# Introduction to Data Science 2

# Overview

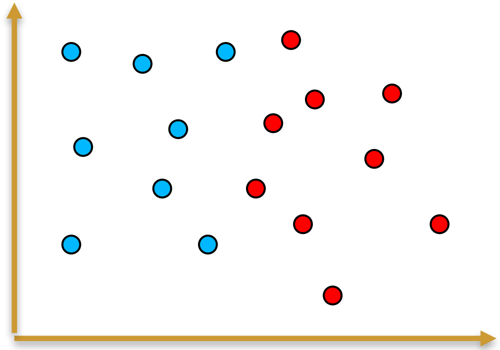
- Classification
  - k-Nearest Neighbour classifier
  - Decision boundaries
- Decision Trees
- Trump versus Clinton

# Classification

- Things are represented by feature vectors (points)

- Each thing (point) has a class label, which we represent by colours

- Examples of classification tasks:
  - Stock Market features —> BUY/SELL?
  - BLOGpost features —> MALE/FEMALE?
  - Fruit features —> ORANGE/APPLE/KIWI?
  - Image features —> INDOOR/OUTDOOR?



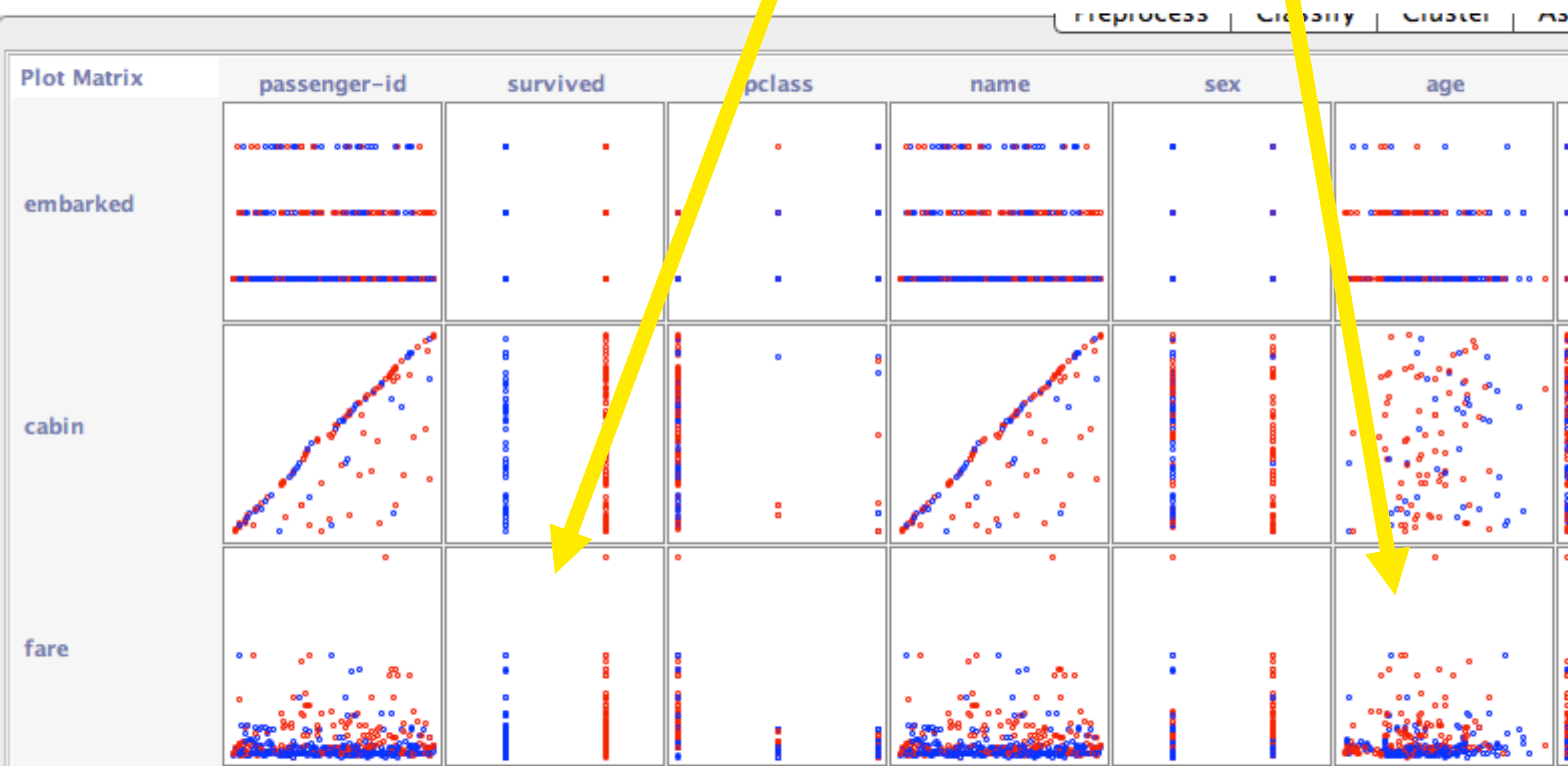
Strongly idealised!
We almost never see
such clear separation
in real datasets.

TILBURG UNIVERSITY

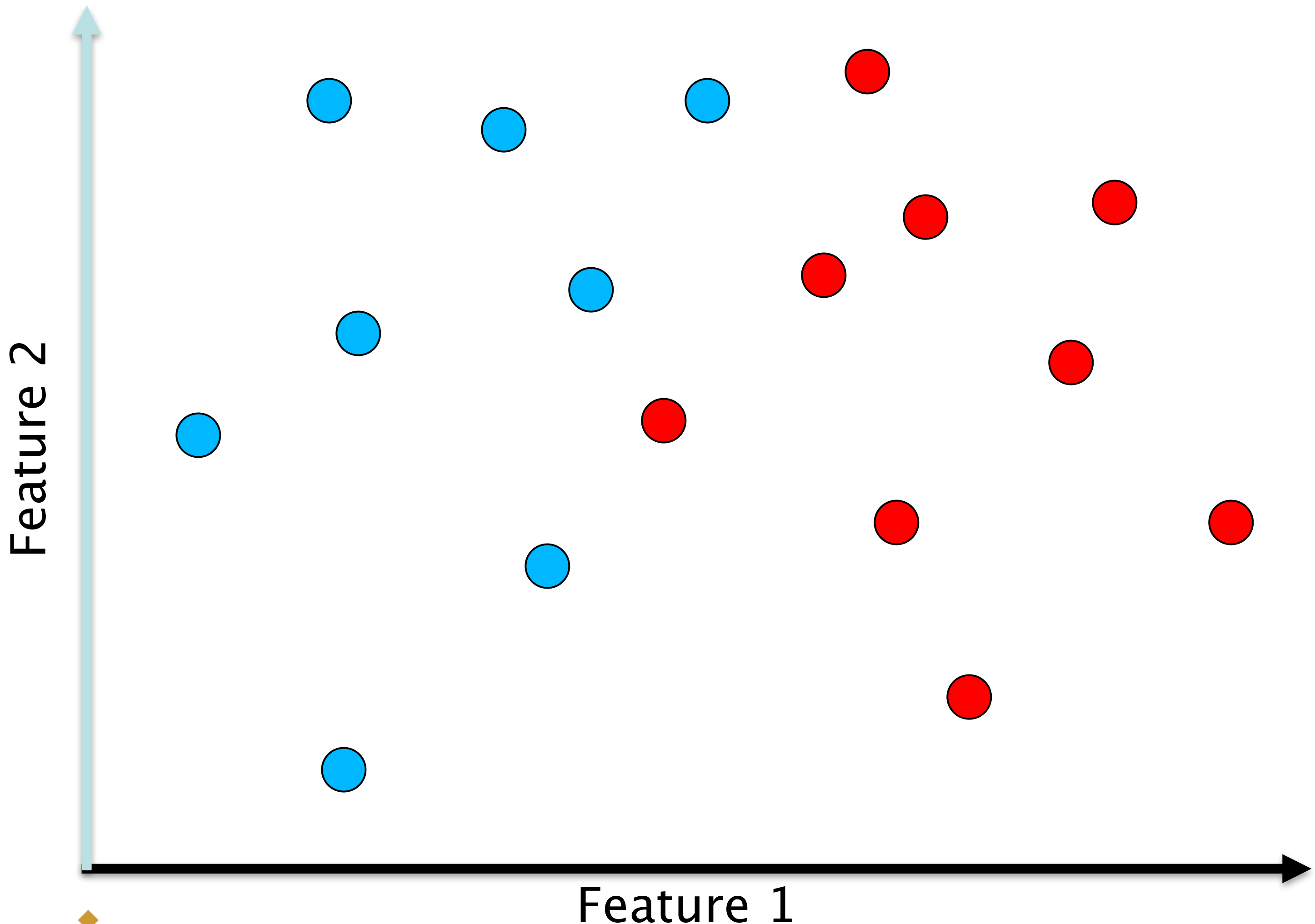
# Scatterplots (WEKA  Visualise - Titanic.arff)

Please note that we can have two types of scatterplots:

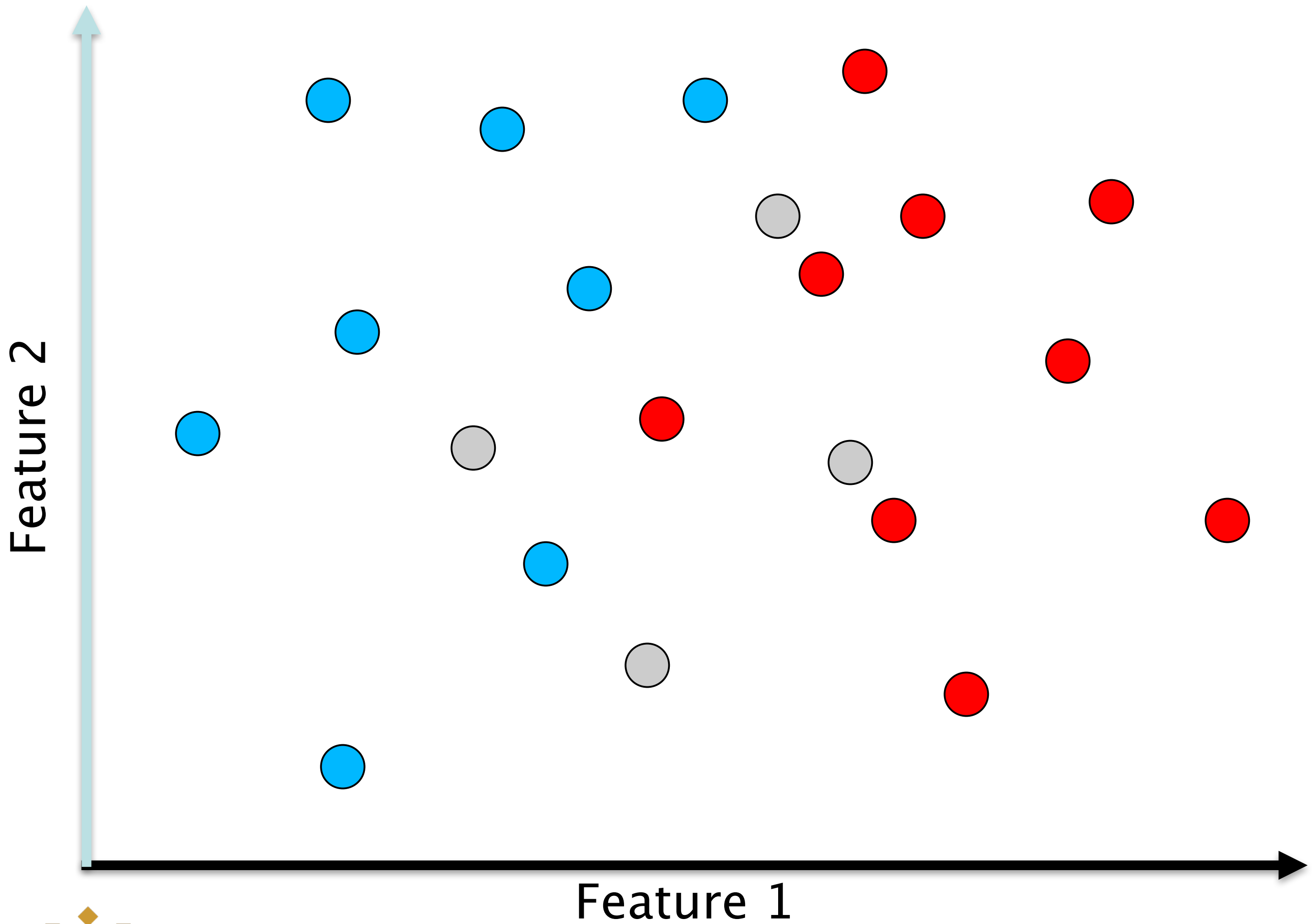
- feature against feature (e.g., **age** against **fare**)
- feature against label (e.g., **survived** against **fare**)



# 2 classes (blue and red) defined by 2 features



# What are the class labels assigned to the grey instances?

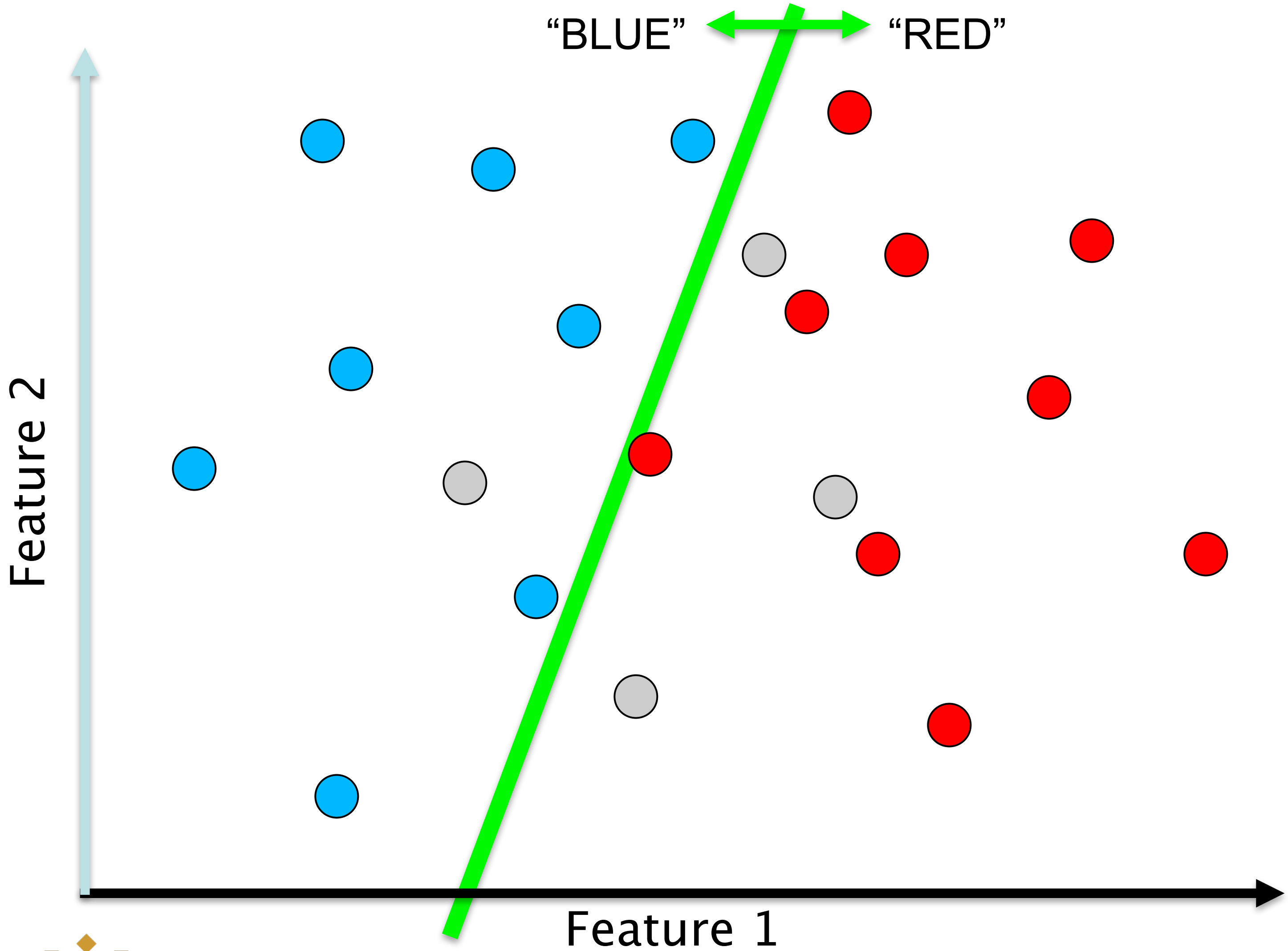



Tilburg University

# Decision Boundaries

- Classifiers are trained on the dataset (labelled data points) and automatically “draw” a decision boundary between the two classes
- The decision boundary can be a straight line (“stiff”) or a wiggly line (“flexible”)
- The decision boundary is considered to be a model of the separation between the two classes
- The model is induced from the data

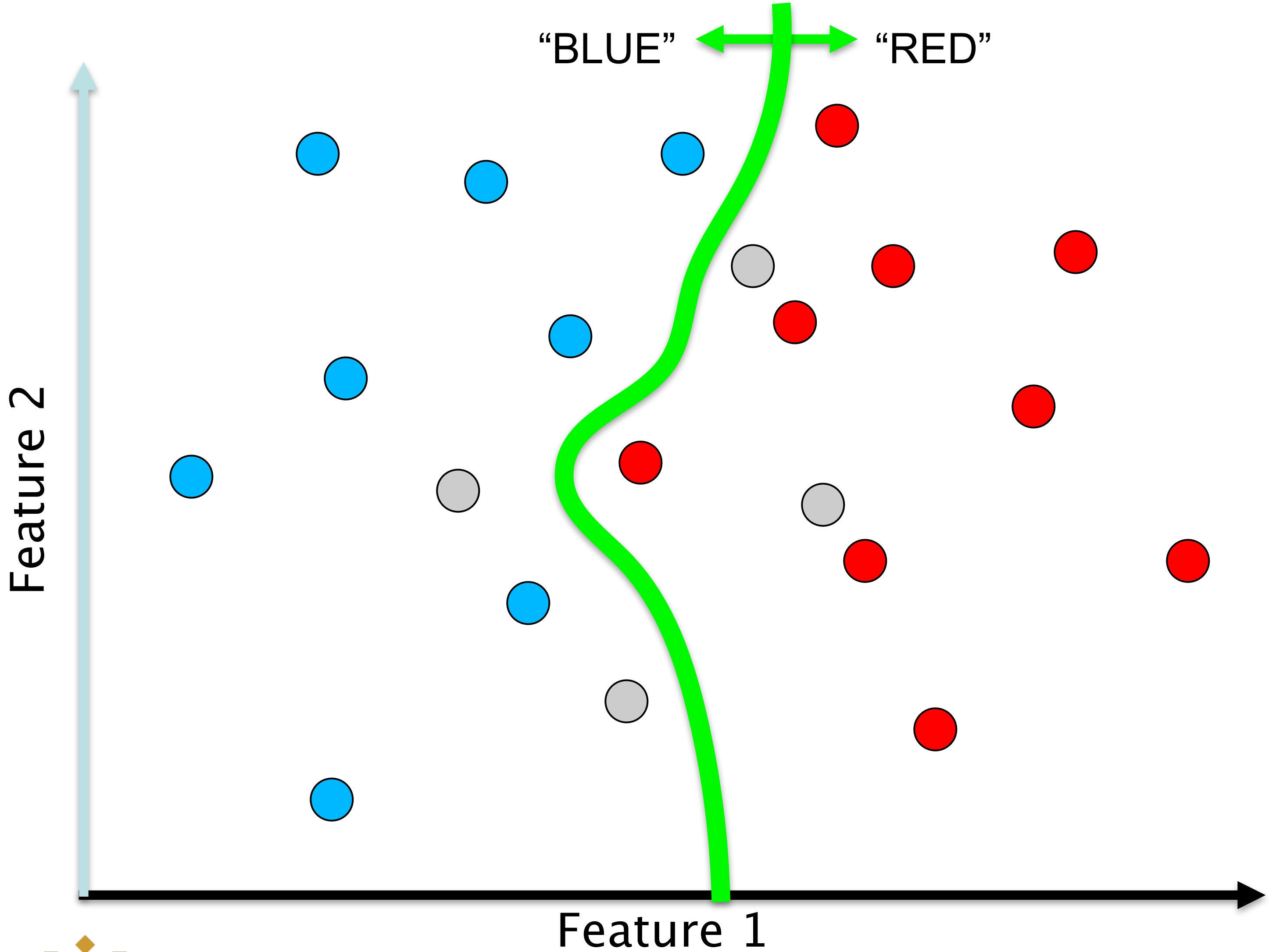
# Decision Boundaries: linear decision boundary




TILBURG UNIVERSITY

simple model

# Decision Boundaries: nonlinear decision boundary

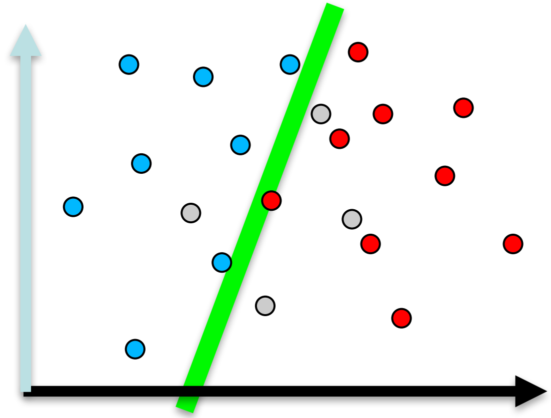



TILBURG UNIVERSITY

complex model

# PLEASE NOTE!
## Classification versus Regression (related to 2 types of scatterplots)

- In classification, the model induced from the data defines a decision boundary that **separates** the data described by 2 features into 2 classes (e.g., *cats* versus *dogs*) or more.



separates the data

- In regression, the model induced from the data **fits** the data to describe the relation between 2 features or between a feature (e.g., *furriness*) and the label (e.g., *cuteness*)

NOT NOW

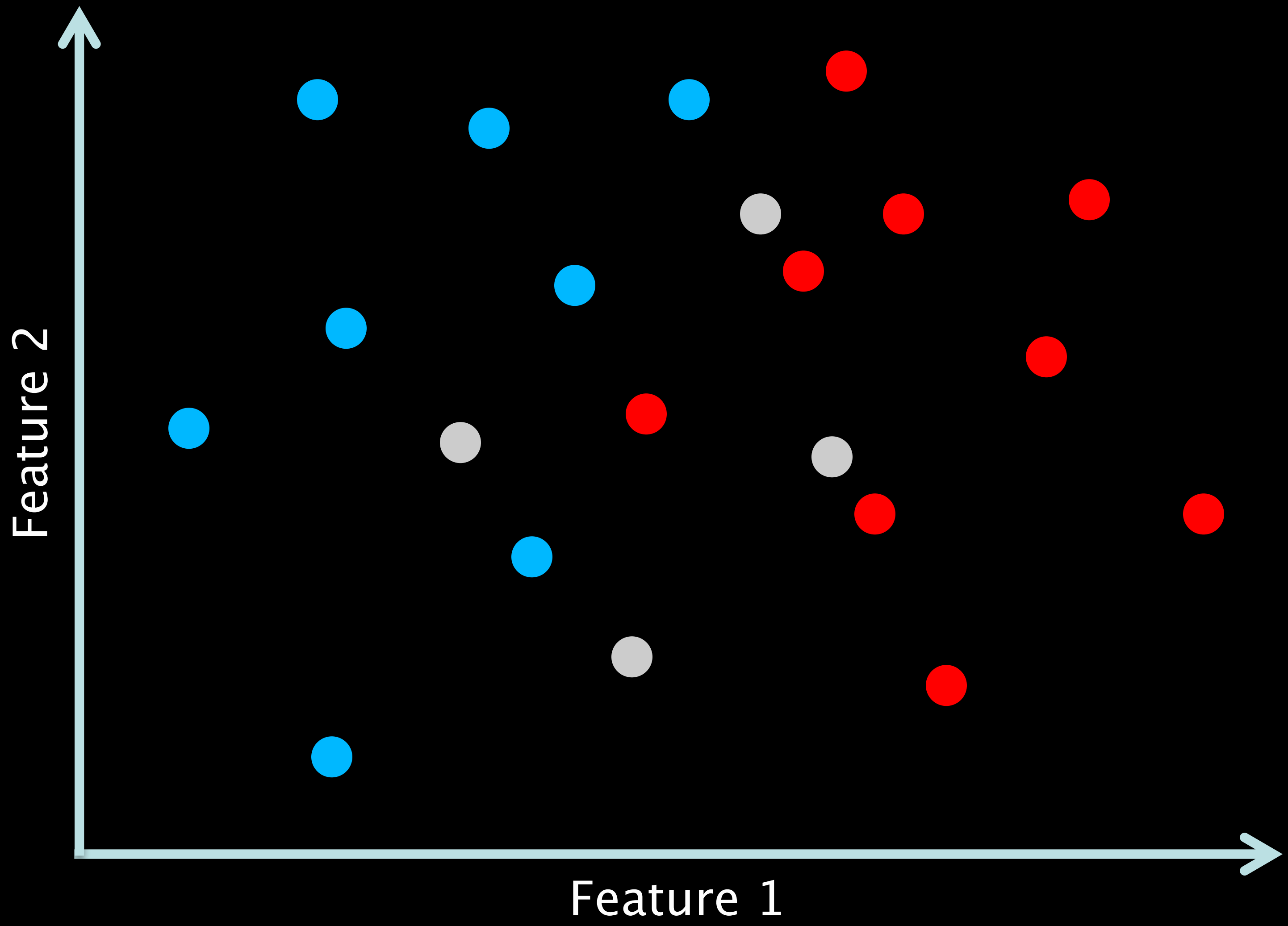


fits the data

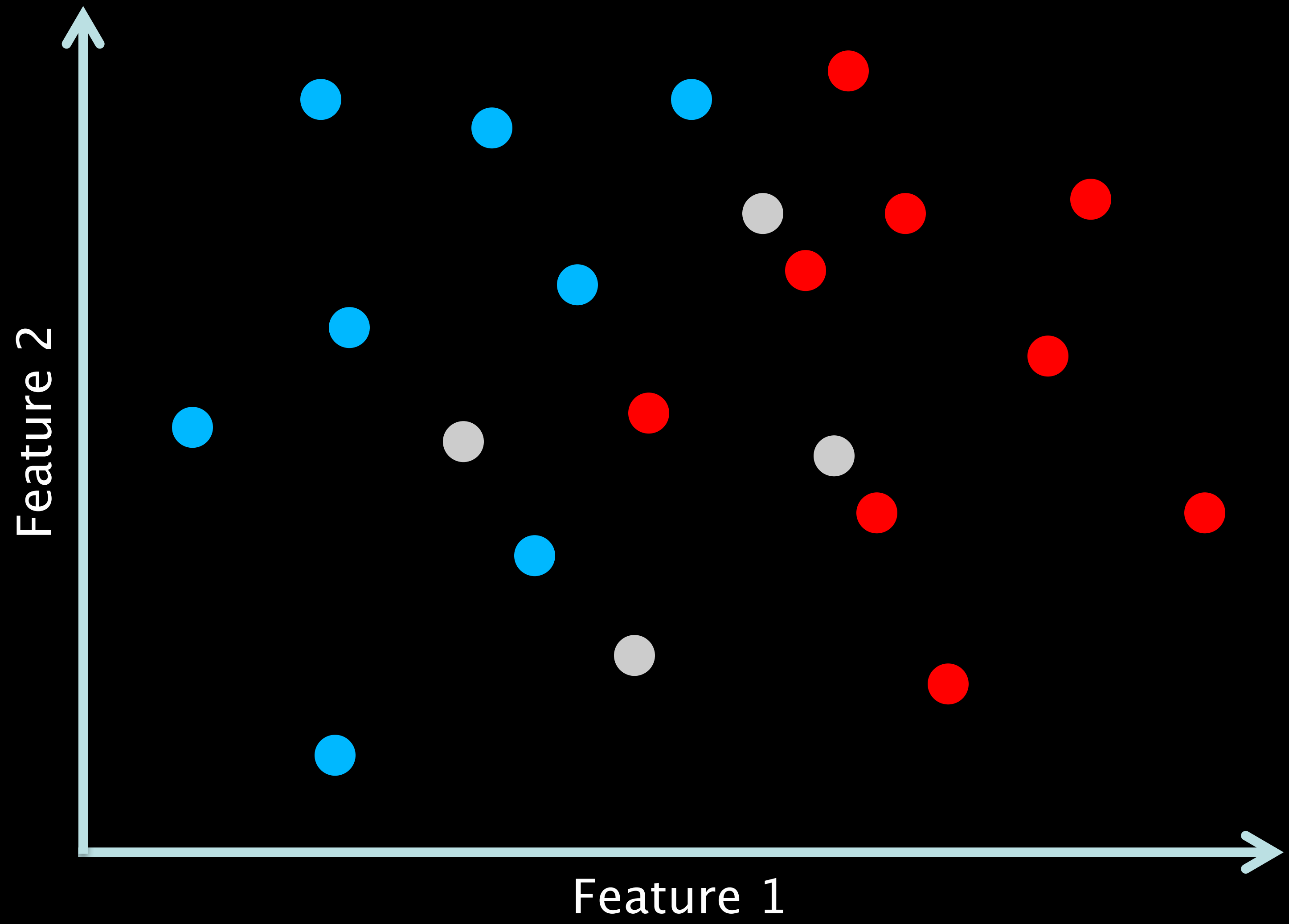
# k-Nearest Neigbour classifier

IBk (“lazy learner”)

# Two Classes (blue & red)



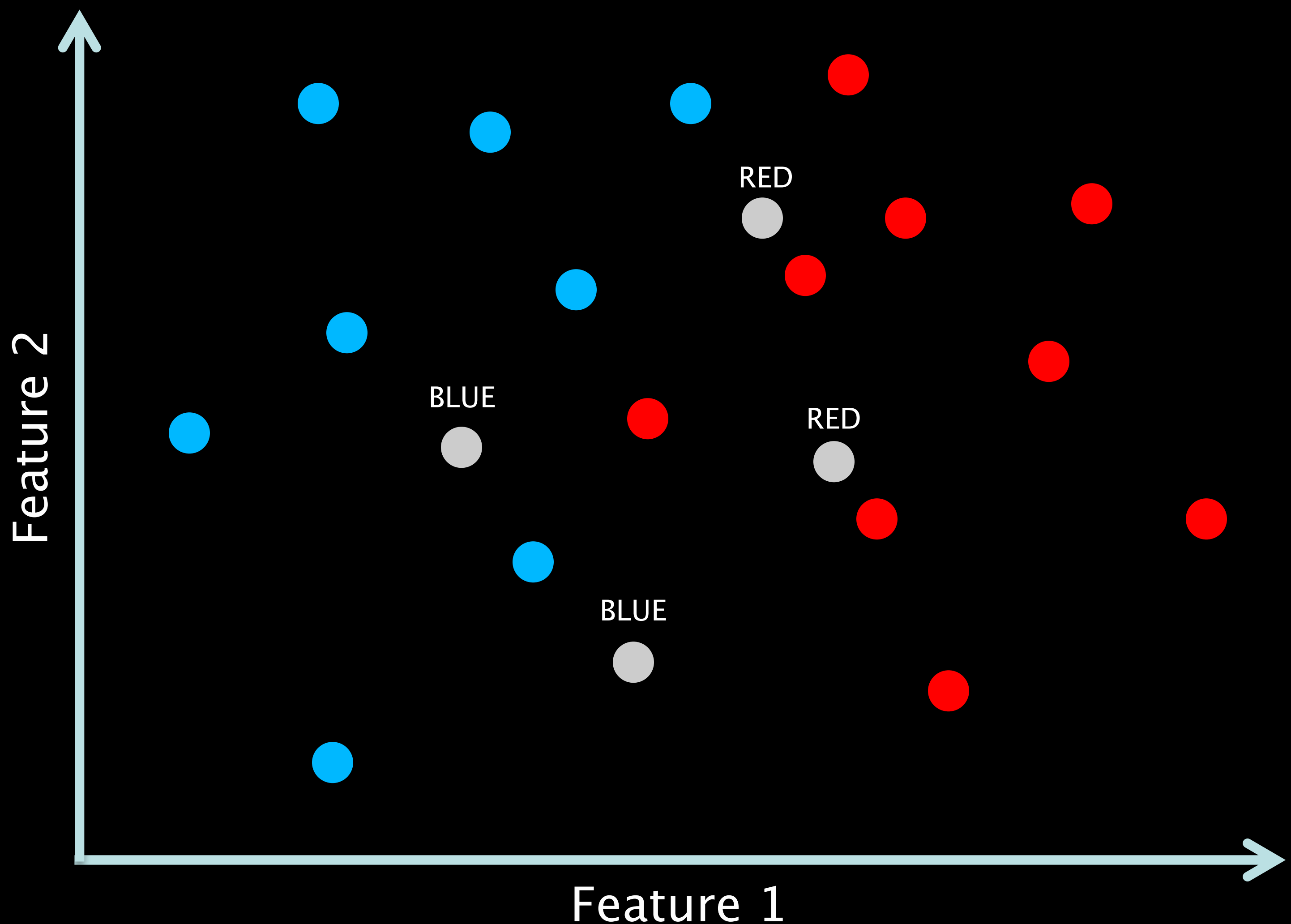
# What are the class labels of the white dots?



# Nearest-neighbour classifier

- Given a set of labeled instances (training set), new instances (test set) are classified according to their nearest labeled neighbour

# Nearest Neighbour Classification ("estimates")



# Evaluating 1-NN performance

or actually: evaluating classifier performance does the estimated color agree with the actual color?

actual color

test instance


| # test | c | ok? ? |
|---|---|---|
| 1 | R | 0 |
| 2 | B | 1 |
| 3 | B | 1 |
| 4 | B | 0 |
| 5 | R | 0 |
| 6 | R | 1 |
| 7 | R | 1 |
| 8 | B | 1 |
| 9 | R | 1 |
| 10 | B | 0 |

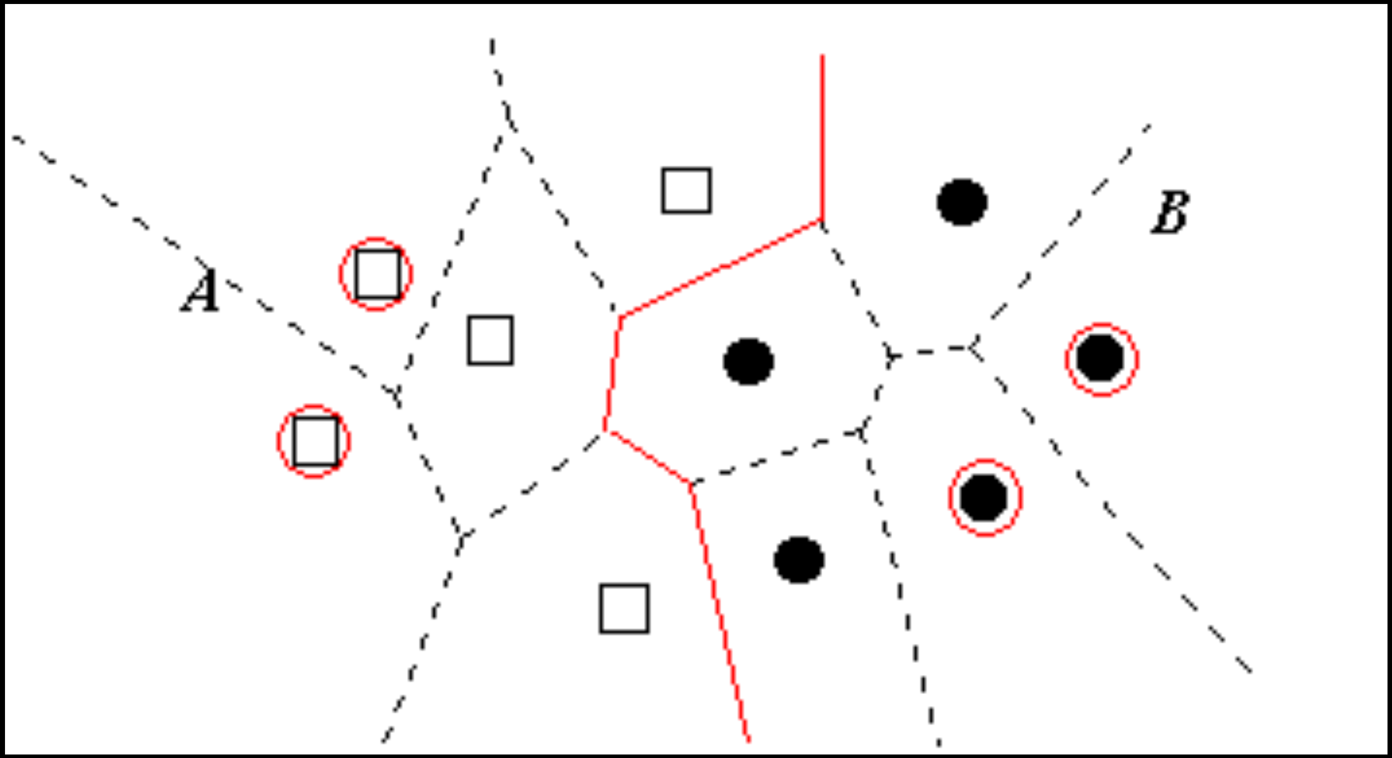
# Two evaluation measures

Accuracy: (number of 1's)/10 x 100%=60%

Confusion Table

| | estimate = Red | estimate = Blue |
|---|---|---|
| actual = Red | 3 | 2 |
| actual = Blue | 2 | 3 |

# Decision Boundary in 1-NN classifier



# *k*-NN

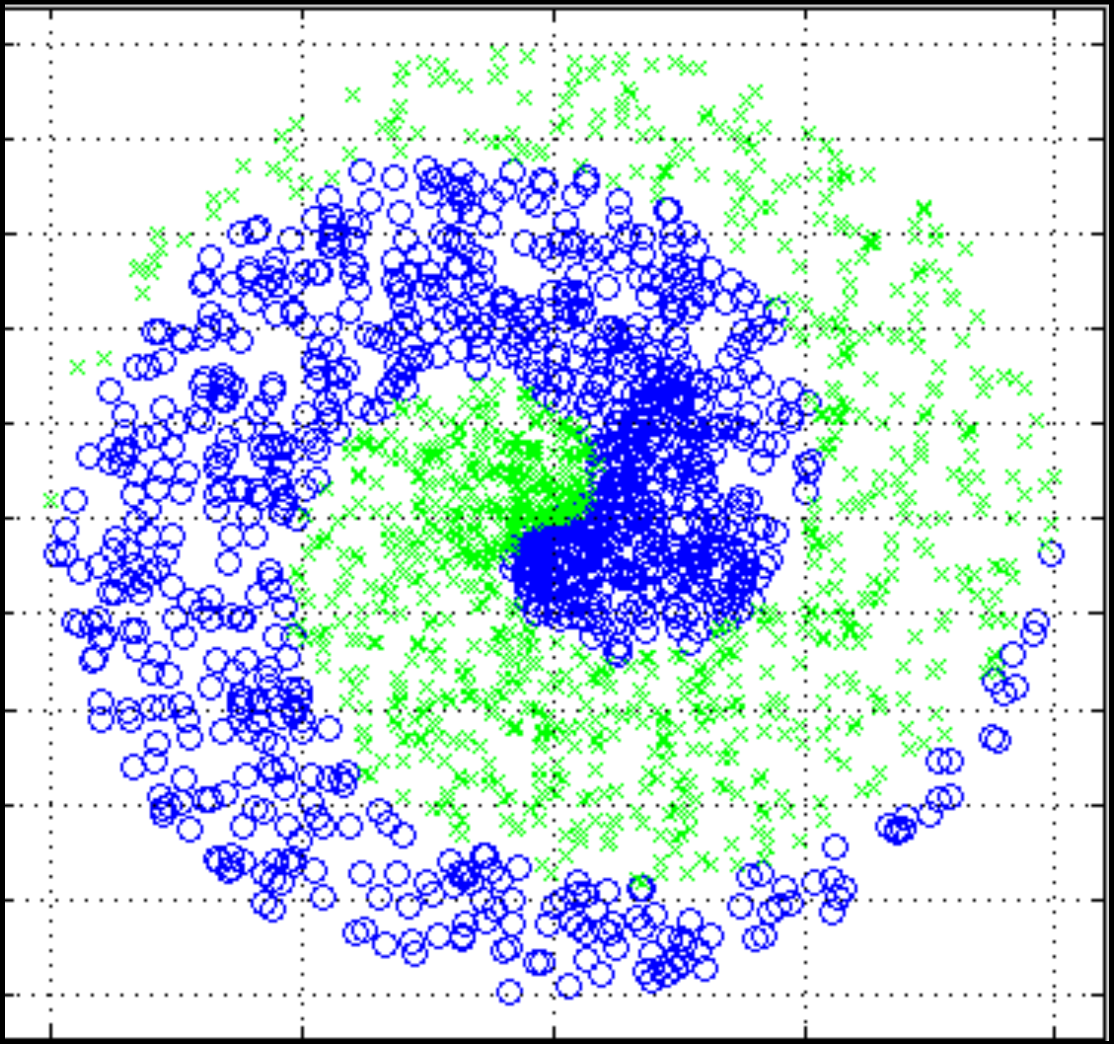
- In the *k*-NN classifier, the parameter *k* represents the number of labeled neighbours considered

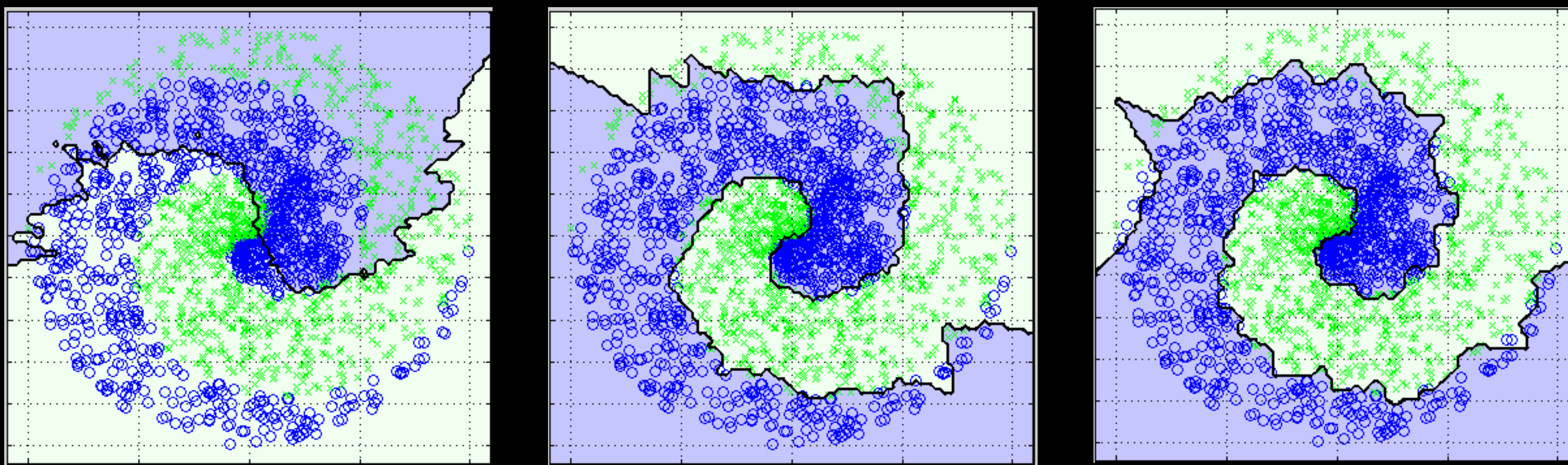
  *k* = 3: test examples are assigned the labels of the (majority of the) 3 nearest neighbours

  *k* = N: test examples are assigned the labels of the (majority of the) N nearest neighbours

  For even N in case of an equal number of nearest neighboring labels of two classes: flip a coin

# Toy dataset: spirals



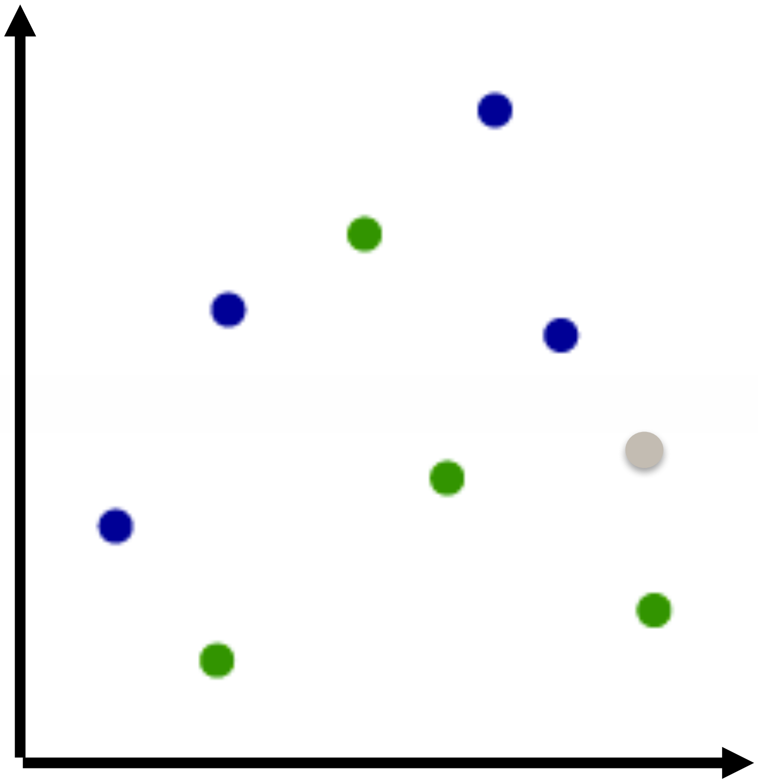


k = 100 k = 10 k = 1

Decreasing number of neighbours

# Decision Tree

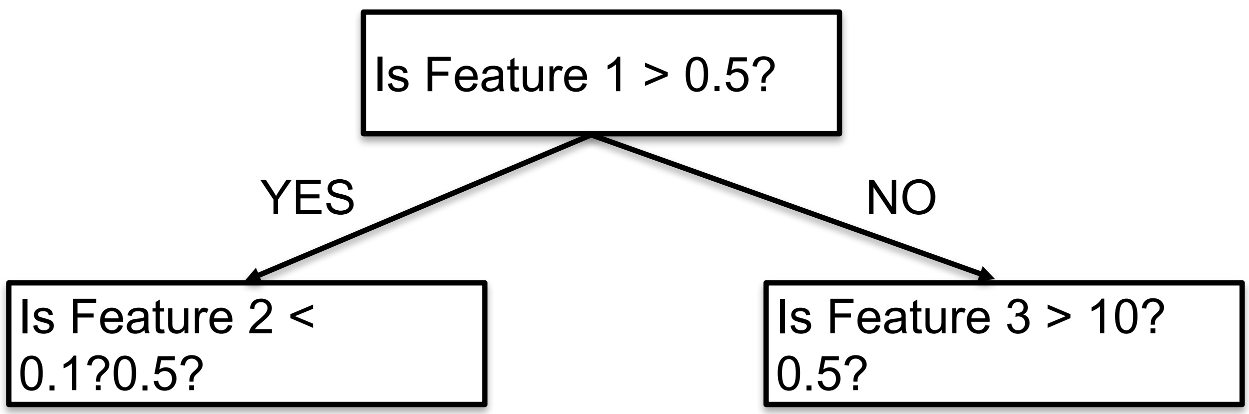
# Classification Problem (blue or green?)



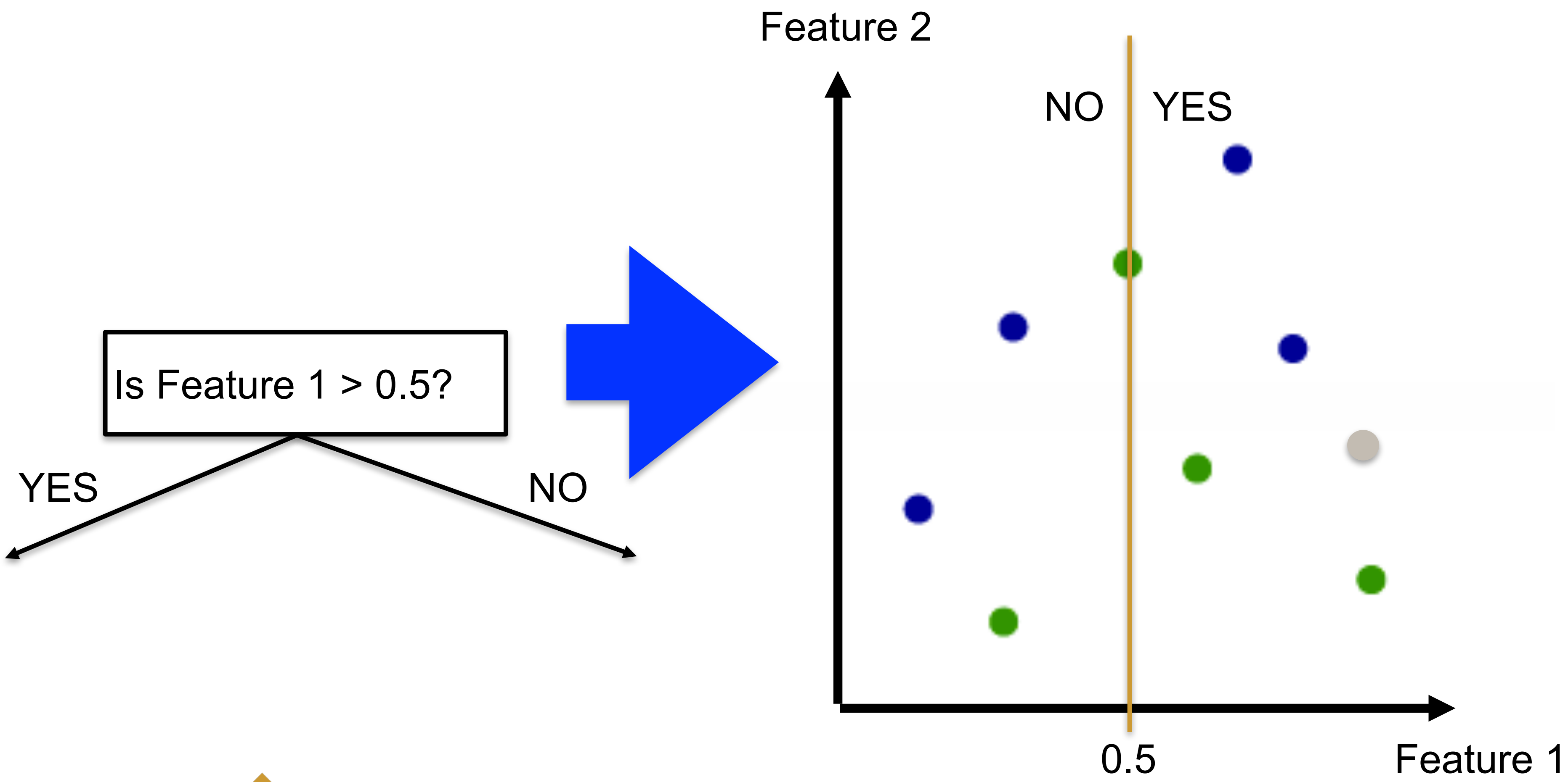

- Train instances
  - blue and green
- Test instance
  - gray
- Classifier induced from the data defines decision boundaries

# Decision Trees

- Decision Trees take one feature at a time and test a binary condition
  For instance: is the feature larger than 0.5?
  If the answer is YES, grow a node to the left
  If the answer is NOW grow a node to the right



# This results in the following decision Boundary

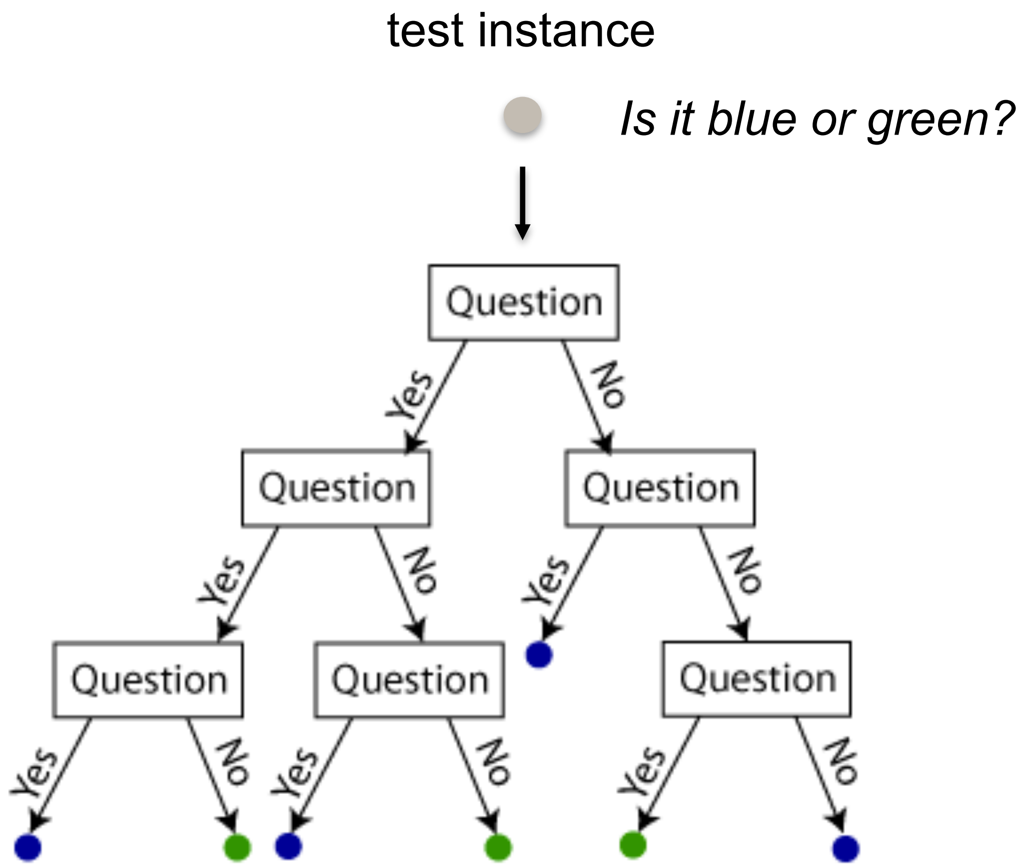



Tilburg University

# Decision Tree grows with each level of questions

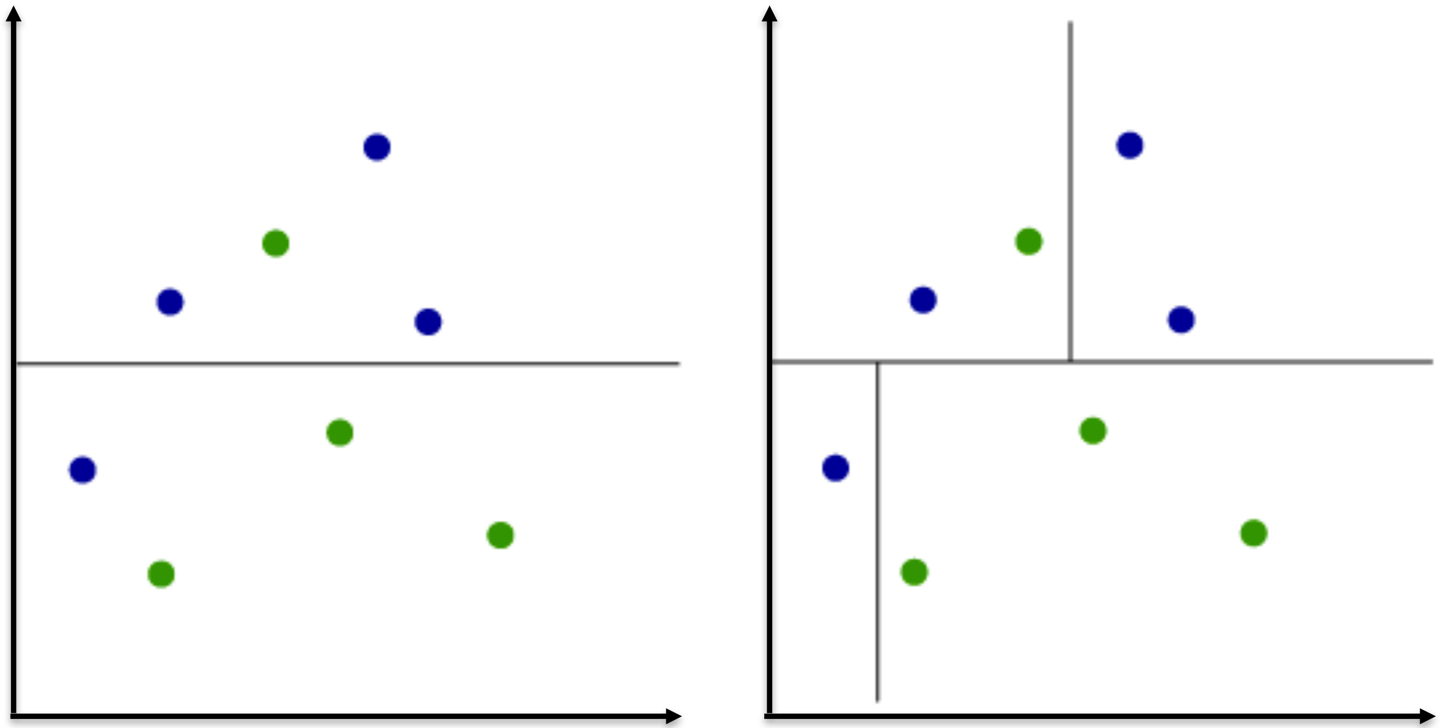
- Each node (box) of the decision tree tests a condition on a feature
- The order of features is important
- It is like playing “20 questions”
  - “Guess the person”: it is better to start with the question “Is he male?”, rather than with “Is it Chris?”
  - The reason is that the answer to the first question maximises the information (“entropy”) gained from the answer.*
- In decision trees the order of features to be tested is determined by means of information theory (ID3 algorithm)

# Decision Tree




Reproduced from: https://shapeofdata.wordpress.com/2013/07/02/decision-trees/

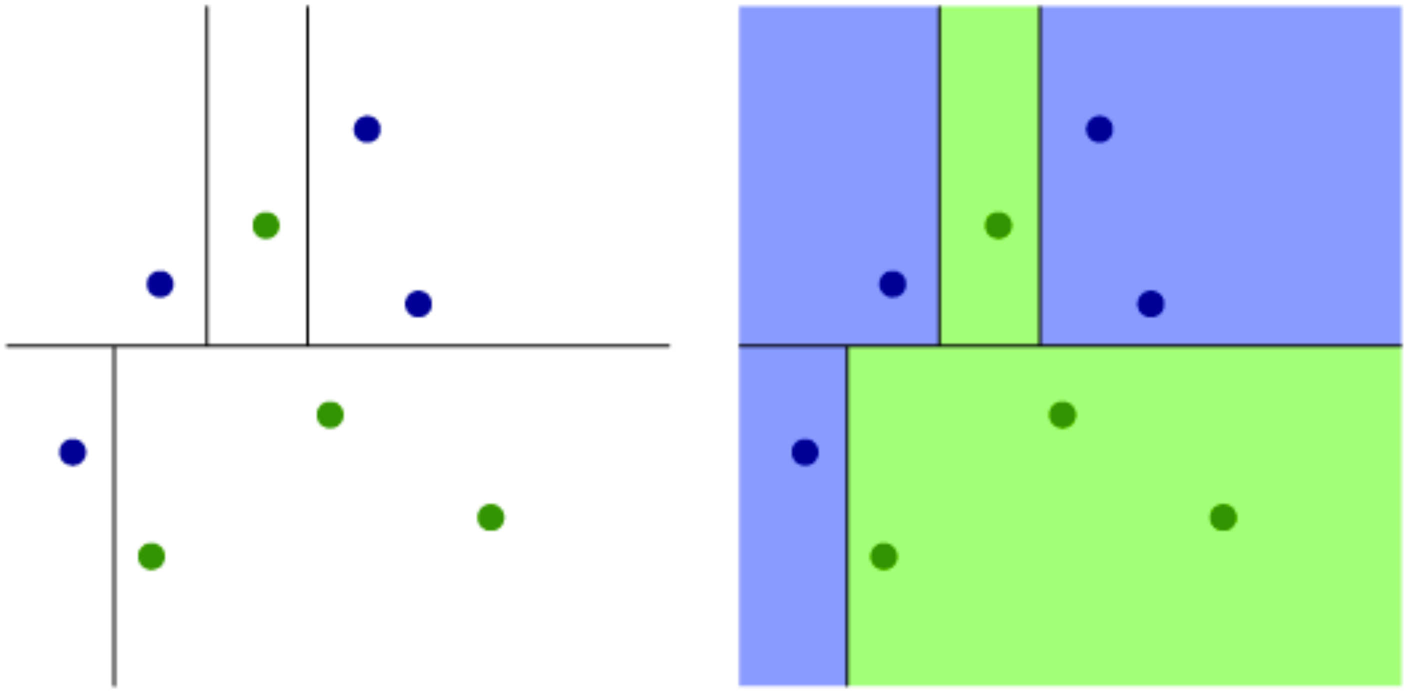
# Each test (box) adds a decision boundary



Reproduced from: https://shapeofdata.wordpress.com/2013/07/02/decision-trees/

Tilburg University

# Adding another decision boundary

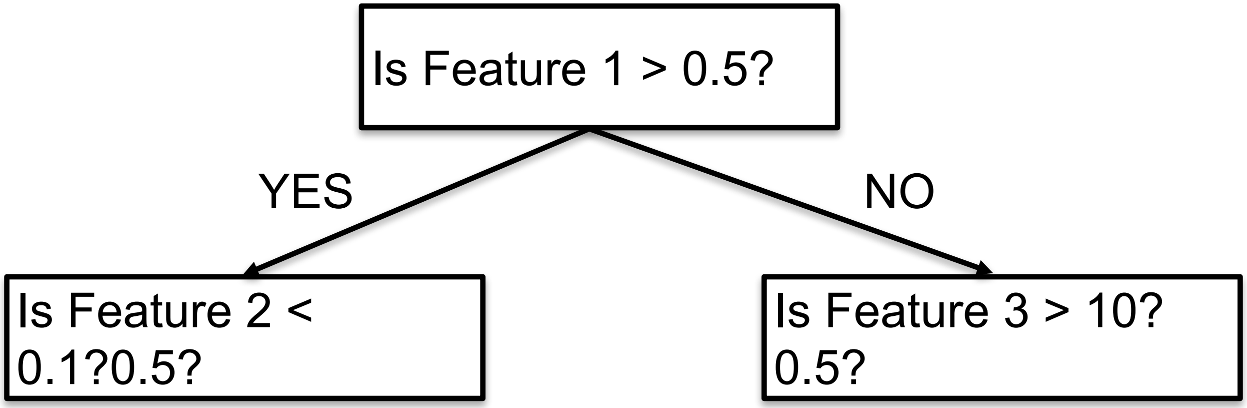


Reproduced from: https://shapeofdata.wordpress.com/2013/07/02/decision-trees/

TILBURG UNIVERSITY

# Complexity of the induced model

- The complexity of the model induced by a decision tree is determined by the depth of the tree
- Increasing the depth of the tree increases the number of decision boundaries
- All decision boundaries are perpendicular to the feature axes, because at each node a decision is made about a single feature



# Summary

We have introduced two classifiers

- nearest neighbour classifier
- decision tree classifier

We know what decision boundaries are

We know how they relate to the complexity of induced models in the classifiers

We have introduced two evaluation measures

# Required Reading

WEKA book:

**Chapter 2**

**Section 3.5** Instance-based representation

50 years of Data Science

David Donoho

Sept. 18, 2015
Version 1.00

A “data science = statistics” overview