algebra and Calculus grades = prob. of being male × prob. of having a grade of $x1$ and $x2$ given that you are male divided by sum for male and female of prob. of being male/female × prob. of having a grade of $x1$ and $x2$ given that you are male/female

Discriminant Analysis
Multivariate Gaussian distribution

- (Univariate) Gaussian distribution

$$X \approx N(\mu, \sigma^2)$$

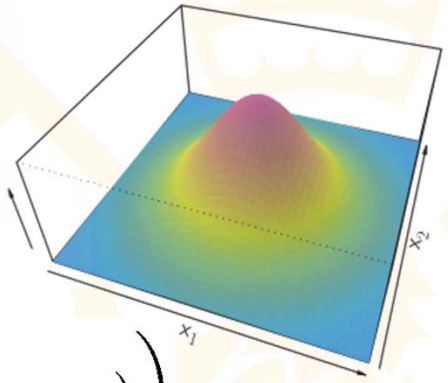
$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(x-\mu)^2/2\sigma^2}$$



Generalization to higher dimensions

- Multivariate Gaussian distribution

$$X \sim N(\mu, \Sigma)$$



$$f(x) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right)$$

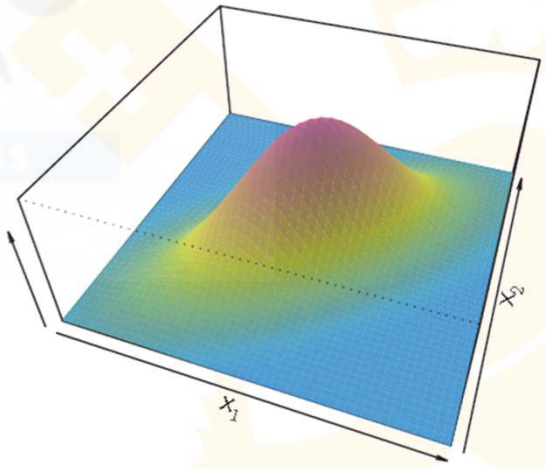
$$\|\mathbf{x}-\boldsymbol{\mu}\|^2_{\Sigma^{-1}}$$

Discriminant Analysis
Multivariate Gaussian distribution

- Parameters

$$X \sim N(\mu, \Sigma)$$

- $\boldsymbol{\mu}$: Vector of means (p-dimensional)
  - Center of the Gaussian in the p-dimensional input space
  - $\mu = E(X)$
- $\boldsymbol{\Sigma}$: Covariance matrix ($p \times p$ matrix)
  - Spread and Shape of the Gaussian
  - $\boldsymbol{\Sigma} = cov(X)$



Number of parameters: $p + p(p+1)/2$

Discriminant Analysis
Multivariate Gaussian distribution

- Bivariate Gaussian distribution ($\boldsymbol{p = 2}$)

$$X \sim N(\mu, \boldsymbol{\Sigma})$$

$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left[\frac{(x-\mu_x)^2}{\sigma_x^2} + \frac{(y-\mu_y)^2}{\sigma_y^2} - \frac{2\rho(x-\mu_x)(y-\mu_y)}{\sigma_x\sigma_y}\right]\right)$$

- Five parameters:

$$\boldsymbol{\mu} = \begin{pmatrix} \mu_x \\ \mu_y \end{pmatrix}, \quad \boldsymbol{\Sigma} = \begin{pmatrix} \sigma_x^2 & \rho\sigma_x\sigma_y \\ \rho\sigma_x\sigma_y & \sigma_y^2 \end{pmatrix}$$

Covariance between variables x and y

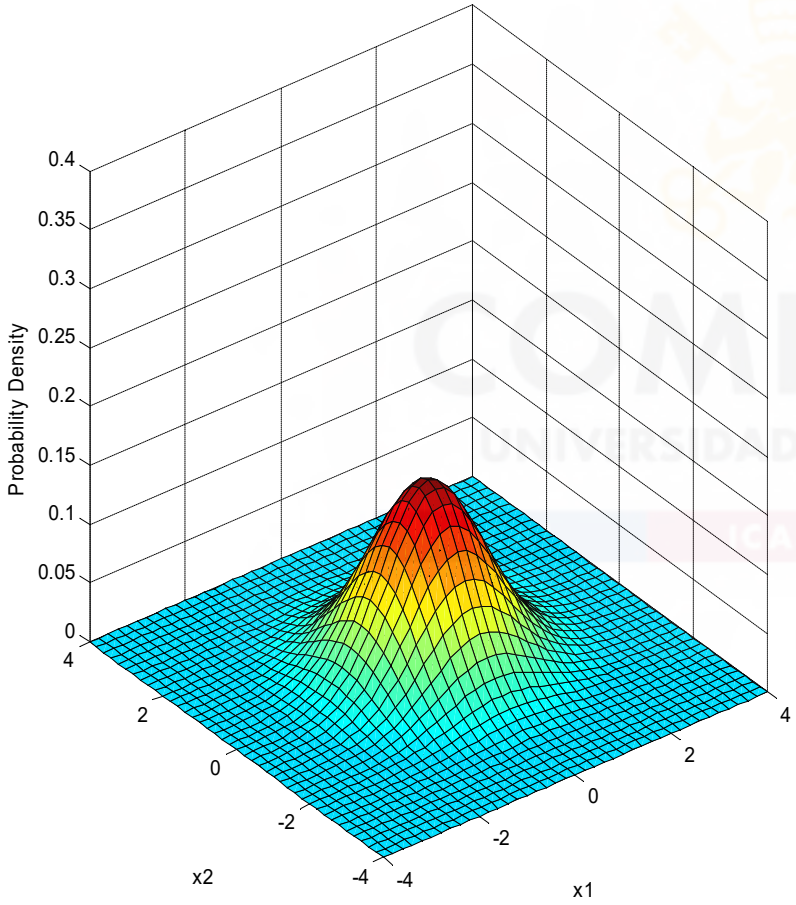
Pearson linear correlation coefficient

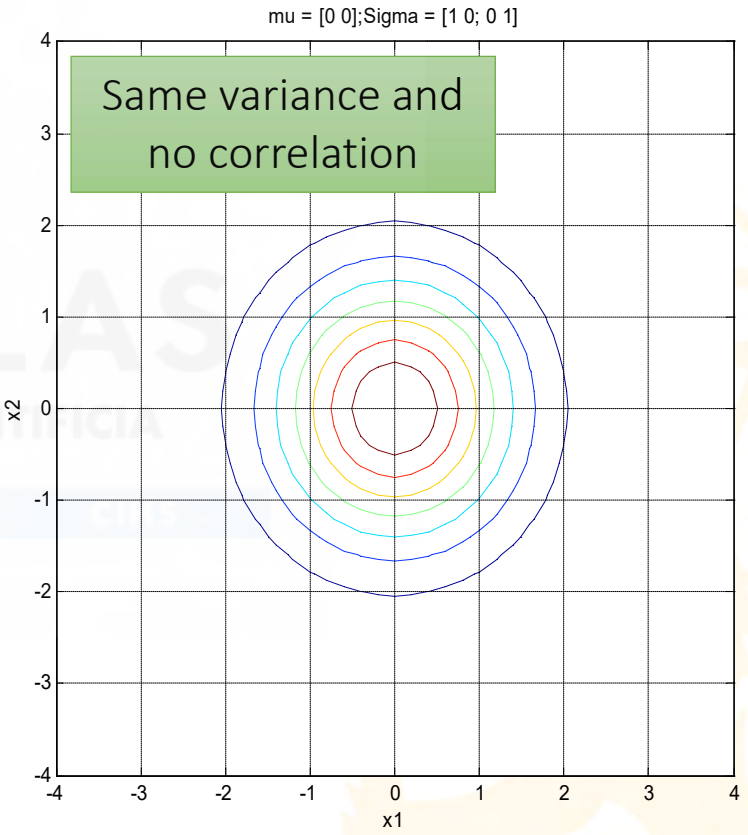
$$\rho_{X,Y} = \frac{\sigma_{XY}}{\sigma_X\sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X\sigma_Y}$$

Discriminant Analysis
Multivariate Gaussian distribution

- Example ($p = 2$)

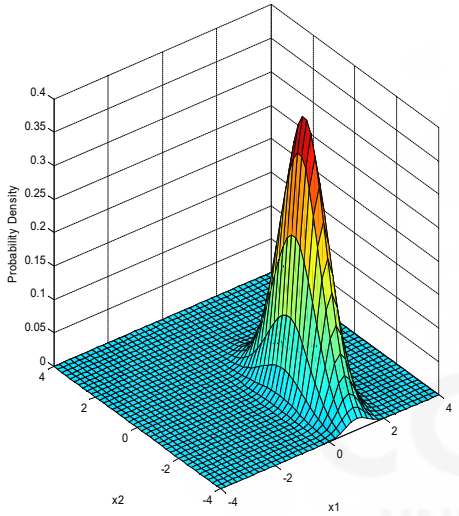
$$\boldsymbol{\mu} = \begin{pmatrix} 0.0 \\ 0.0 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$

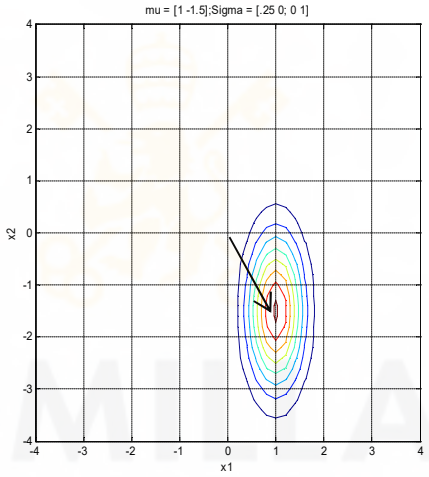




Discriminant Analysis
Multivariate Gaussian distribution

- Example ($p = 2$)

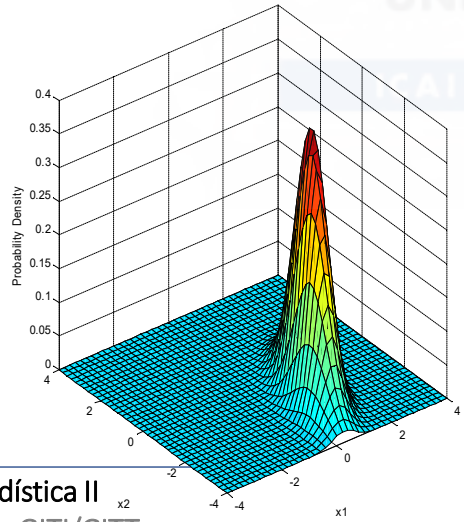


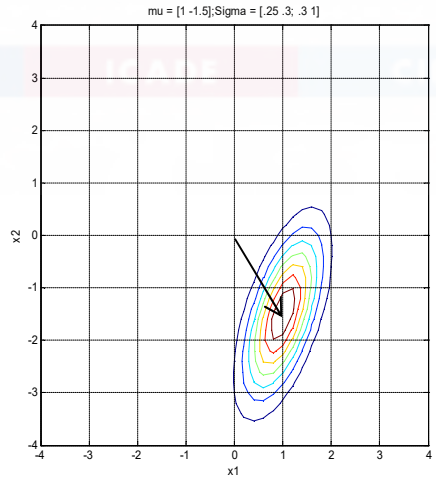


$$\boldsymbol{\mu} = \begin{pmatrix} 1.0 \\ -1.5 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 0.25 & 0.0 \\ 0.0 & 1.0 \end{pmatrix}$$

$$\rho = \frac{0.0}{\sqrt{0.25 \cdot 1}} = 0.0$$

var($x2$) is larger than var($x1$) , there is no correlation





$$\boldsymbol{\mu} = \begin{pmatrix} 1.0 \\ -1.5 \end{pmatrix} \quad \Sigma = \begin{pmatrix} 0.25 & 0.3 \\ 0.3 & 1.0 \end{pmatrix}$$

$$\rho = \frac{0.3}{\sqrt{0.25 \cdot 1}} = 0.6$$

var($x2$) is larger than var($x1$) , there is positive correlation

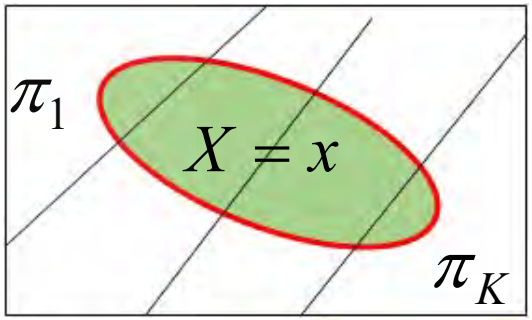
Discriminant Analysis
Using Bayes' Theorem for Classification

- Bayes' theorem states that

$$P(B_i|A) = P(A|B_i)P(B_i) \Big/ \sum_{j=1}^{n} [P(A|B_j)P(B_j)]$$

$$\Pr(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Posterior probabilities



- Thus, the ***K*** posterior probabilities can be computed by plugging in estimates of
  - The priors (easy if we have a random sample of the population)

$$\pi_1, \ldots, \pi_K$$

  - The likelihood (the difficult part, using Multivariate Gaussians)

$$f_k(X) \equiv \Pr(X = x|Y = k)$$

Discriminant Analysis
Bayes Classifier: the optimal rule

- Assign each observation to the most likely class, given its input values
  - This rule minimizes the TEST error rate

$$\Pr(Y = j | X = x_0) \quad \text{for } j = 1, \ldots$$

Assign the class j for which the conditional probability is the largest

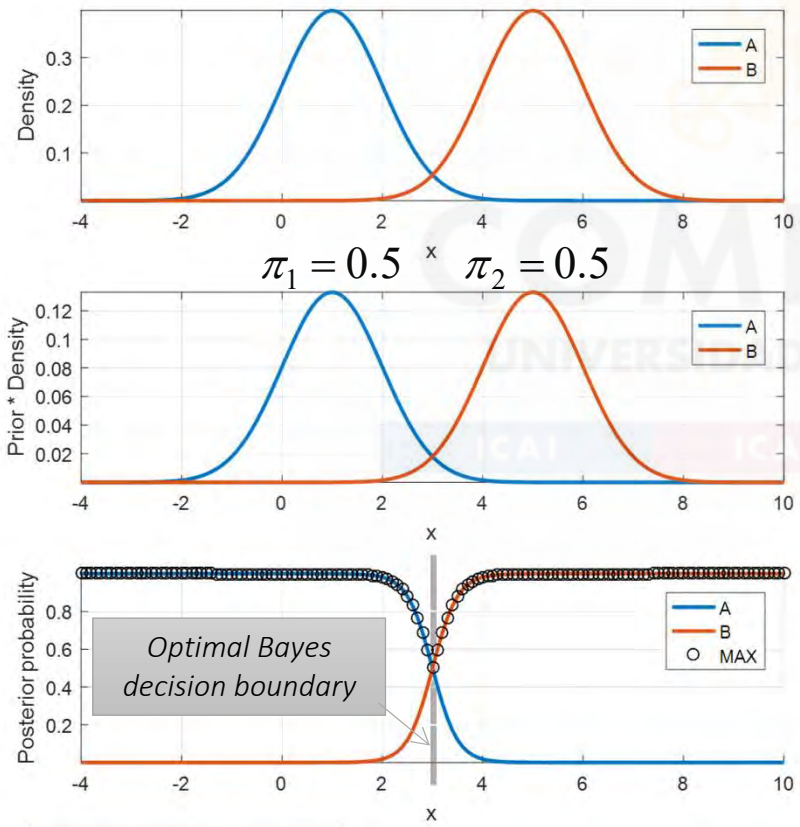
- Classification rule for two classes with equal priors (e.g., Y equals 1 or 2)

$$\Pr(Y = 1 | X = x_0) > 0.5 \quad \begin{array}{l} \xrightarrow{\text{YES}} \; Y = 1 \\ \xrightarrow{\text{NO}} \; Y = 2 \end{array}$$

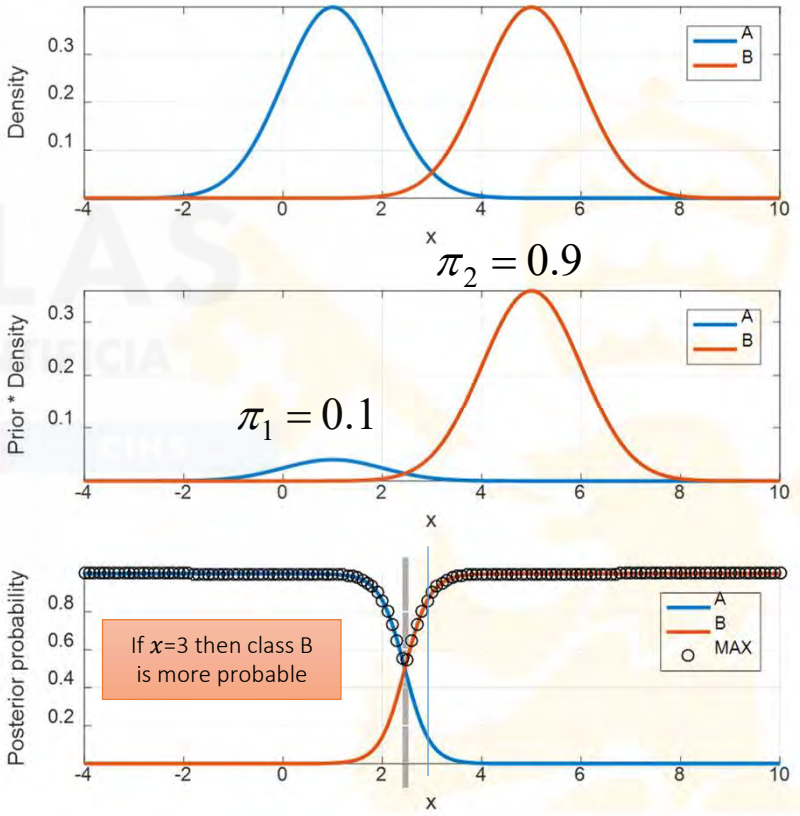
Discriminant Analysis
Bayes Classifier: the optimal rule

- Two-class problem with one input variable ($\boldsymbol{K = 2, p = 1}$)
  - Same probability for each class

Same priors



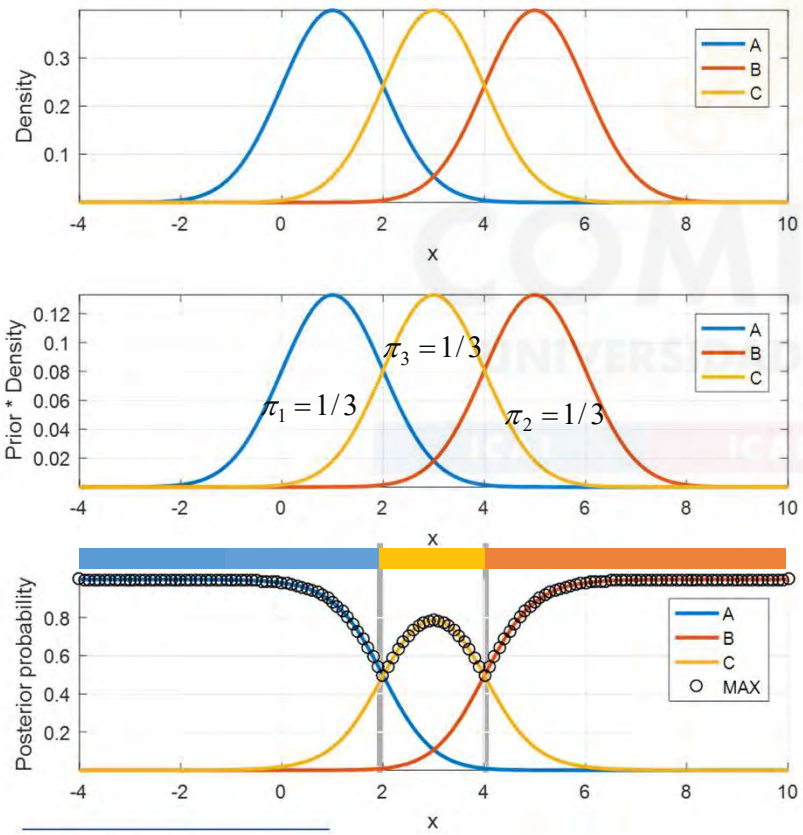
Different priors



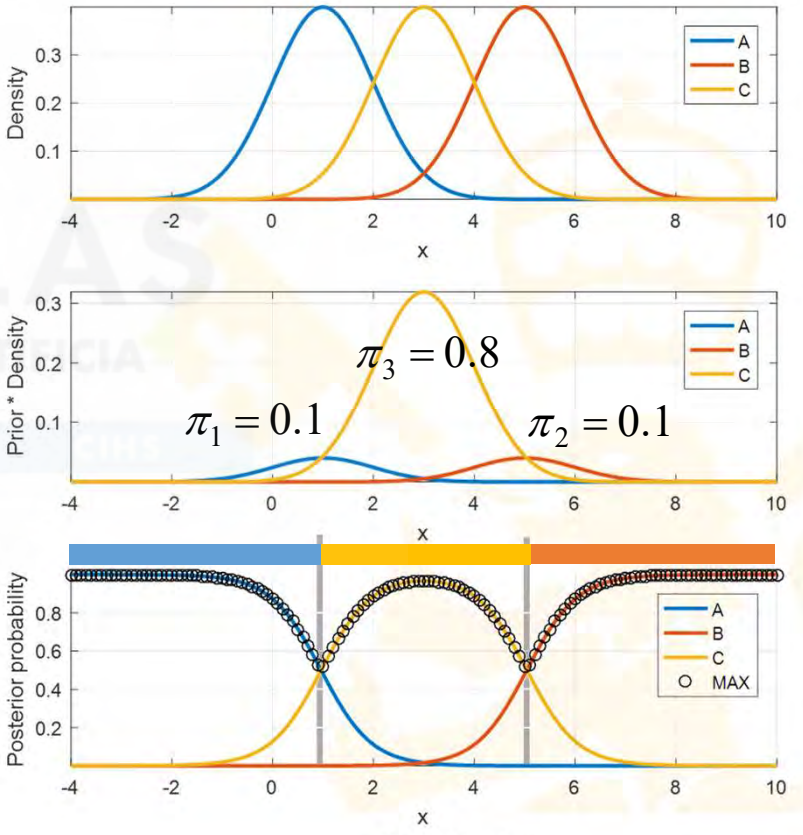
Discriminant Analysis
Bayes Classifier: the optimal rule

- Three-class problem with one input variable ($K = 3, p = 1$)
  - Same probability for each class

Same priors

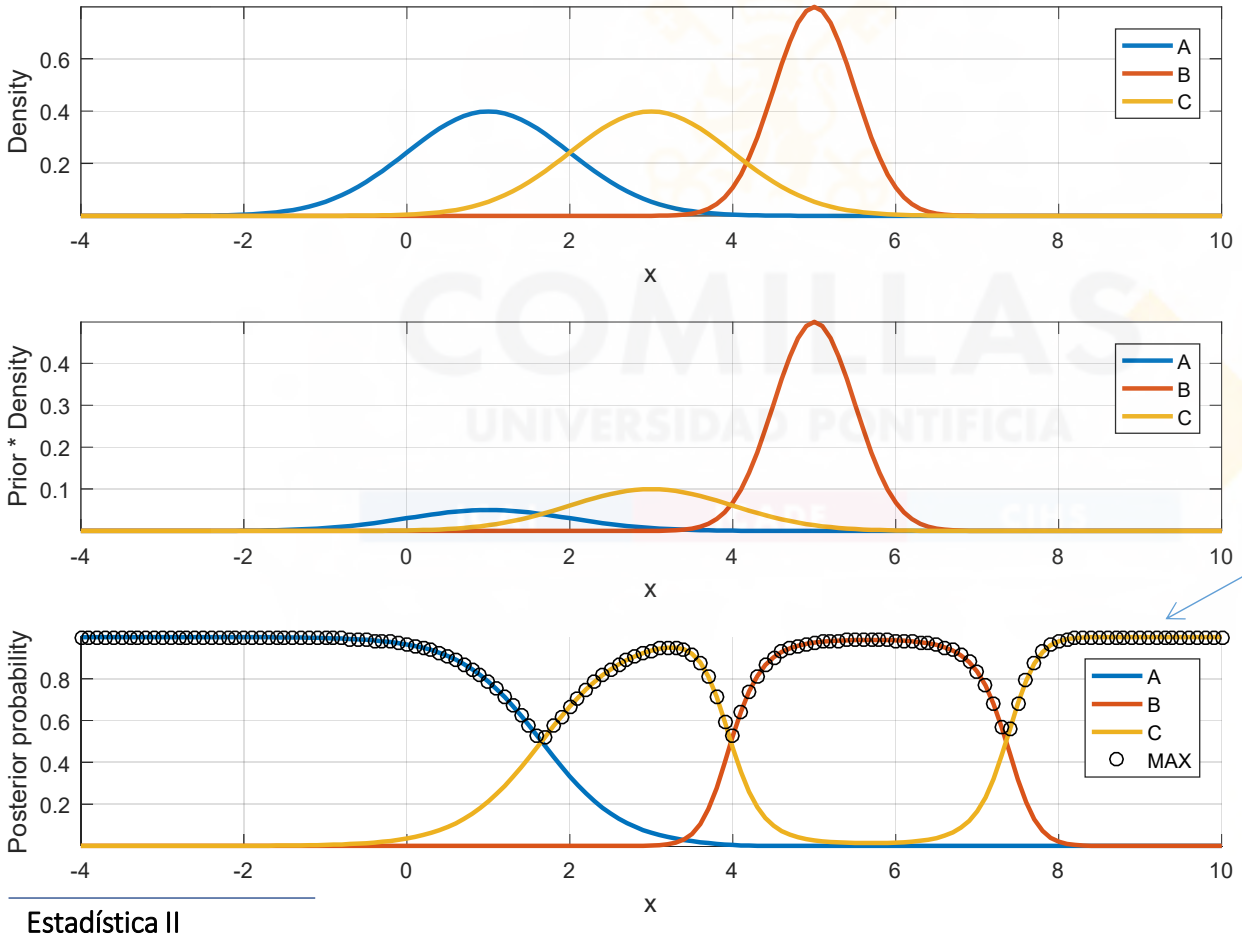


Different priors



Discriminant Analysis
Bayes Classifier: the optimal rule

- Three-class problem with one input variable ($K = 3, p = 1$)
  - Different probabilities and priors can imply very different posterior probabilities and partition of the input space



If $x > 7.35$ then $Y = C$

Discriminant Analysis
Linear Discriminant Analysis for p=1

- To estimate the densities, the linear version makes some assumptions about its shape
  - They are Gaussian
  - They have equal variance

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu_k)^2\right)$$

$$\sigma_1^2 = \ldots = \sigma_K^2$$

- Then, the posterior probabilities are given by

$$\Pr(Y=k|X=x) = \frac{\pi_k \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)}{\sum_{l=1}^{K} \pi_l \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2\sigma^2}(x-\mu_l)^2\right)}$$

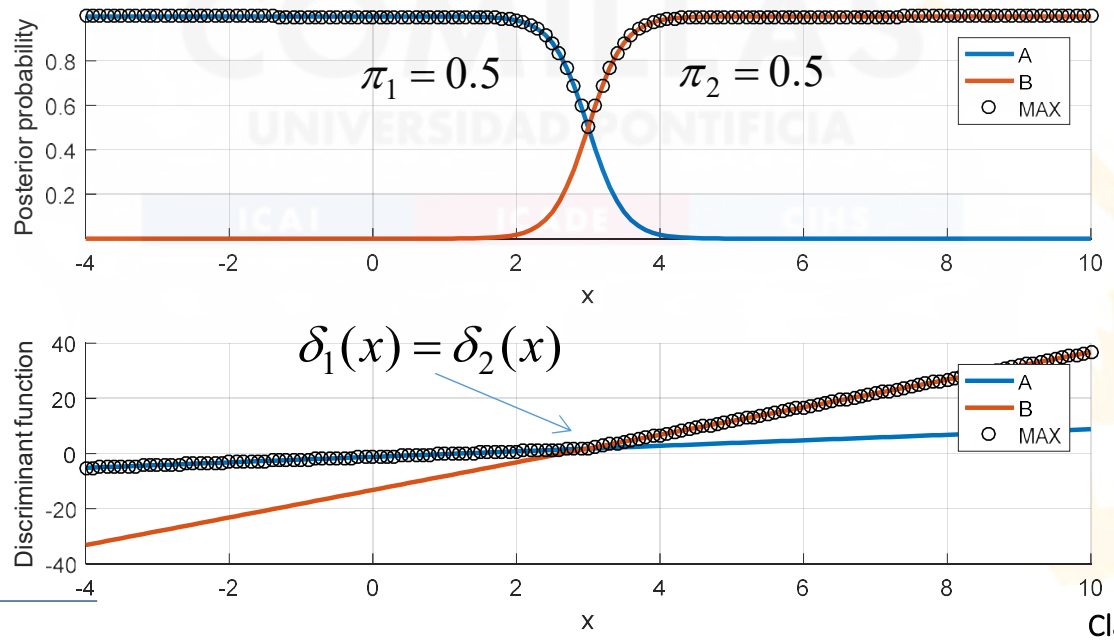
Discriminant Analysis
Linear Discriminant Analysis for p=1

- The Bayes classifier rule assigns an observation to the largest posterior probability. It is equivalent to assigning the observation to the class for which the discriminant function is the largest

$$\delta_k(x) = x \cdot \frac{\mu_k}{\sigma^2} - \frac{\mu_k^2}{2\sigma^2} + \log(\pi_k)$$

These are LINEAR functions of x

- Example



Discriminant Analysis
Linear Discriminant Analysis for p=1

- The LDA classifier first estimates all the required parameters of the classification model

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\pi}_k = n_k/n$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

- Then, it plugs the estimates into the discriminant functions

$$\hat{\delta}_k(x) = x \cdot \frac{\hat{\mu}_k}{\hat{\sigma}^2} - \frac{\hat{\mu}_k^2}{2\hat{\sigma}^2} + \log(\hat{\pi}_k)$$

- Finally, assign an observation $X = x$ to the class for which the discriminant function is the largest

Discriminant Analysis
Linear Discriminant Analysis for p>1

- To estimate the densities, the linear version makes some assumptions about its shape
  - They are multivariate Gaussian
  - They have an equal covariance matrix

$$N(\mu_k, \boldsymbol{\Sigma})$$

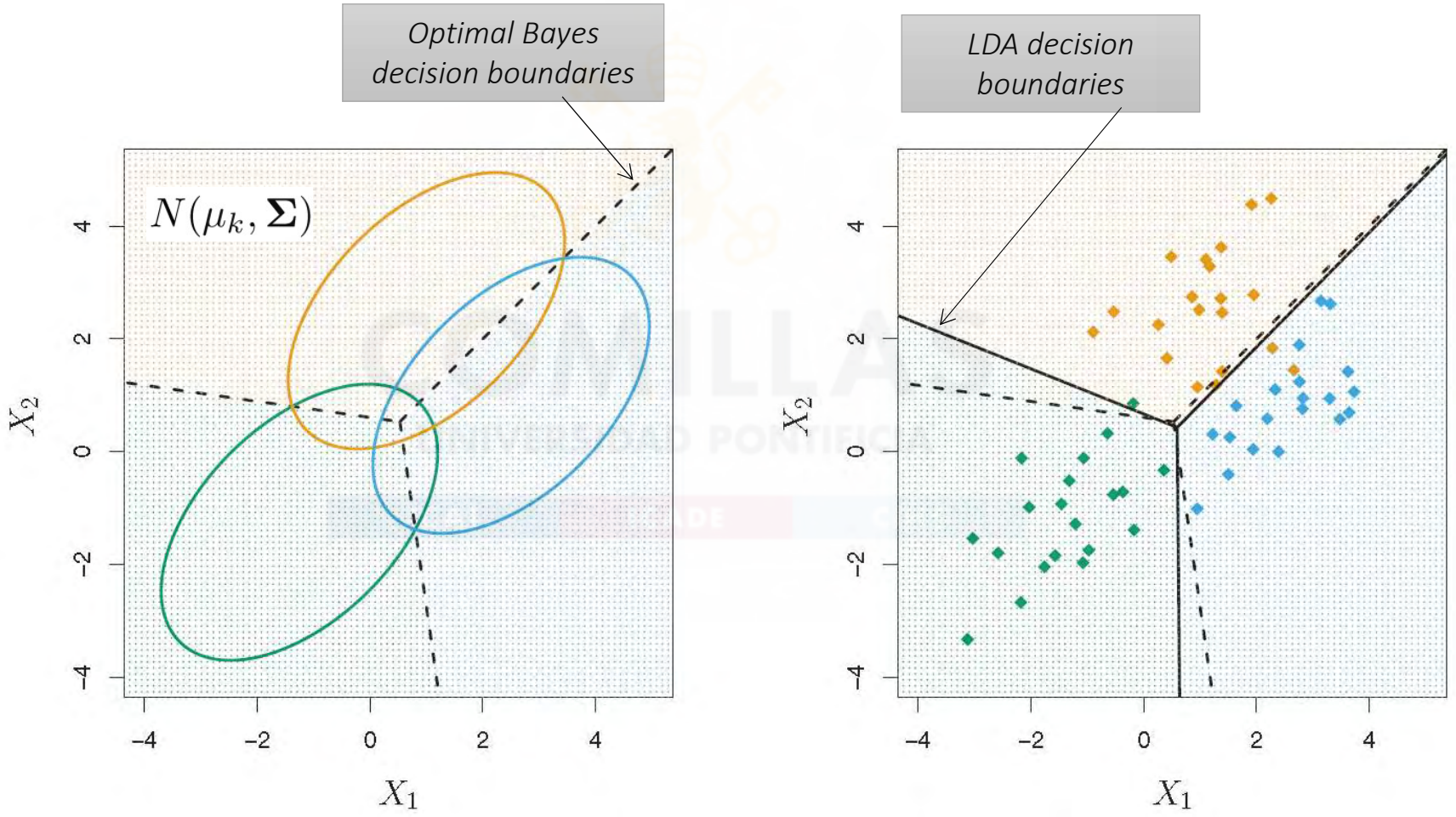
$$f(x) = \frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\boldsymbol{\Sigma}^{-1}(x-\mu)\right)$$

- In this case, the discriminant function is

$$\delta_k(x) = x^T\boldsymbol{\Sigma}^{-1}\mu_k - \frac{1}{2}\mu_k^T\boldsymbol{\Sigma}^{-1}\mu_k + \log\pi_k$$

Discriminant Analysis
Linear Discriminant Analysis for p>1

- Three-class example (20 observations for each class)



Discriminant Analysis
Quadratic Discriminant Analysis

- Alternative approach to LDA
- QDA assumes that the observations of each class are drawn from a Multivariate Gaussian distribution, but each class has its covariance matrix

$$N(\mu_k, \Sigma) \quad \Rightarrow \quad X \sim N(\mu_k, \Sigma_k)$$

LDA: $Kp + p(p+1)/2$

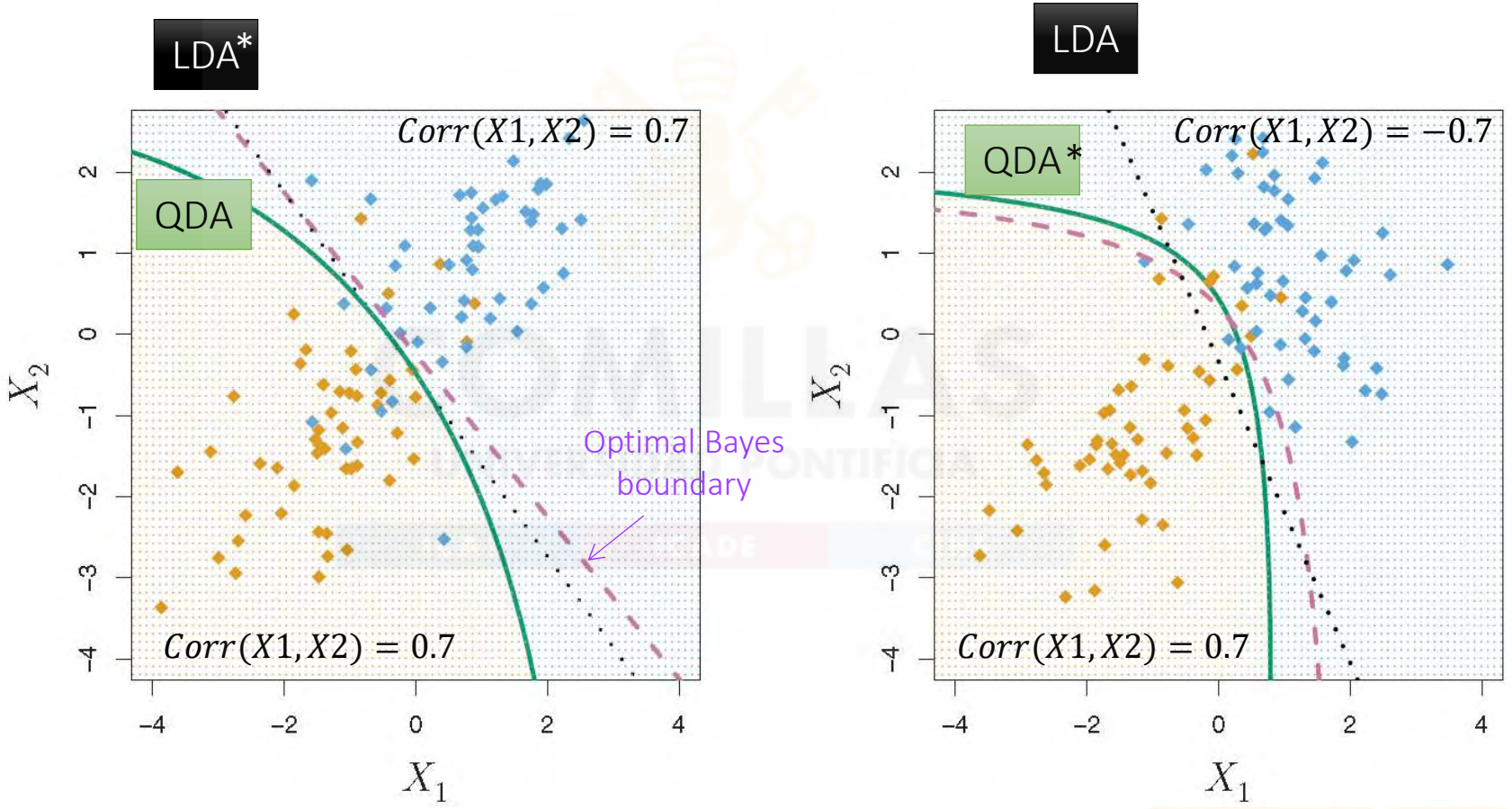
QDA: $Kp + Kp(p+1)/2$

- In this case, the discriminant function is quadratic in x

$$\begin{aligned}\delta_k(x) &= -\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k) - \frac{1}{2}\log|\Sigma_k| + \log\pi_k \\ &= -\frac{1}{2}x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma_k^{-1}\mu_k - \frac{1}{2}\log|\Sigma_k| + \log\pi_k\end{aligned}$$

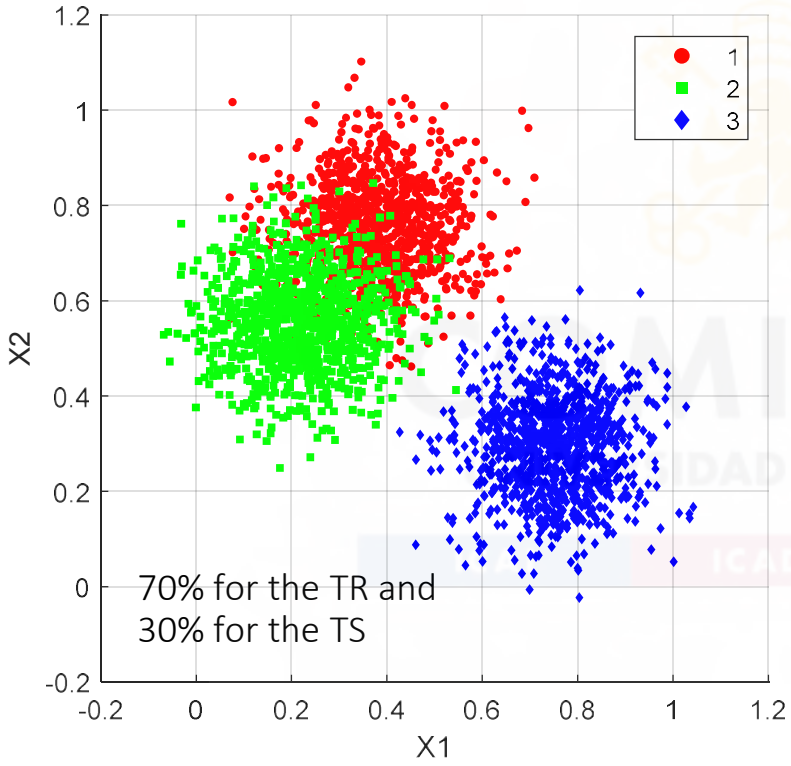
Discriminant Analysis
Quadratic Discriminant Analysis

- Two-class example (50 observations for each class)

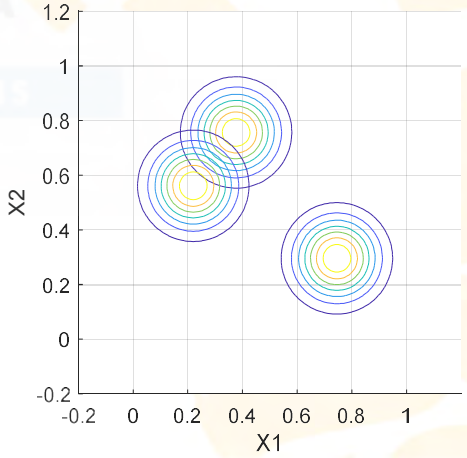


Discriminant Analysis
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)
  - IDEAL case for Linear Discriminant Analysis



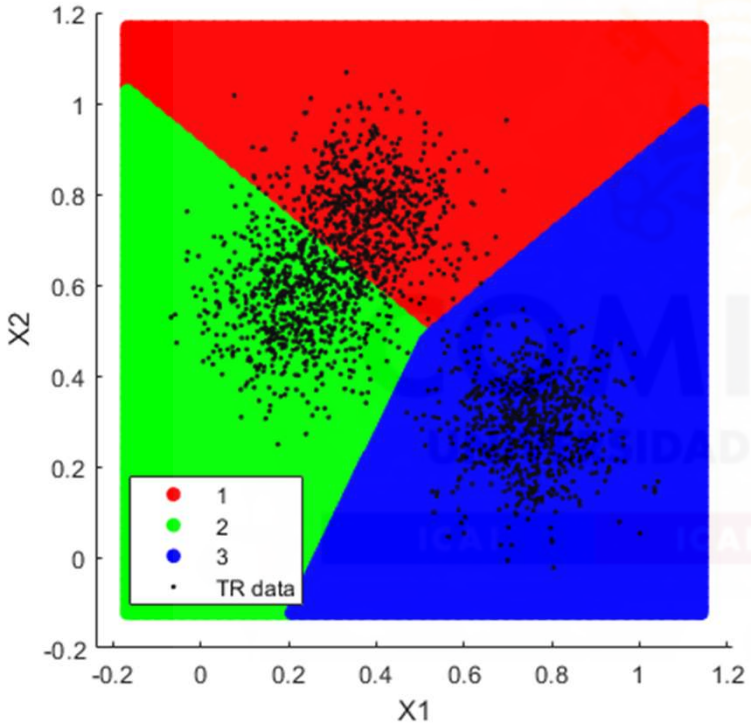
- This dataset has been generated by drawing random data for three different Multinormal distributions according to the following TRUE parameters:
  - $\mu_1 = (0.3767 \quad 0.7566)$
  - $\mu_2 = (0.2200 \quad 0.5612)$
  - $\mu_3 = (0.7454 \quad 0.2959)$
  - $\Sigma = \begin{pmatrix} 0.01 & 0.0 \\ 0.0 & 0.01 \end{pmatrix}$



- Class 3 (blue) is easy to split from the other two
- The border between classes 1 and 2 is not very clear

Discriminant Analysis
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)
  - Fit a Linear Discriminant (LDA) using TR data (2100 obs.)




```
lda = fitcdiscr(tr, 'Y~X1+X2');

Estimated Mu's
    0.3787    0.7588
    0.2184    0.5595
    0.7520    0.2938

Estimated Sigma
    0.0101    0.0001
    0.0001    0.0100
```

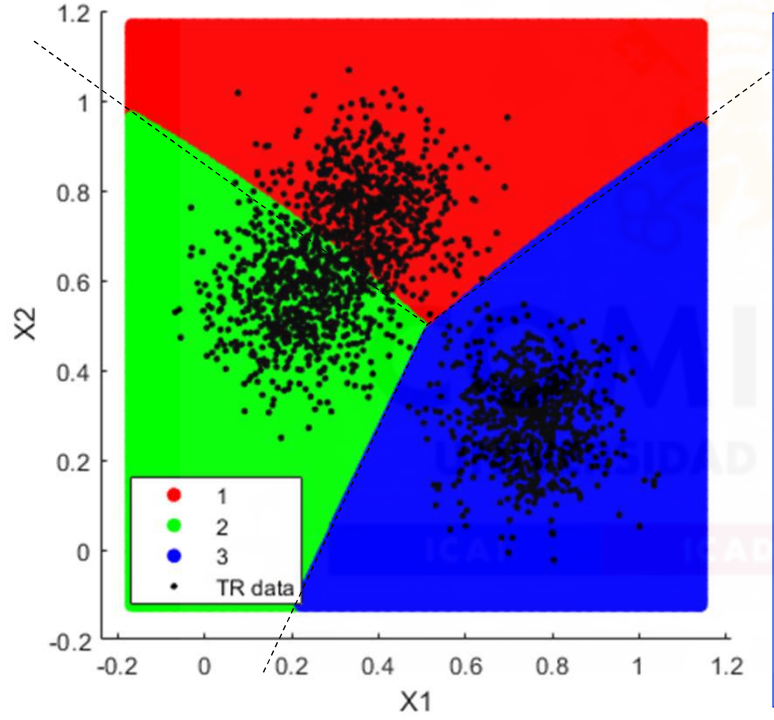
LDA

The same sigma for all the classes

Note that the estimated class borders are straight lines (linear discriminants)

Discriminant Analysis
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)
  - Fit a quadratic discriminant (QDA) using TR data (2100 obs.)




```
qda =
fitcdiscr(tr,'Y~X1+X2','DiscrimType','quadratic');

Estimated Mu's
    0.3787    0.7588
    0.2184    0.5595
    0.7520    0.2938

Estimated Sigma class 1
    0.0106    0.0000
    0.0000    0.0102

Estimated Sigma class 2
    0.0102    0.0003
    0.0003    0.0097

Estimated Sigma class 3
    0.0095   -0.0001
   -0.0001    0.0101
```

QDA

They are quite similar

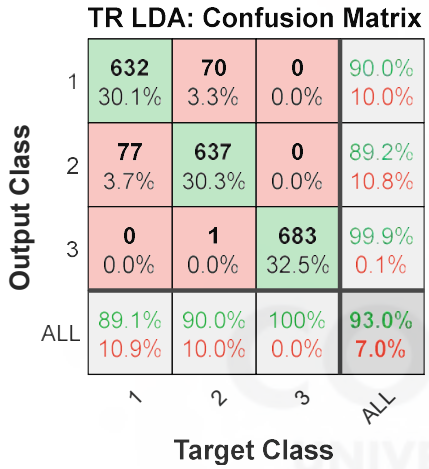
Note that the estimated class borders aren't perfect straight lines (quadratic discriminants)

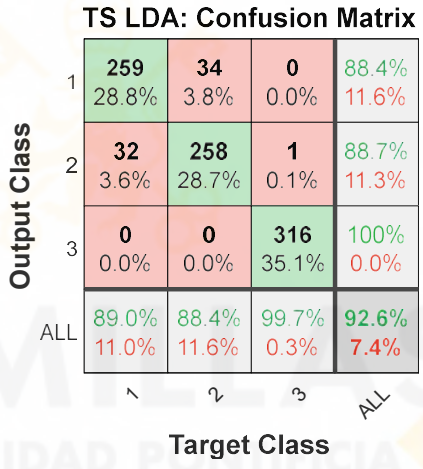
Due to the large dataset, QDA has low overfitting

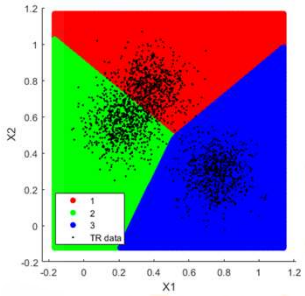
Discriminant Analysis
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)
  - LDA vs. QDA (confusion matrix, training, and test sets)

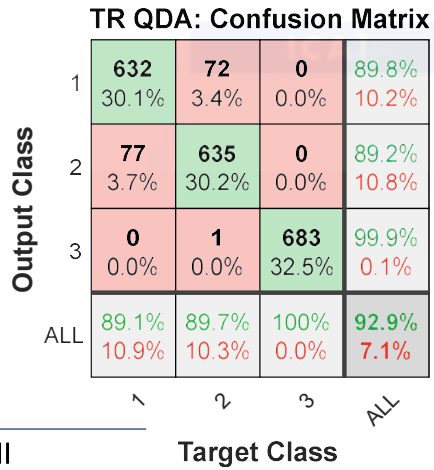
LDA

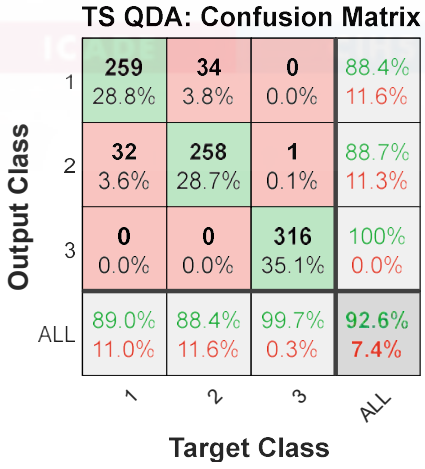


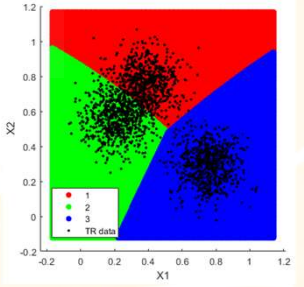




QDA







LDA is better than QDA in this problem (less complex with very similar error rates)

Discriminant Analysis
Illustrative synthetic cases

- C2: 3 classes with many data (3 x 1000)
  - LDA vs. Classification tree (confusion matrix, training, and test sets)

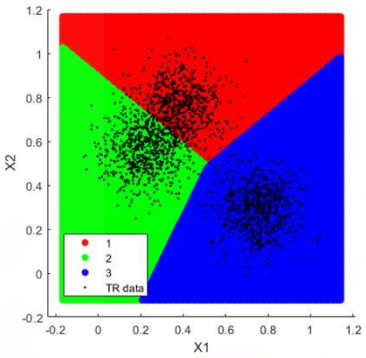
LDA

TR LDA: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	632 30.1%	70 3.3%	0 0.0%	90.0% 10.0%
2	77 3.7%	637 30.3%	0 0.0%	89.2% 10.8%
3	0 0.0%	1 0.0%	683 32.5%	99.9% 0.1%
ALL	89.1% 10.9%	90.0% 10.0%	100% 0.0%	**93.0%** **7.0%**

TS LDA: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	259 28.8%	34 3.8%	0 0.0%	88.4% 11.6%
2	32 3.6%	258 28.7%	1 0.1%	88.7% 11.3%
3	0 0.0%	0 0.0%	316 35.1%	100% 0.0%
ALL	89.0% 11.0%	88.4% 11.6%	99.7% 0.3%	**92.6%** **7.4%**



LDA is better than the estimated classification tree in this problem

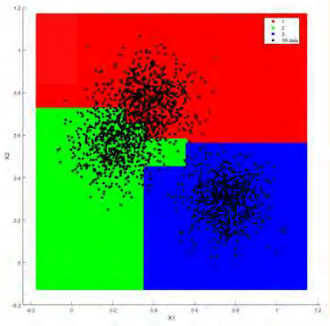
Class. tree

TR Optimal tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	643 30.6%	77 3.7%	0 0.0%	89.3% 10.7%
2	66 3.1%	623 29.7%	1 0.0%	90.3% 9.7%
3	0 0.0%	8 0.4%	682 32.5%	98.8% 1.2%
ALL	90.7% 9.3%	88.0% 12.0%	99.9% 0.1%	**92.8%** **7.2%**

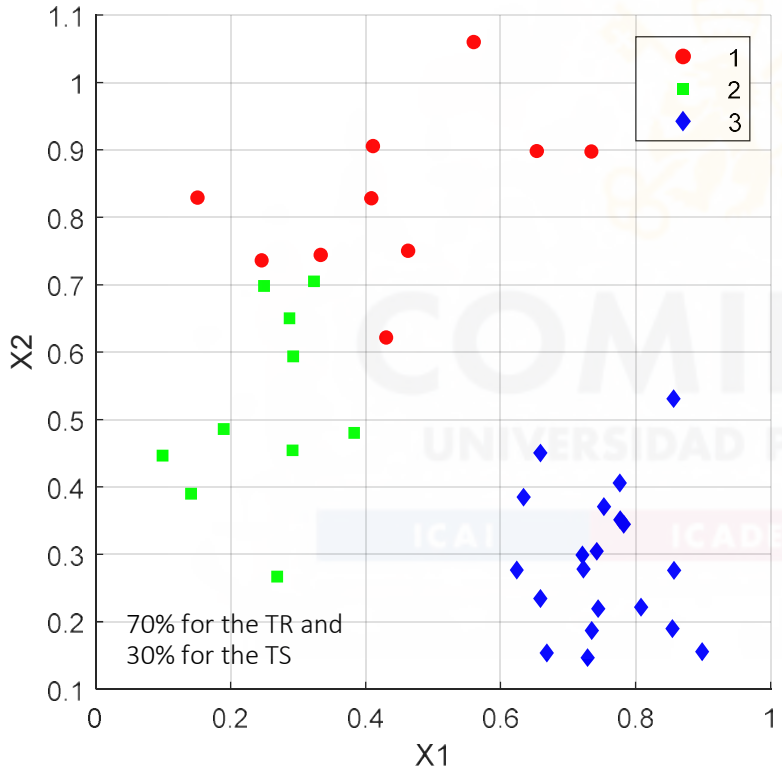
TS Optimal tree: Confusion Matrix

Output Class \ Target Class	1	2	3	ALL
1	261 29.0%	37 4.1%	5 0.6%	86.1% 13.9%
2	30 3.3%	252 28.0%	1 0.1%	89.0% 11.0%
3	0 0.0%	3 0.3%	311 34.6%	99.0% 1.0%
ALL	89.7% 10.3%	86.3% 13.7%	98.1% 1.9%	**91.6%** **8.4%**

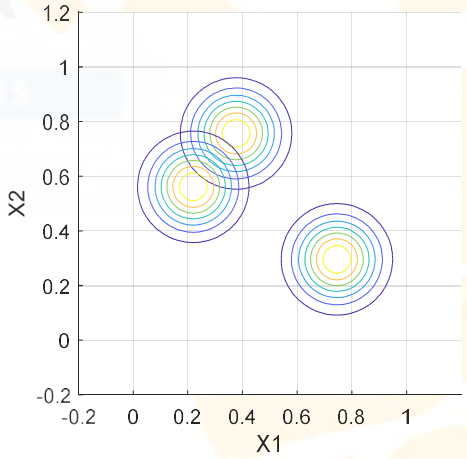


Discriminant Analysis
Illustrative synthetic cases

- C3: 3 classes with very few data (10 + 10 + 20)
  - IDEAL case for Linear Discriminant Analysis

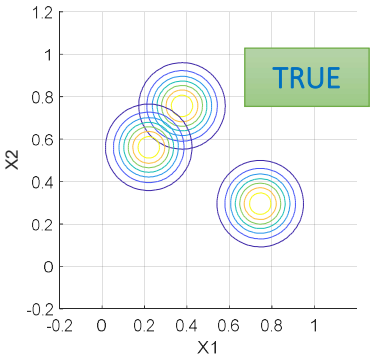


- This dataset has been generated by drawing random data for three different multinormal distributions according to the following TRUE parameters:
  - $\mu_1 = (0.3767 \quad 0.7566)$
  - $\mu_2 = (0.2200 \quad 0.5612)$
  - $\mu_3 = (0.7454 \quad 0.2959)$
  - $\Sigma = \begin{pmatrix} 0.01 & 0.0 \\ 0.0 & 0.01 \end{pmatrix}$

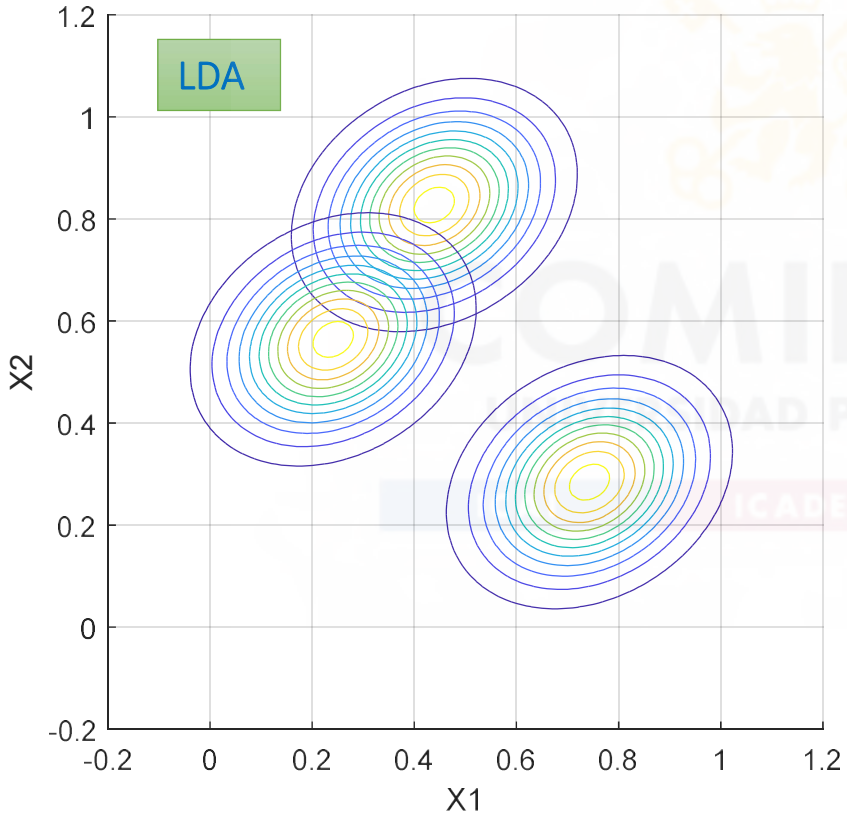


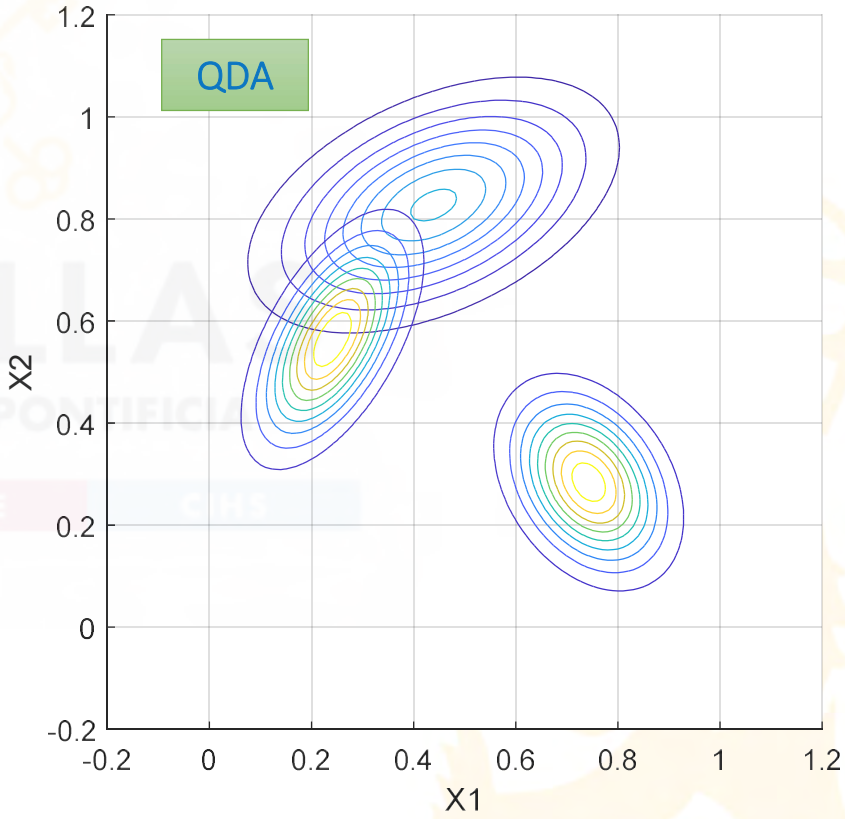
- Class 3 (blue) is easy to split from the other two
- The border between classes 1 and 2 is not very clear

Discriminant Analysis
Illustrative synthetic cases



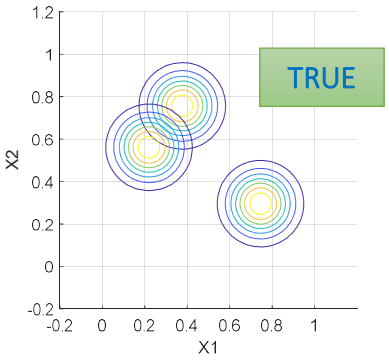
- C3: 3 classes with very few data (10 + 10 + 20)
  - LDA vs. QDA

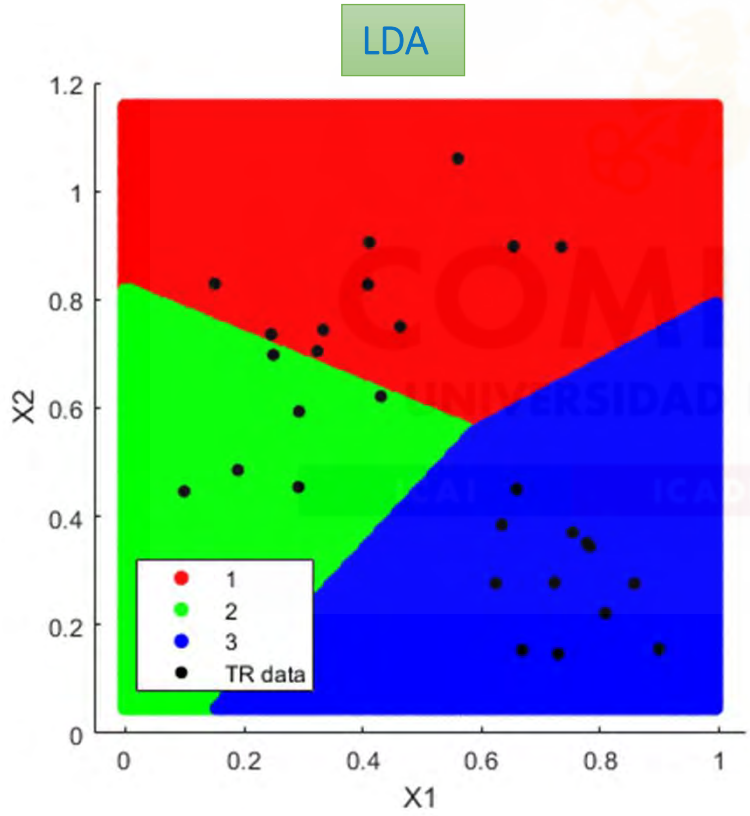


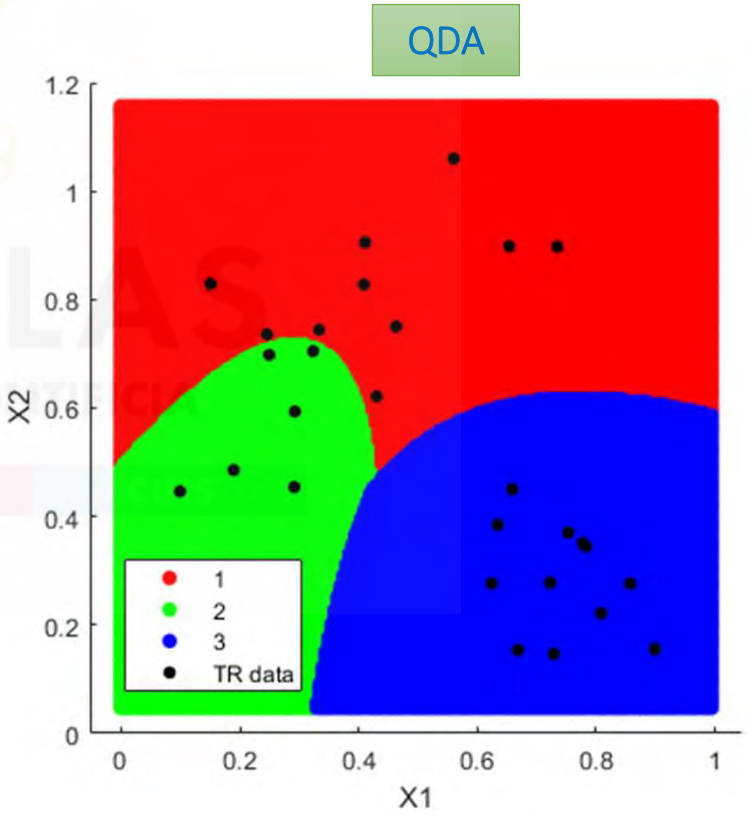


Discriminant Analysis
Illustrative synthetic cases

- C3: 3 classes with very few data (10 + 10 + 20)
  - LDA vs. QDA



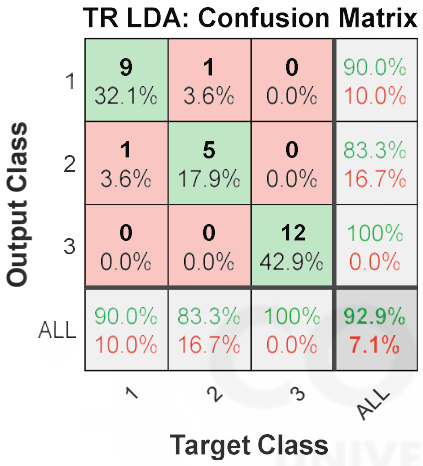


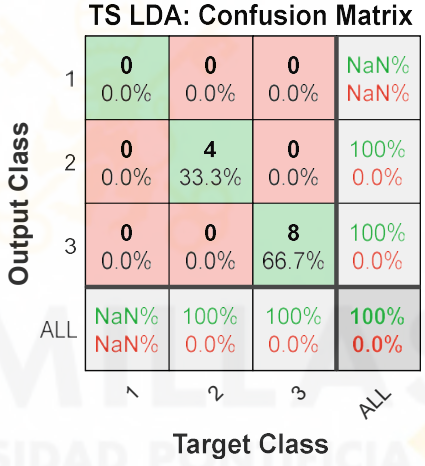


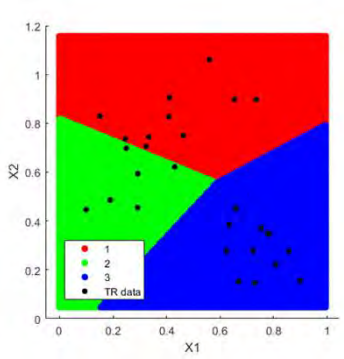
Discriminant Analysis
Illustrative synthetic cases

- C3: 3 classes with very few data (10 + 10 + 20)
  - LDA vs. QDA (confusion matrix, training, and test sets)

LDA

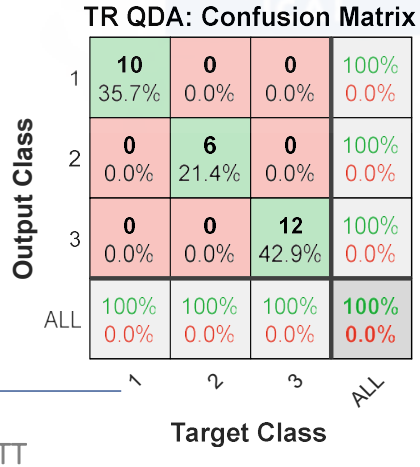


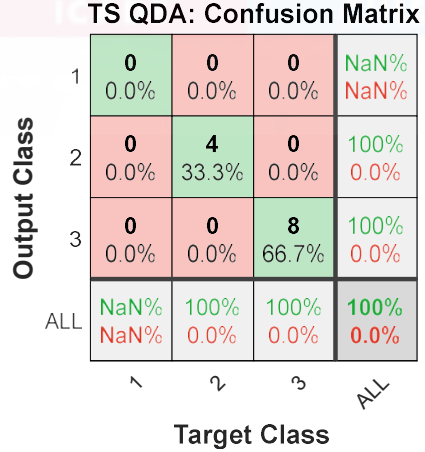


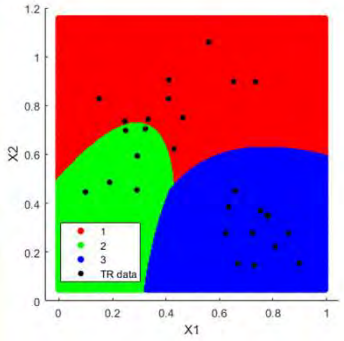


LDA or QDA?

QDA

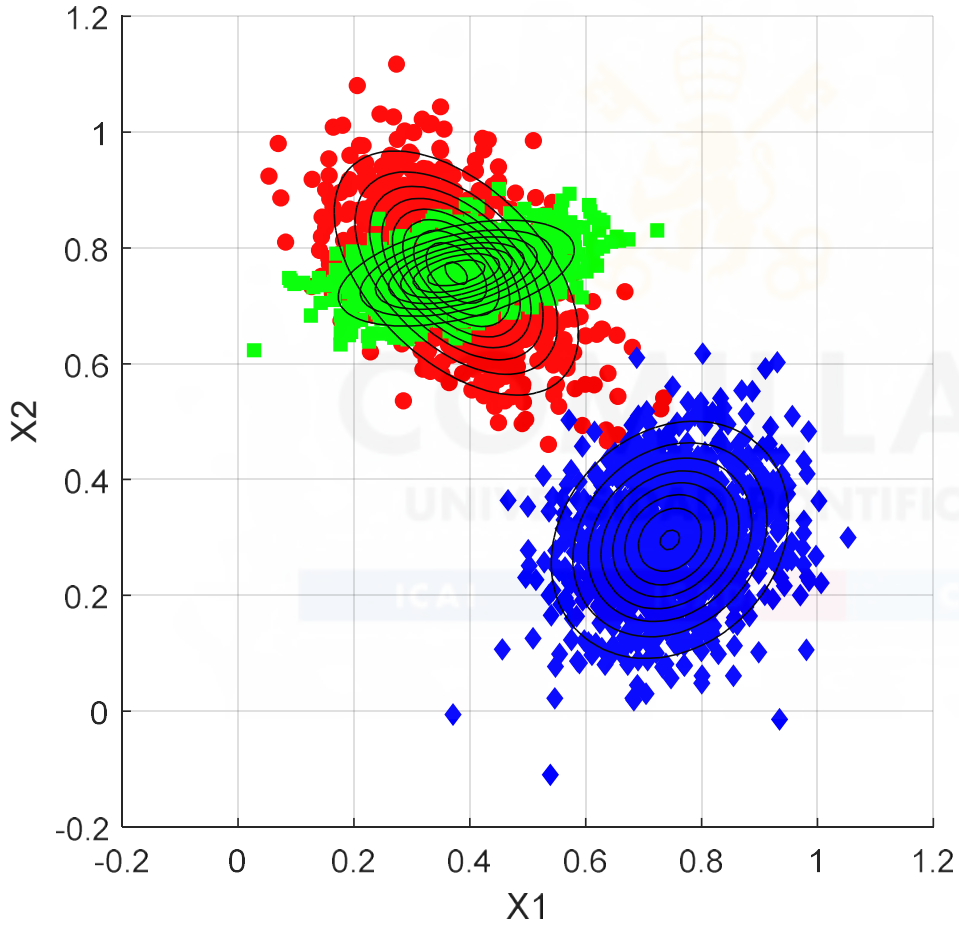




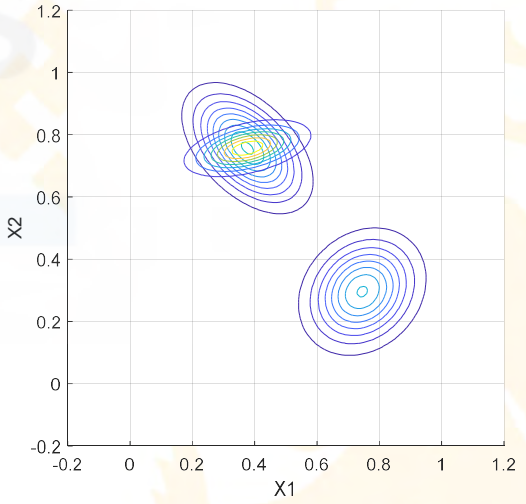


Discriminant Analysis
Illustrative synthetic cases

- C4: 3 classes with many data (3 x 1000)

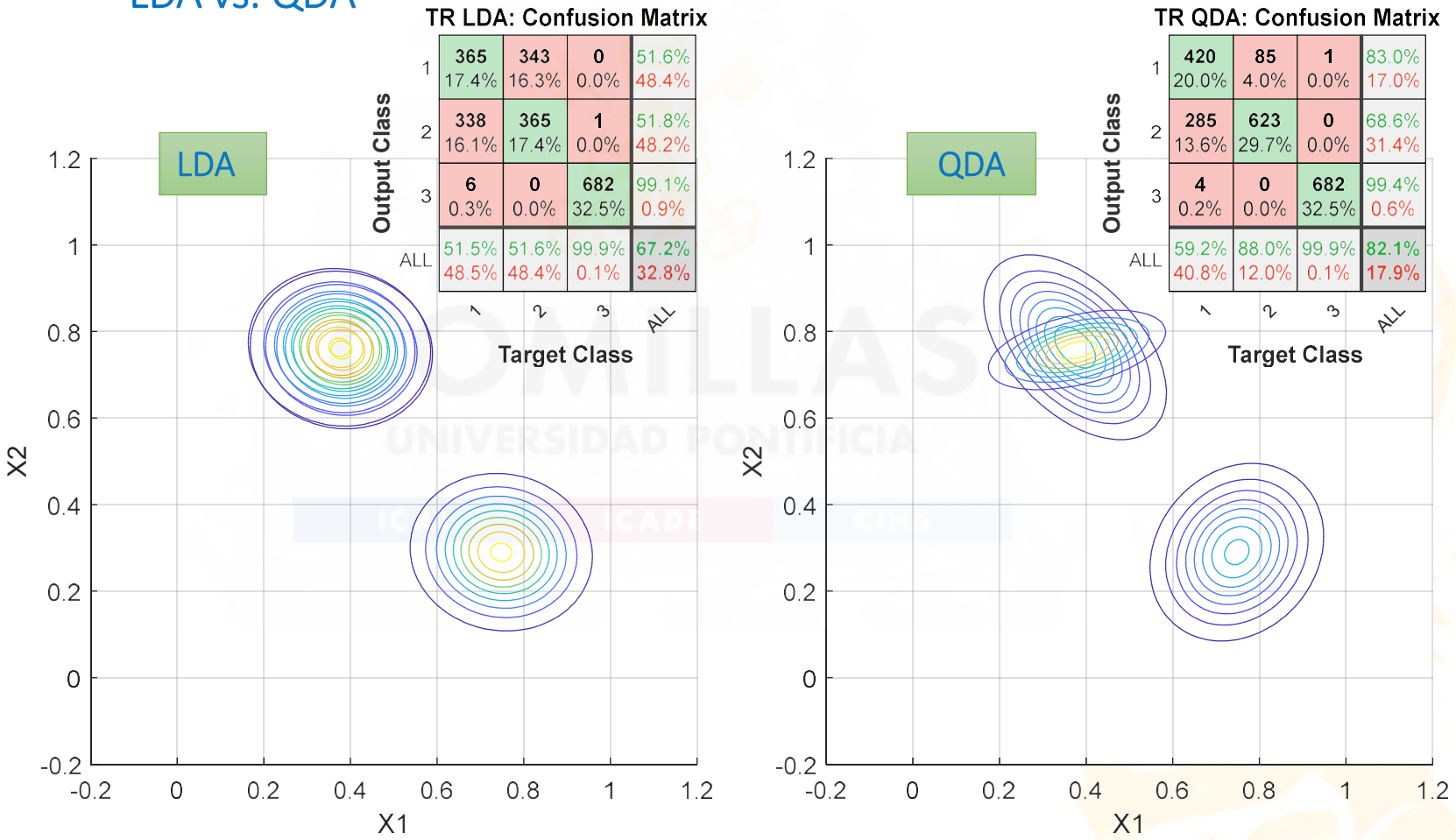


- This dataset has been generated by drawing random data for three different multinormal distributions (see contour lines)
- The green density is larger than the red one



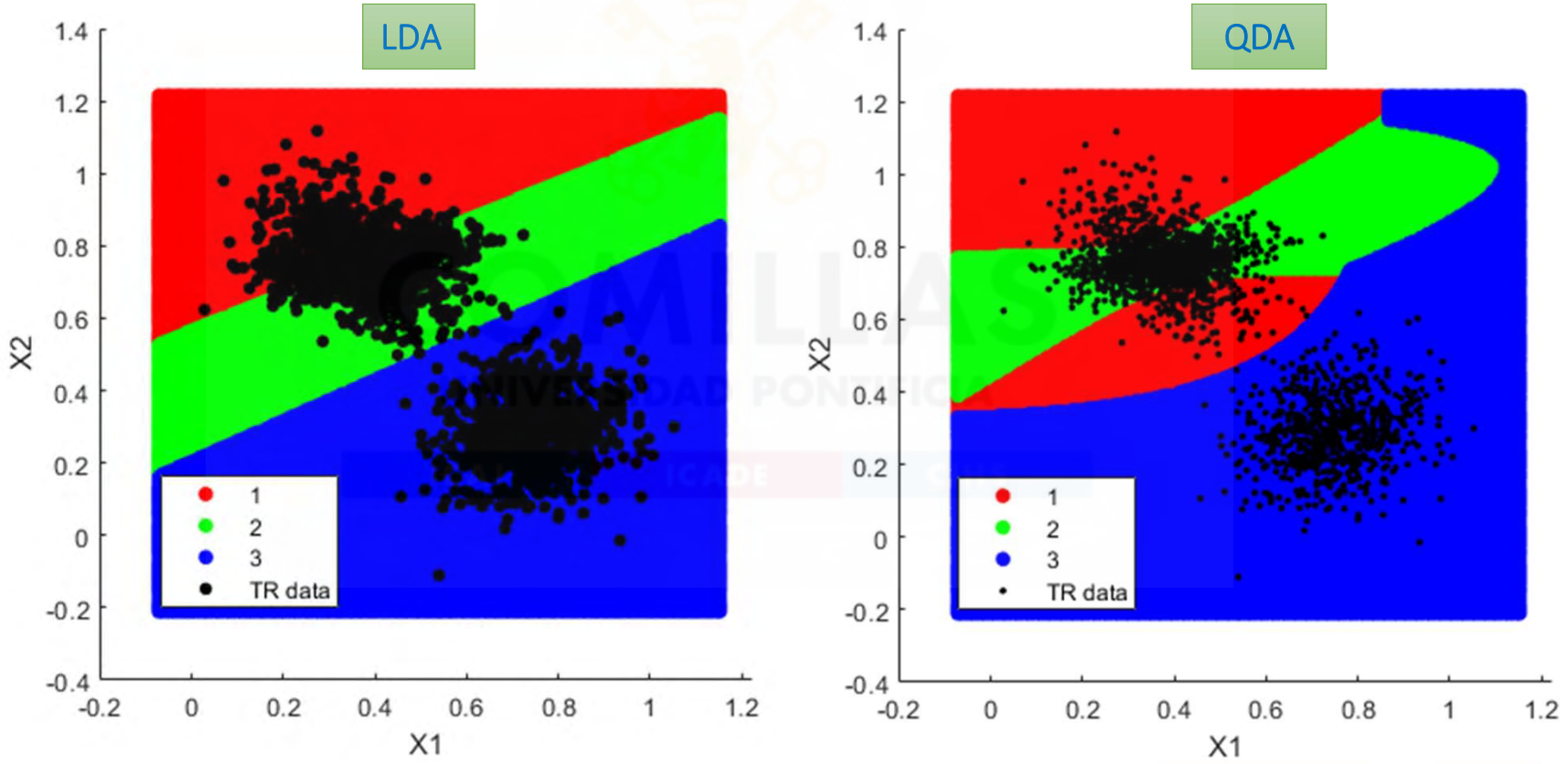
Discriminant Analysis
Illustrative synthetic cases

- C4: 3 classes with many data (3 x 1000)
  - LDA vs. QDA



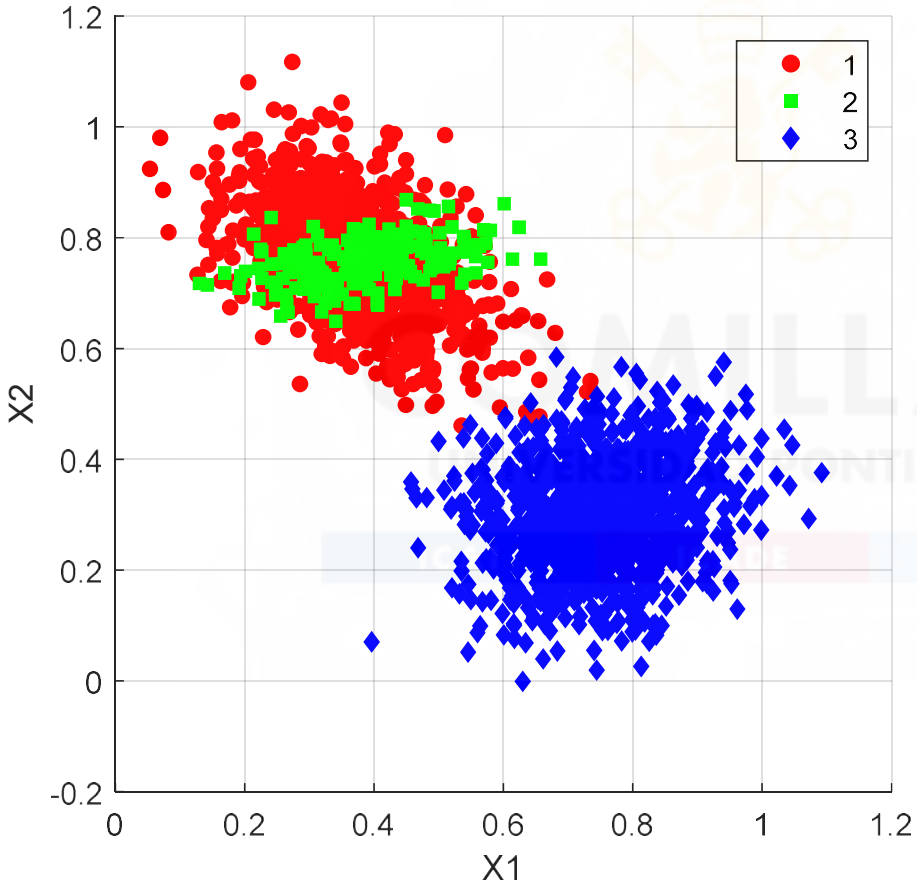
Discriminant Analysis
Illustrative synthetic cases

- C4: 3 classes with many data (3 x 1000)
  - LDA vs. QDA

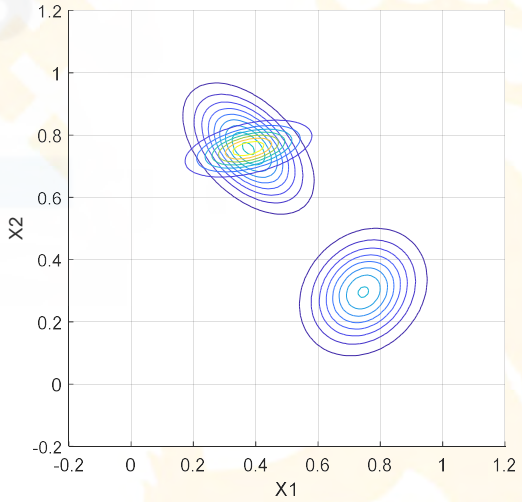


Discriminant Analysis
Illustrative synthetic cases

- C4: 3 classes with many data (1000 + 200 + 1000)

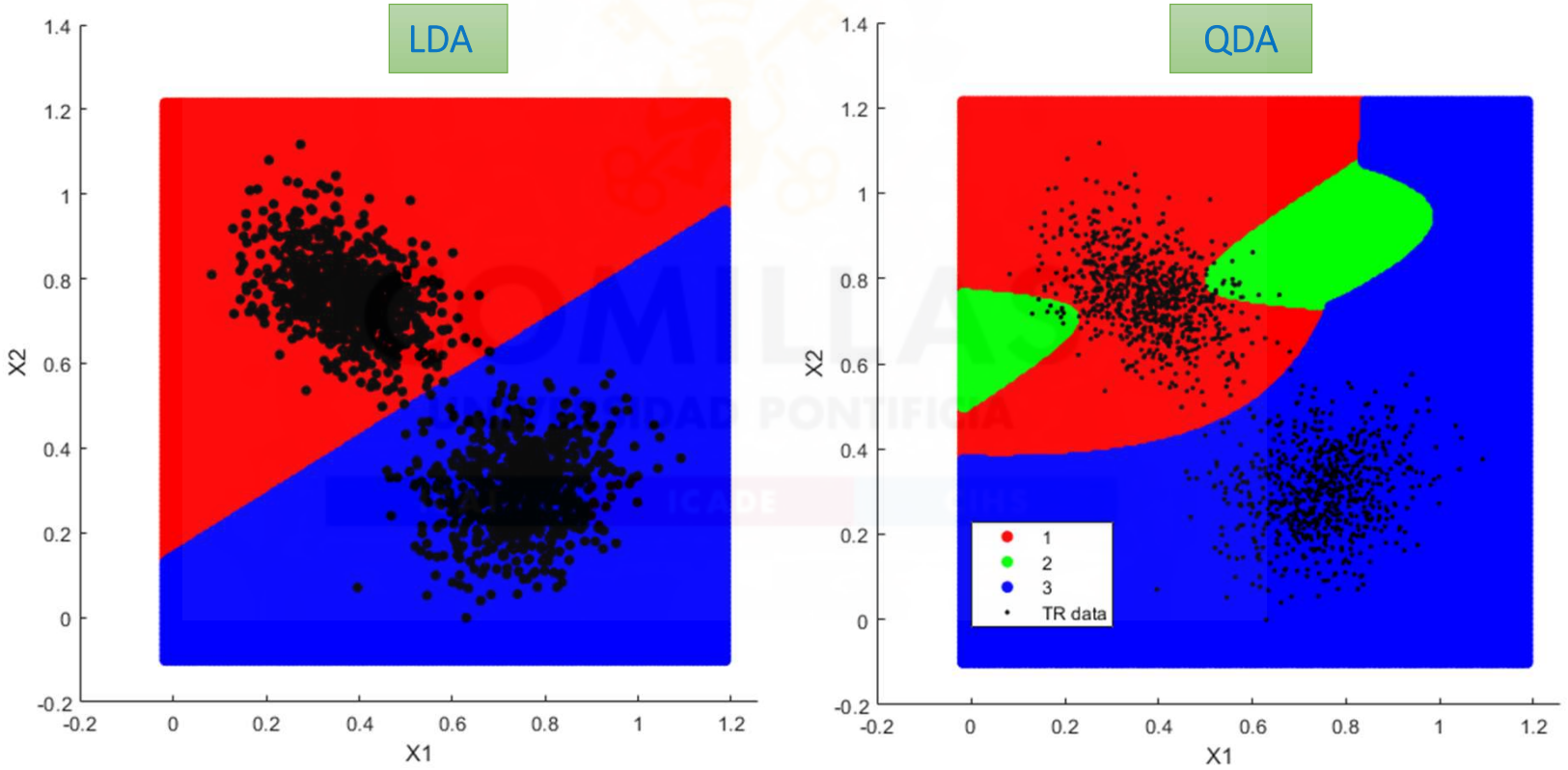


- This dataset has been generated by drawing random data for three different multinormal distributions (see contour lines)
- The green density is larger than the red one, but there are 200 green points



Discriminant Analysis
Illustrative synthetic cases

- C4: 3 classes with many data (1000 + 200 + 1000)
  - LDA vs. QDA



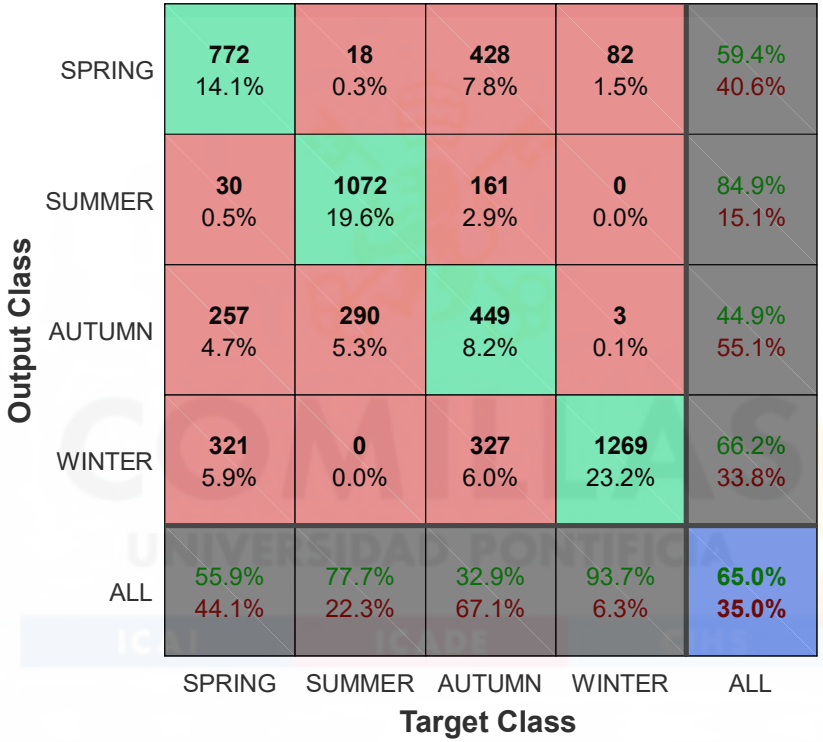
1. Introduction
2. Model complexity vs. generalization error
3. Direct approach: Classification trees
4. Probabilistic approach: Linear Discriminant Analysis
5. Quiz
6. Real examples

5

Quiz

Quiz
Question 1

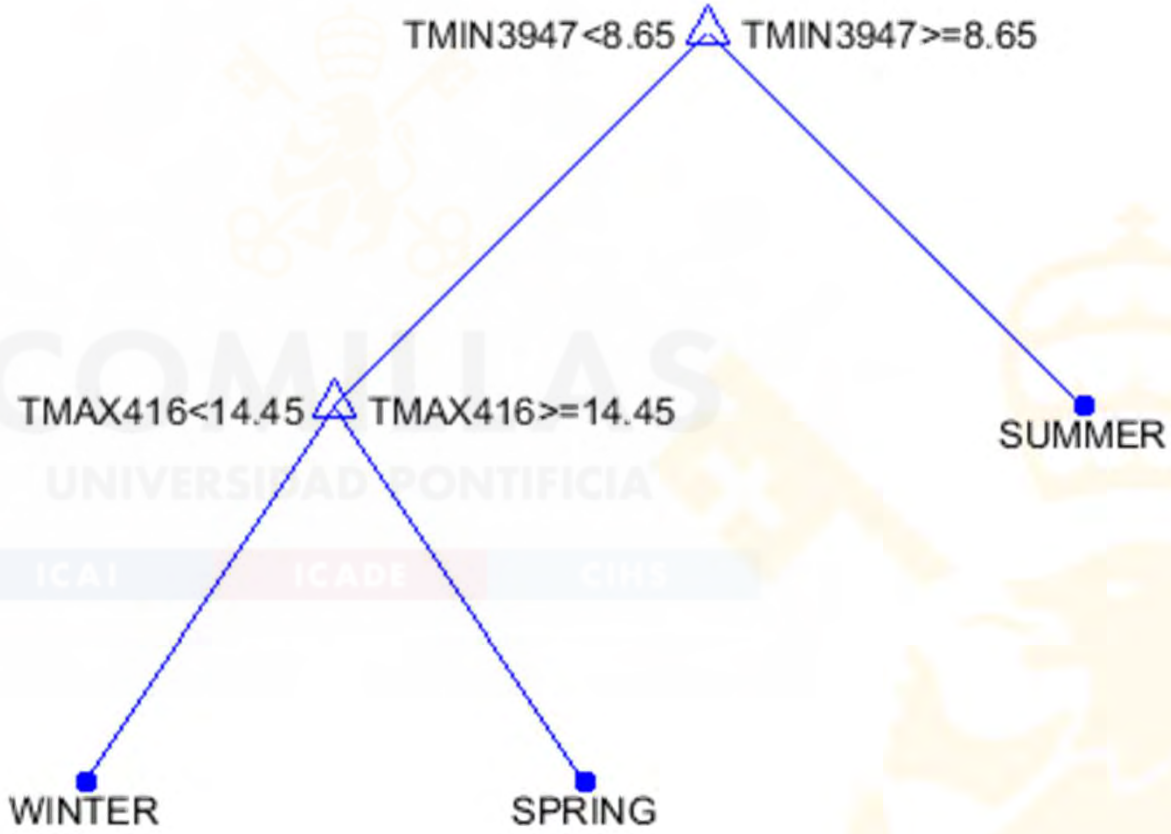
- Según la siguiente matriz de confusión se puede afirmar que



A. El 32.9% del total de las observaciones se han clasificado correctamente como pertenecientes a la clase AUTUMN.
B. La tasa de error de clasificación en la clase SUMMER es del 77.7%.
C. De todas las observaciones de la clase WINTER, un 93.7% de ellas son clasificadas correctamente por el modelo.

Quiz
Question 2

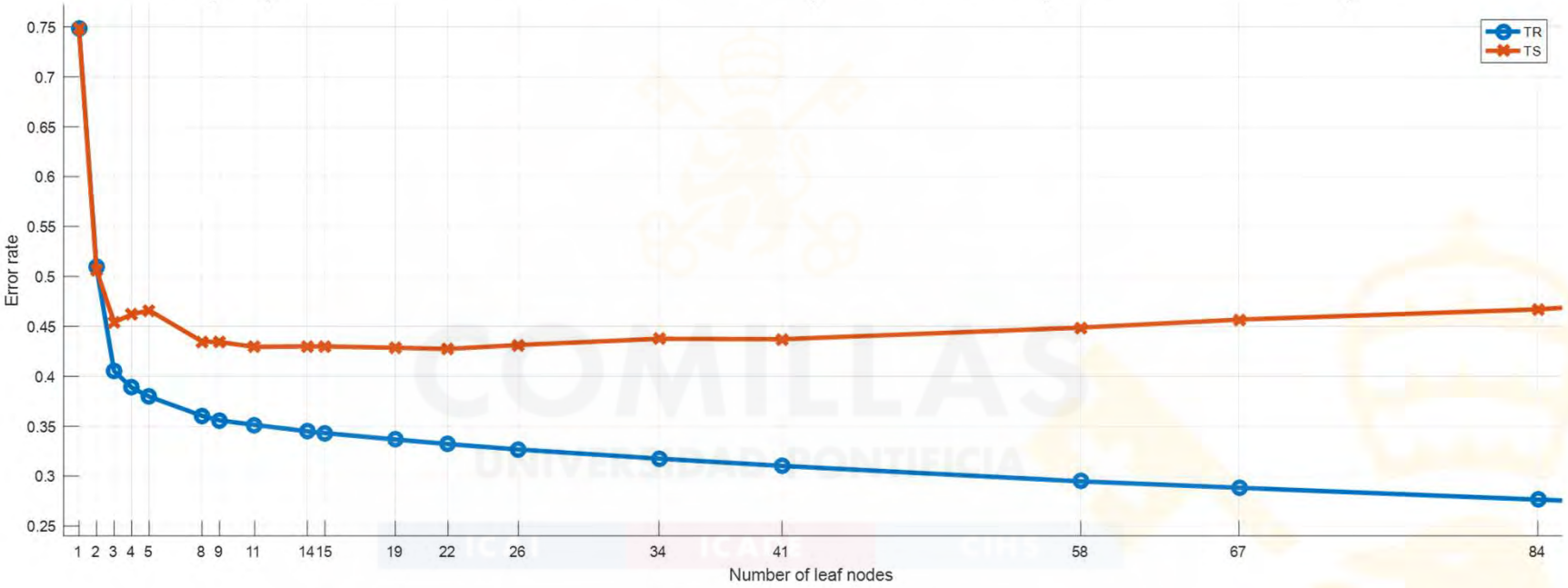
- Según el árbol adjunto, una observación con TMAX416 = 34.6 y TMIN3947 = 7.60 se clasificará como



A. SUMMER.
B. SPRING.
C. WINTER.

Quiz
Question 3

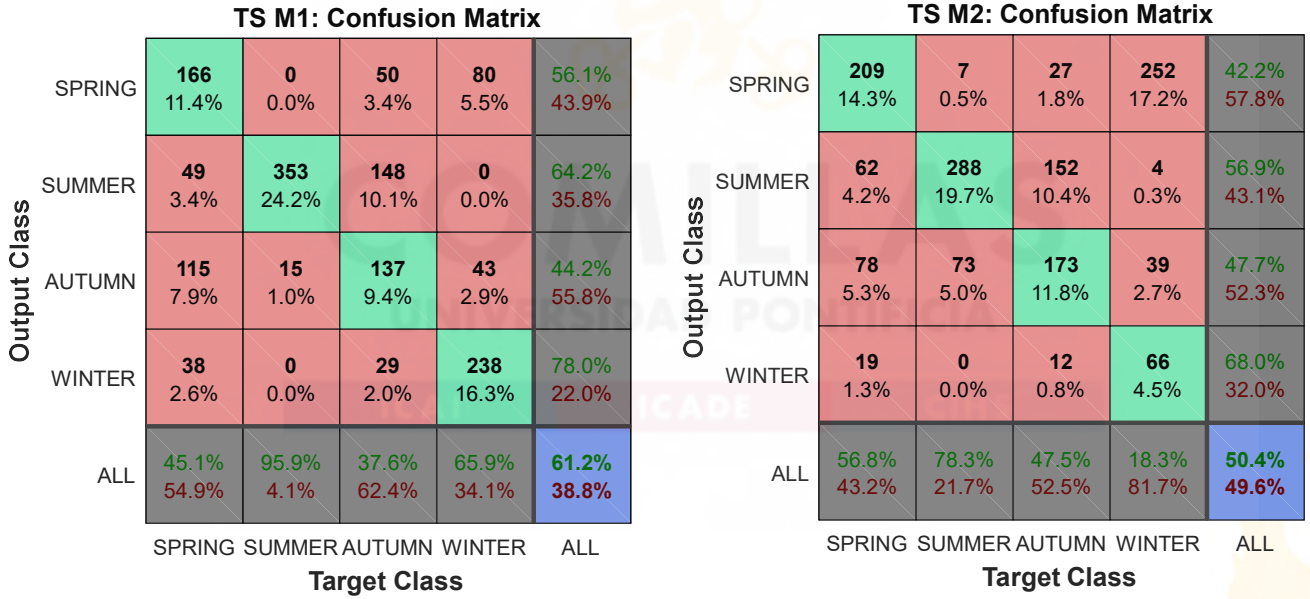
- Considerando las tasas de error en los conjuntos de entrenamiento (TR) y test (TS) de la secuencia de árboles podados se puede afirmar que



A. El árbol con 22 nodos terminales se corresponde con un compromiso razonable entre complejidad y error esperado.
B. El árbol con 2 nodos terminales se corresponde con un compromiso razonable entre complejidad y error esperado.
C. El árbol con 8 nodos terminales se corresponde con un compromiso razonable entre complejidad y error esperado.

Quiz
Question 4

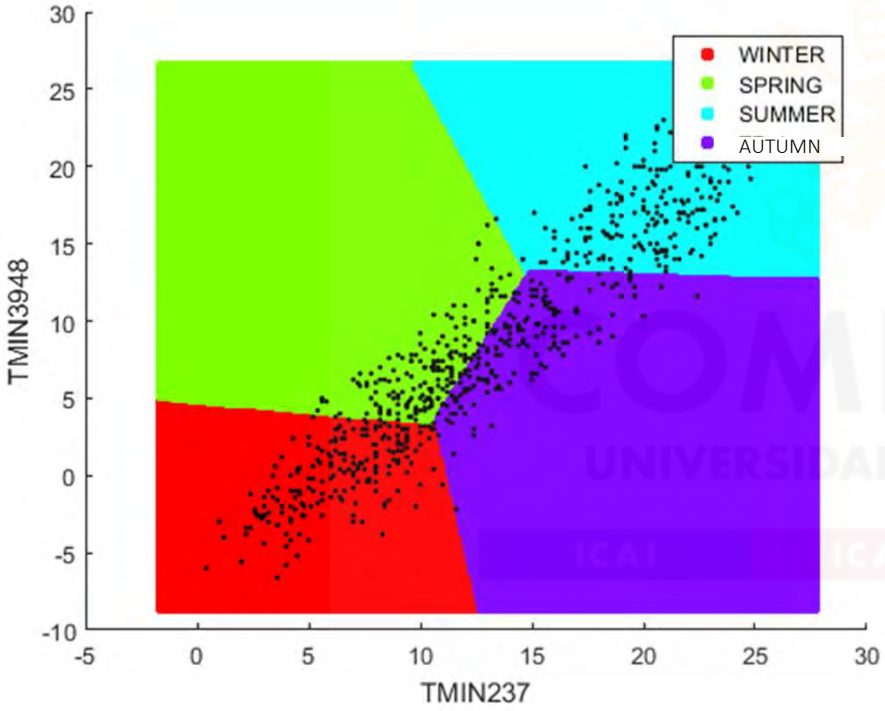
- Se han ajustado dos discriminantes diferentes para estimar la variable 'SEASON':
  - M1: discriminante lineal según 'SEASON ~ TMIN3948 + TMIN237'
  - M2: discriminante cuadrático según 'SEASON ~ TMIN3910 + TMIN417'
- Considerando su complejidad y la matriz de confusión de los modelos en un conjunto de datos de test (TS) se puede afirmar que

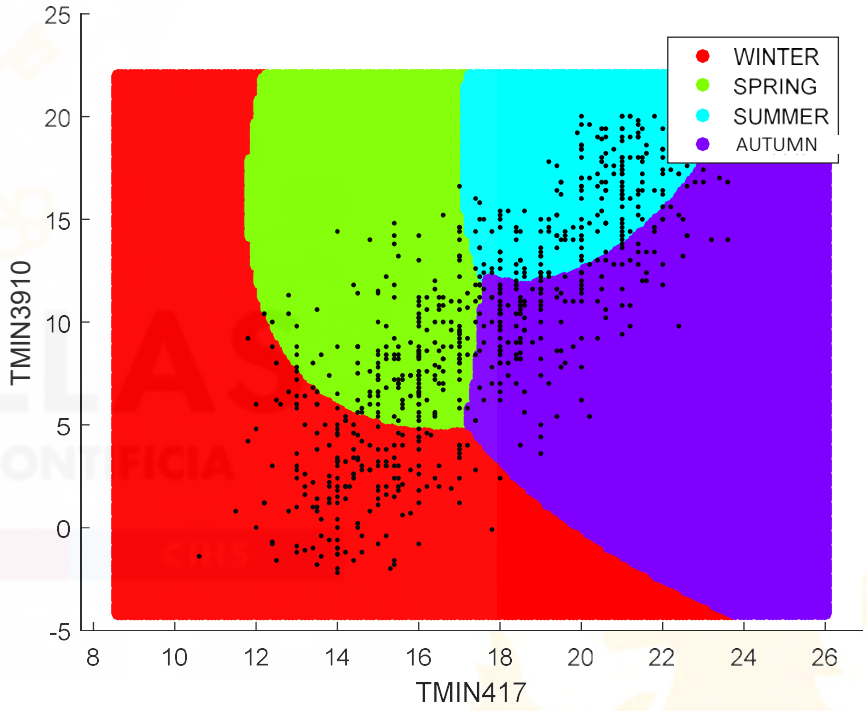


A. M1 es igual de complejo que M2, pero es mejor en términos de error en TS.
B. M1 es mejor que M2 en términos de error en TS, además de ser más sencillo.
C. No tiene sentido la comparación ya que utilizan variables de entrada diferentes.

Quiz
Question 5

- Se han ajustado dos discriminantes diferentes para estimar la variable 'SEASON'. Según la partición del espacio de entrada se puede afirmar que





A. El modelo M1 se corresponde con un discriminante lineal, y el M2 con uno cuadrático.
B. El modelo M1 se corresponde con un discriminante cuadrático, y el M2 con uno lineal.
C. El modelo M1 se corresponde con un discriminante rectilíneo, y el M2 con uno curvilíneo.

Quiz
Question 6

- Selecciona **la técnica más adecuada** a utilizar para cada uno de los siguientes problemas:
  - PRO1: Se quiere estimar el nº de hijos de una pareja a partir de la edad media de la misma y del tipo de coche que tiene. Se tiene información de 100 parejas tanto de su edad, como del nº de hijos y del tipo de coche (valores posibles: ninguno, todoterreno, monovolumen, deportivo, otro).
  - PRO2: Se quiere estimar la nota final en la asignatura de Mecánica Cuántica de un alumno a partir de las notas finales de las asignaturas de Física y Mecánica. Se tienen datos de 500 alumnos, en concreto la nota en las tres asignaturas.

A. Para el problema PRO1 habría que usar un árbol de clasificación, y para PRO2 regresión lineal multivariante.
B. Para el problema PRO1 habría que usar regresión lineal multivariante y para PRO2 un análisis discriminante.
C. Tanto para PRO1 como para PRO2 habría que usar un árbol de clasificación.

Quiz Answers

- Q1-C
- Q2-B
- Q3-C
- Q4-B
- Q5-A
- Q6-A

1. Introduction
2. Model complexity vs. generalization error
3. Direct approach: Classification trees
4. Probabilistic approach: Linear Discriminant Analysis
5. Quiz
6. Real examples

6

Real examples

Real cases
Transmission planning with classification trees

Enhancing Optimal Transmission or Subtransmission Planning by using Decision Trees

J. Peco Jesus.Peco@iit.upco.es
E. F. Sánchez-Úbeda Eugenio.Sanchez@iit.upco.es
T. Gómez, Member, IEEE Tomas.Gomez@iit.upco.es
Universidad Pontificia Comillas, Instituto de Investigación Técnológica (IIT)
Alberto Aguilera 23, 28015 Madrid – SPAIN

Abstract – Due to the large size of electric power systems, there is a very high computational burden when obtaining the optimum network by using classical optimization techniques. Several authors have used heuristics and/or sensitivities in order to guide the search of optimal network investments. This paper proposes an Automatic Learning approach in order to decide whether a network change will improve the overall costs or not. More specifically, Decision Trees methods are used to identify a set of simple and reliable rules which combine criteria based on both heuristics and sensitivities. These decision trees are integrated in a subtransmission planning tool, improving dramatically both the "optimality" of the resultant network and the computational time.

Keywords: Transmission planning, planning rules, automatic learning, decision trees, genetic algorithms, data

papers [5, 7, 8]; and (ii) less effort is needed to generate the set of expansion alternatives. AL methods have been successfully applied to various problems (e.g. see [9, 10]).

Three steps should be fulfilled when applying the proposed methodology. Firstly, the types of network changes to efficiently explore the search space should be identified. Secondly, a set of parameters (network attributes) describing relevant aspects of the network state related to a particular network change must be determined. Finally, a set of rules has to be obtained for each type of network change. For transmission planning purposes, we have identified five types of changes and selected 82 candidate attributes based on heuristics and sensitivities.

The planning rules are automatically extracted by

J. Peco, E.F. Sánchez-Úbeda, and T. Gómez Enhancing Optimal Transmission or Subtransmission Planning by using Decision Trees IEEE Budapest Power Tech, Budapest, Hungary, August 1999

Real cases
Transmission planning with classification trees

The proposed candidate attributes comprise different aspects (see Fig. 1): (i) topological, related to the building facilities (e.g. lines, transformers); (ii) flows and load levels; (iii) reliability indexes; (iv) sensitivities of total costs according to a given network change; and (v) heuristic ratios. We maintain that this set of attributes is valid for most of the practical transmission and subtransmission planning problems.

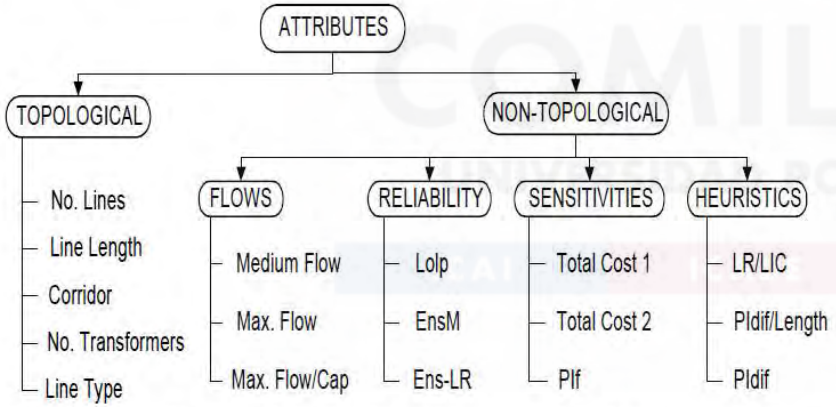


Fig. 1. Simplified scheme of the proposed candidate attributes.

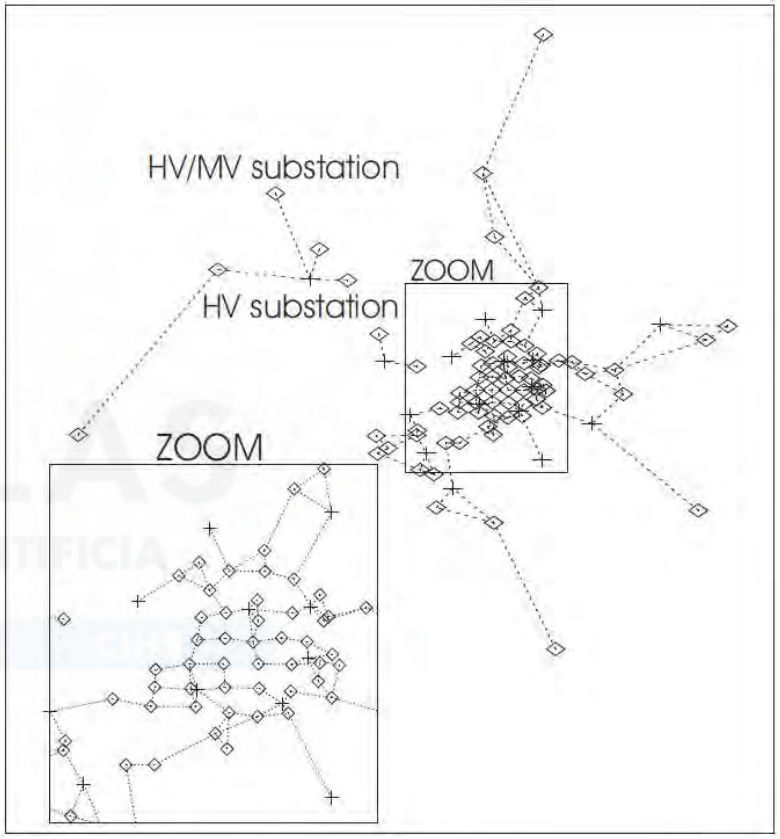


Fig. 8. Obtained subtransmission network.

Real cases
Transmission planning with classification trees

1) Decision Tree for Installing a New Line: The main reason for installing a new line is to increase the network's reliability. According to Fig. 4, the main network attribute is *PIdif*. Other candidate attributes regarded as interesting, such as the LOLP index, where neglected by the DT. Therefore, in network planning, it seems that the spot price is a better estimator of the reliability of a bus. The spot price of a bus considers both the network losses from the generator to that bus, and the reliability of that bus. If the difference between the spot prices across the line is negligible, then installing a new line probably would neither improve the reliability nor the network losses. On the other hand, if this difference is significant and one of both ends of the path is disconnected, it is clear that installing a new line will be worthy.

Finally, note that this tree is effective because it classifies 94.92% of the test cases correctly using only three tests.

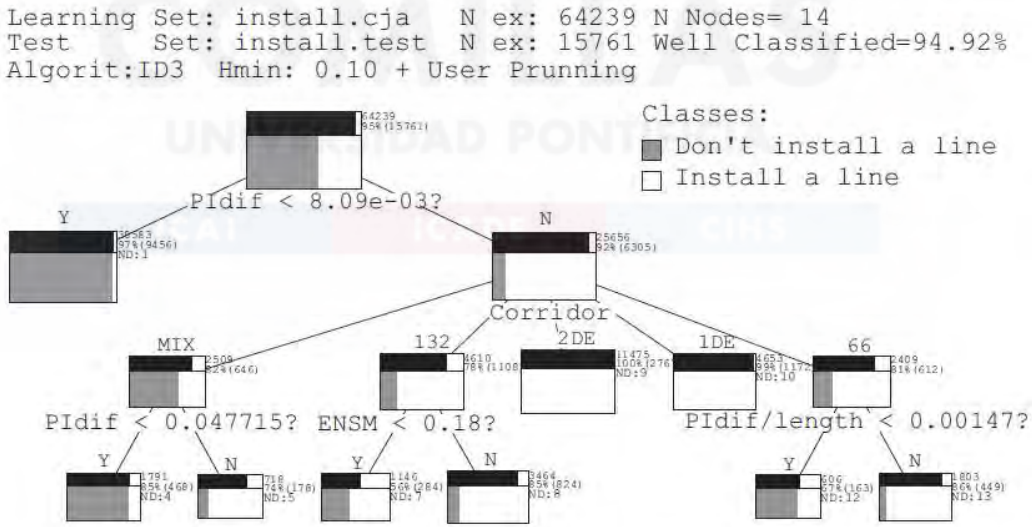


Fig. 4. The decision tree for installing a new line.

A

Dictionary

Classification
Dictionary

- **Bivariate Gaussian distribution** – Distribución binormal (2 variables)
- **Feature** – característica, variable de entrada
- **Multivariate distribution** – Distribución multivariada (varias entradas)
- **Overfitting** – sobreaprendizaje
- **Prior probability** – Probabilidad a priori
- **Split** – Corte, división
- **Threshold** – Umbral
- **Training** – Entrenamiento, ajuste

Thank you for your attention

Eugenio Sánchez Úbeda