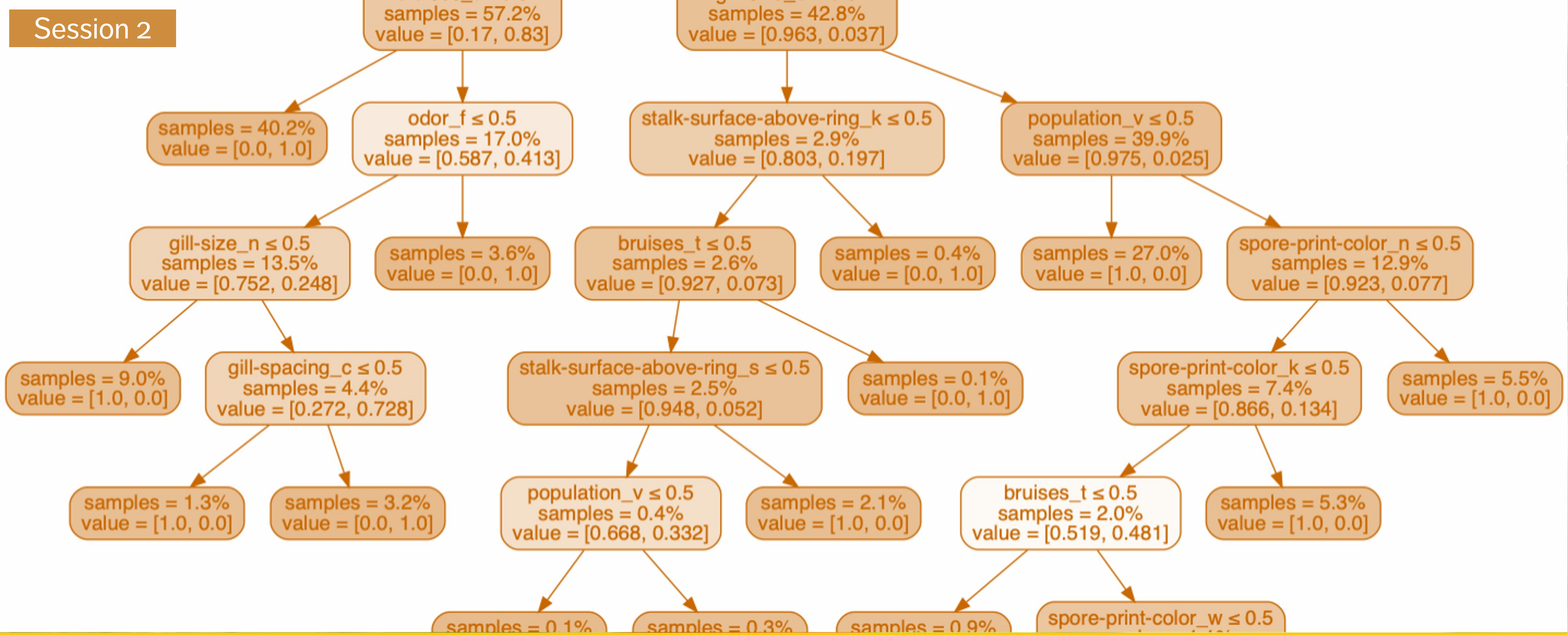


5. Decision Trees and Other Algorithms

Classification Algorithms

Logistic Regression

- classical model
- affected by variance inflation and variable selection process

Neural Networks

- hard to interpret
- requires all variables to be of the same type
- easier to train since backpropagation (chain rule)

Bayesian Methods

Decision Trees

- may overfit the data if not pruned correctly (manually?)

Classification Algorithms

Naïve Bayes Classifiers

- quite successful for text mining applications (spam filter)
- assumptions not often met in practice

Support Vector Machines

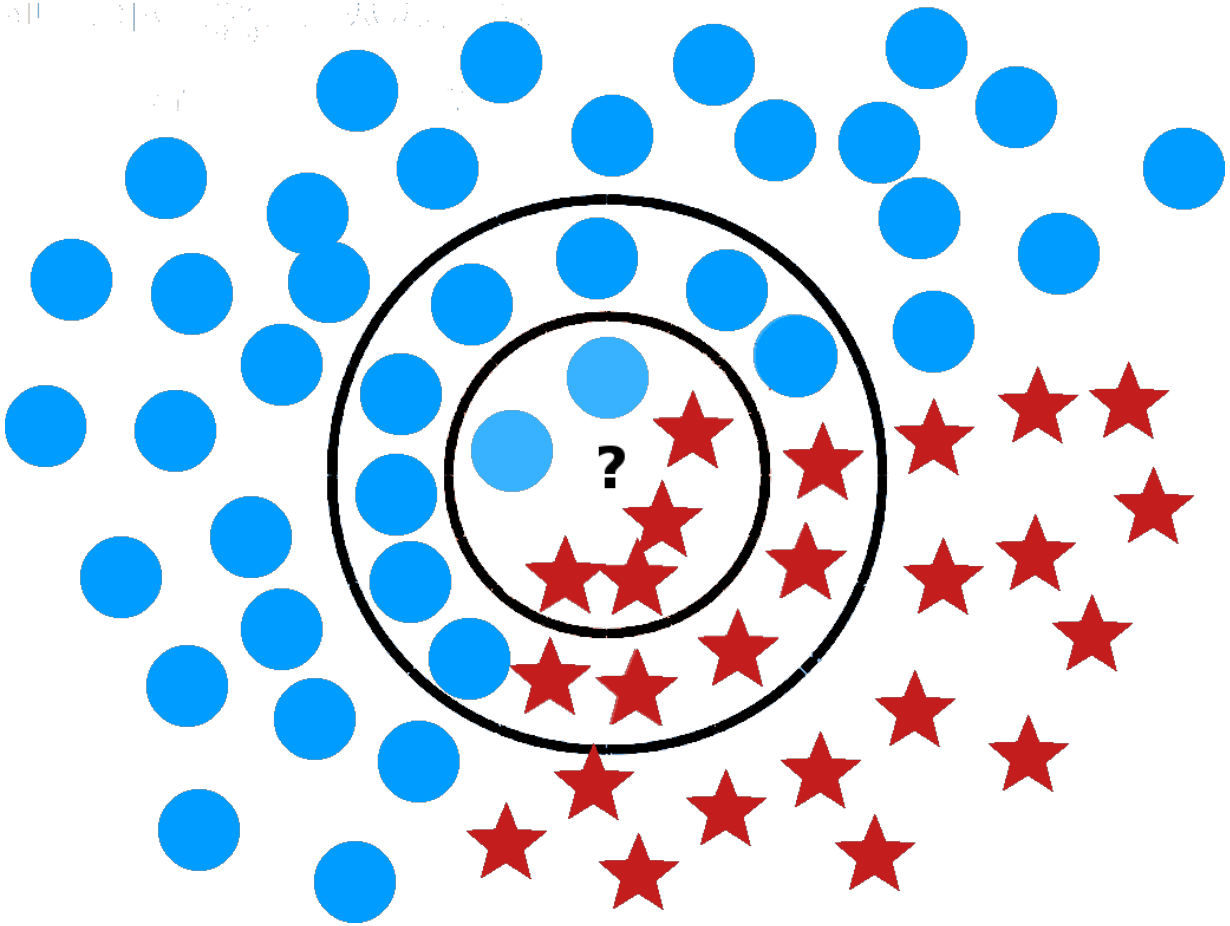
- may be difficult to interpret (non-linear boundaries)
- can help mitigate big data difficulties

Boosting Methods

Nearest Neighbours Classifiers

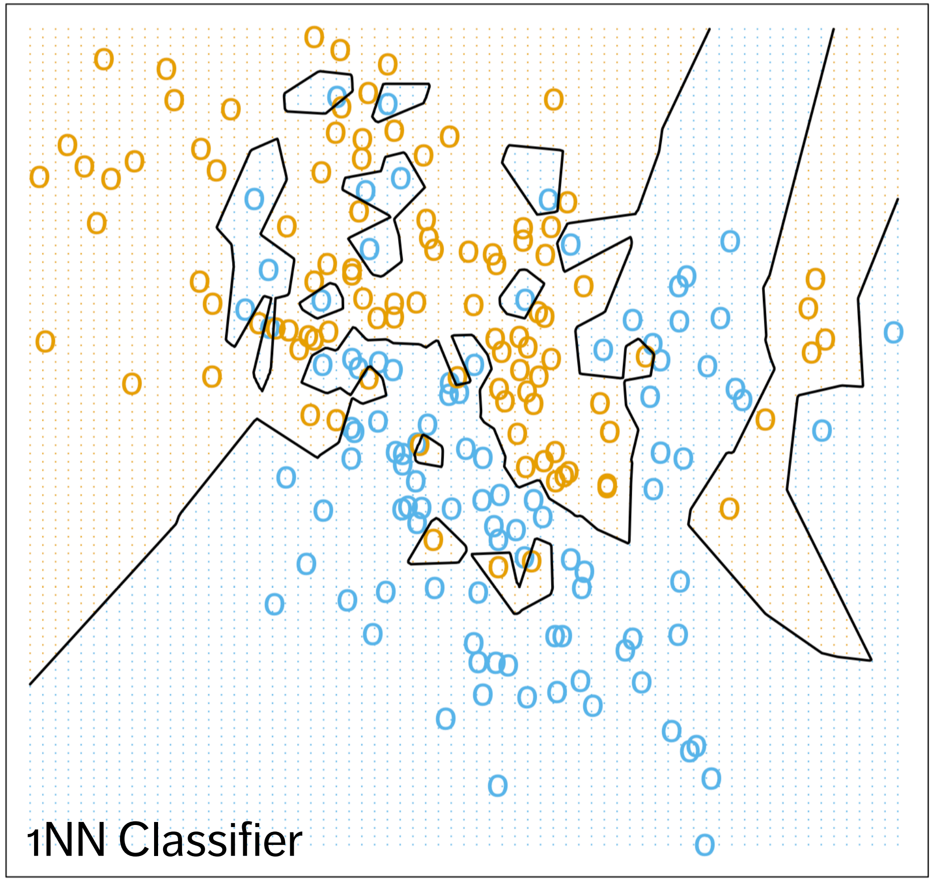
- require very little assumptions about the data
- not very stable (adding points may substantially modify the boundary)

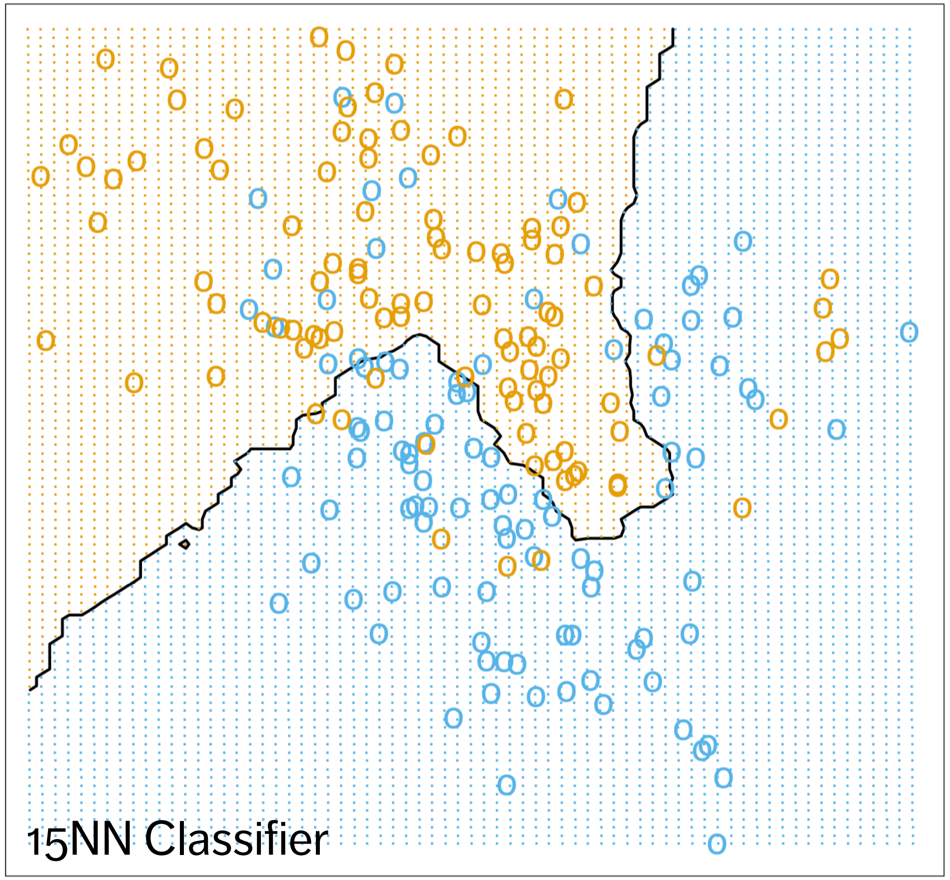
k –Nearest Neighbours Classifier



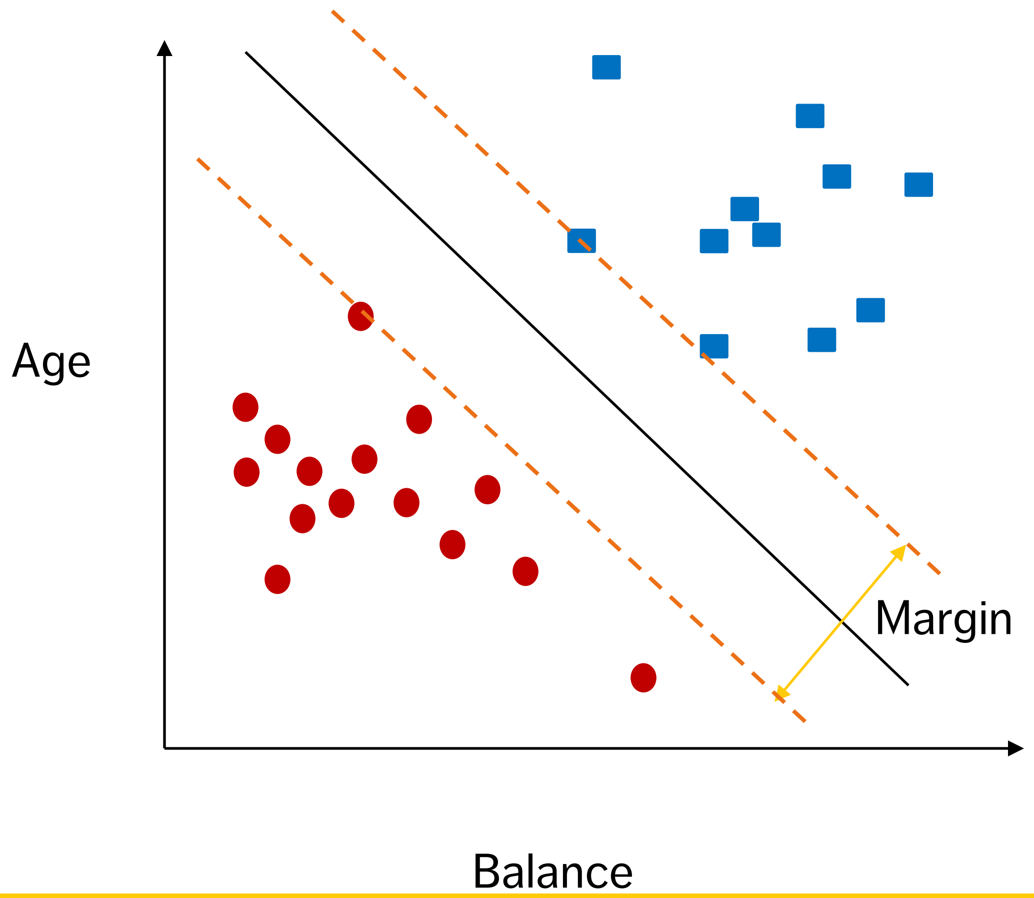
[Tibshirani, Hastie & Friedman, *The Elements of Statistical Learning*]

k –Nearest Neighbours Classifier



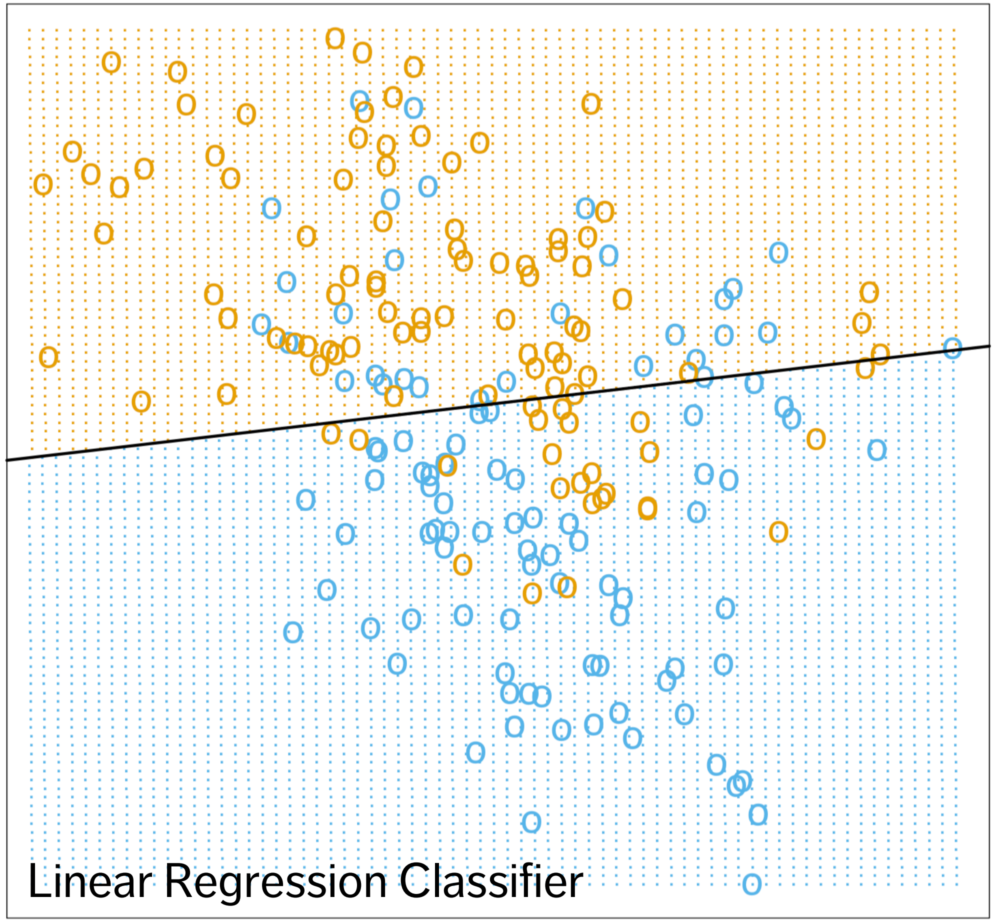


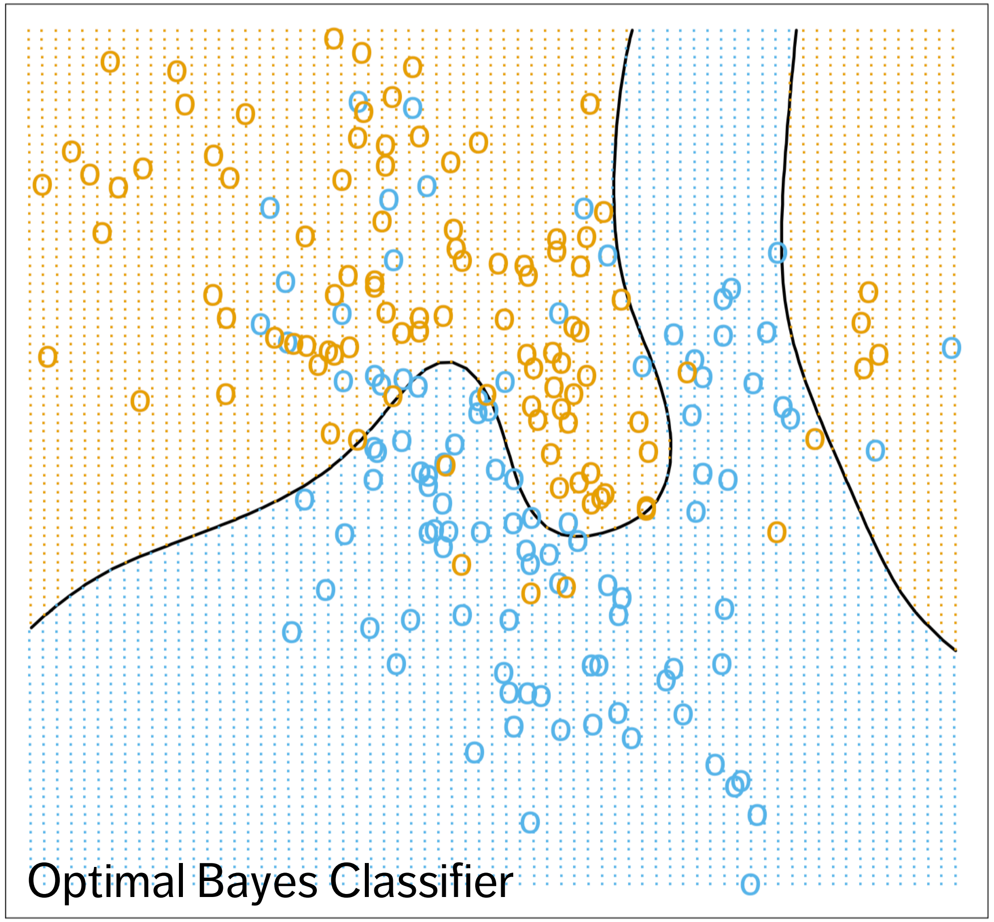
Support Vector Machines



[Tibshirani, Hastie & Friedman, *The Elements of Statistical Learning*]

Other Classifiers





[adapted from A. Ng, K. Soo, *Numsense!*, Surviving a Disaster]

Decision Trees

Decision trees are perhaps the most **intuitive** of these methods: classification is achieved by following a path up the tree, from its **root**, through its **branches**, and ending at its **leaves**.

Decision Trees

To make a **prediction** for a new instance, follow the path down the tree, and read the prediction directly once a leaf is reached.

Creating the tree and traversing it might be **time-consuming** if there are too many variables.

Prediction accuracy can be a concern in trees whose growth is **unchecked**. In practice, the criterion of **purity** at the leaf-level is linked to bad prediction rates for new instances.

- other criteria are often used to prune trees, which may lead to **impure** leaves (i.e. with non-trivial entropy).

Decision Tree Algorithm (ID3)

Task: grow a decision tree using a training set (a subset of the data for which the correct classification of the target is known).

Overview:

1. Split the training data (**parent**) set into (**children**) subsets, using the different levels of a particular attribute
2. Compute the **information gain** for each subset
3. Select the **most advantageous** split
4. Repeat for each node until some **leaf** criterion is met (each item in the leaf has the same classification is one possibility)

Information Gain

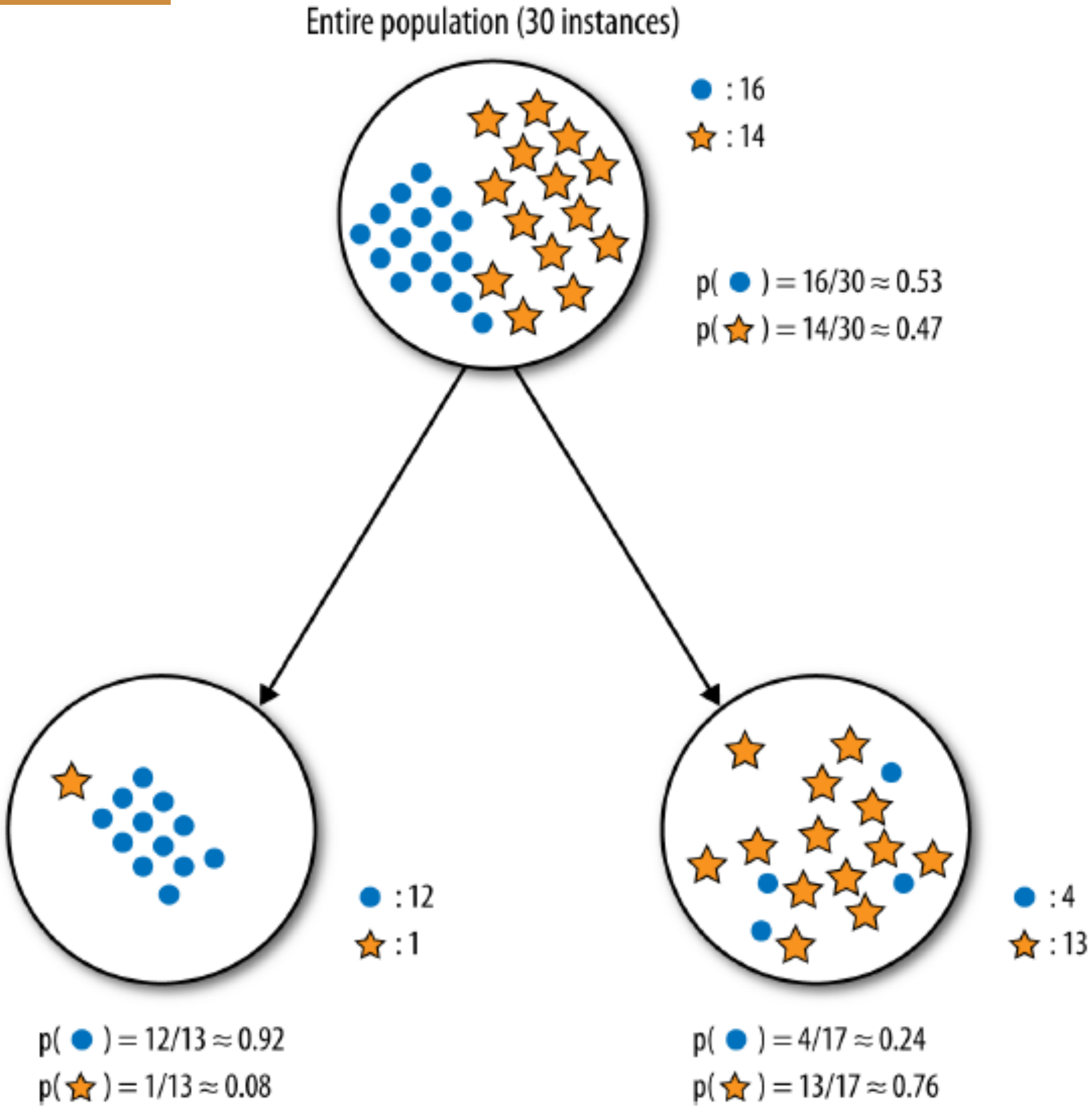
Entropy is a measure of disorder in a set S. Let p_i be the % of observations in S belonging to category i, for $i = 1, \ldots, n$. The entropy of S is given by

$$E(S) = -p_1 \log p_1 - p_2 \log p_2 - \cdots - p_n \log n.$$

If the **parent set** S consisting of m records is split into k **children sets** $C_1, \ldots, C_k$ containing $q_1, \ldots, q_k$ records (resp.), then the **information gain** from the split is given by

$$\mathrm{IG}(S; C) = E(S) - \frac{1}{m}[q_1 E(C_1) + \cdots + q_k E(C_k)].$$

[Provost & Fawcett, *Data Science for Business*]



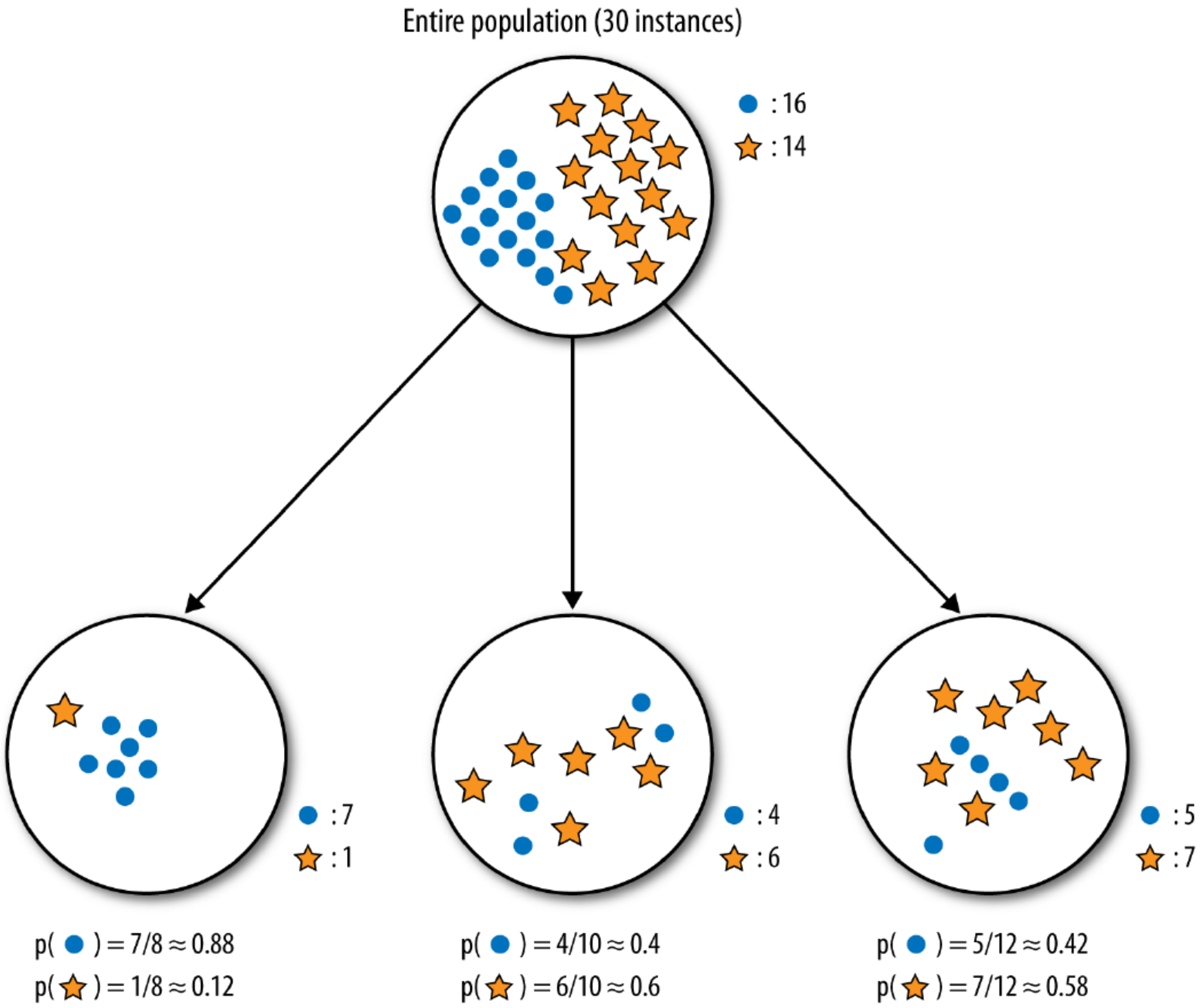
$$E(S) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{16}{30}\log\frac{16}{30} - \frac{14}{30}\log\frac{14}{30} \approx 0.99$$

$$E(L) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{12}{13}\log\frac{12}{13} - \frac{1}{13}\log\frac{1}{13} \approx 0.39$$

$$E(R) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{4}{17}\log\frac{4}{17} - \frac{13}{17}\log\frac{13}{17} \approx 0.79$$

$$\mathrm{IG} = E(S) - \tfrac{1}{30}[q_L E(L) + q_R E(R)]$$
$$\approx 0.99 - \tfrac{1}{30}[13(0.39) \ + 17(0.79)]$$
$$\approx \mathbf{0.37}$$

[Provost & Fawcett, *Data Science for Business*]



$$E(S) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{16}{30}\log\frac{16}{30} - \frac{14}{30}\log\frac{14}{30} \approx 0.99$$

$$E(L) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{7}{8}\log\frac{7}{8} - \frac{1}{8}\log\frac{1}{8} \approx 0.54$$

$$E(C) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{4}{10}\log\frac{4}{10} - \frac{6}{10}\log\frac{6}{10} \approx 0.97$$

$$E(R) = -p_\circ \log p_\circ - p_* \log p_*$$
$$= -\frac{5}{12}\log\frac{5}{12} - \frac{7}{12}\log\frac{7}{12} \approx 0.98$$

$$\mathrm{IG} = E(S) - \tfrac{1}{30}[q_L E(L) + q_C E(C) + q_R E(R)]$$
$$\approx 0.99 - \tfrac{1}{30}[8(0.54) + 10(0.97) + 12(0.98)]$$
$$\approx \mathbf{0.13}$$

Decision Trees Strengths

White box model

- predictions can always be explained by following the appropriate paths

Can be used with **incomplete** datasets

Built-in feature selection

- less relevant features don't tend to be used as splitting features

Makes **no assumption** about

- independence, constant variance, underlying distributions, co-linearity

Decision Trees Limitations

Not as accurate as other algorithms (usually)

Not robust: small changes in the training dataset can lead to a completely different tree, with a completely different predictions

Particularly vulnerable to **overfitting** in the absence of **pruning**

- pruning procedures are typically convoluted

Optimal decision tree learning is **NP-complete**

Biased towards categorical features with **high** number of levels

Decision Trees Notes

Splitting metrics:

- information gain, Gini impurity, variance reduction, etc.

Common variants:

- Iterative Dichotomiser 3, C4.0, C4.5, CHAID, MARS, conditional inference trees, CART

Decision trees can also be combined together using boosting algorithms (**AdaBoost**) or **Random Forests**, providing a type of voting procedure (Ensemble Learning).

Suggested Reading

Decision Trees and Other Algorithms

Data Understanding, Data Analysis, Data Science
Machine Learning 101

Classification and Value Estimation

- Classification Algorithms
- Decision Trees
- Toy Example: Kyphosis Dataset

R Examples

- Classification: Kyphosis Dataset

Spotlight on Classification

*Simple Classification Methods (advanced)
*Rare Occurrences (advanced)
*Other Supervised Approaches (advanced)
*Ensemble Learning (advanced)

Exercises

Decision Trees and Other Algorithms

1. Go over the kyphosis classification example found in DUDADS (see suggested reading). Repeat the process with the titanic dataset (you may wish to visualize the dataset first) in order to build a decision tree that will help you determine if a passenger survived the sinking or not.

Exercises

Decision Trees and Other Algorithms

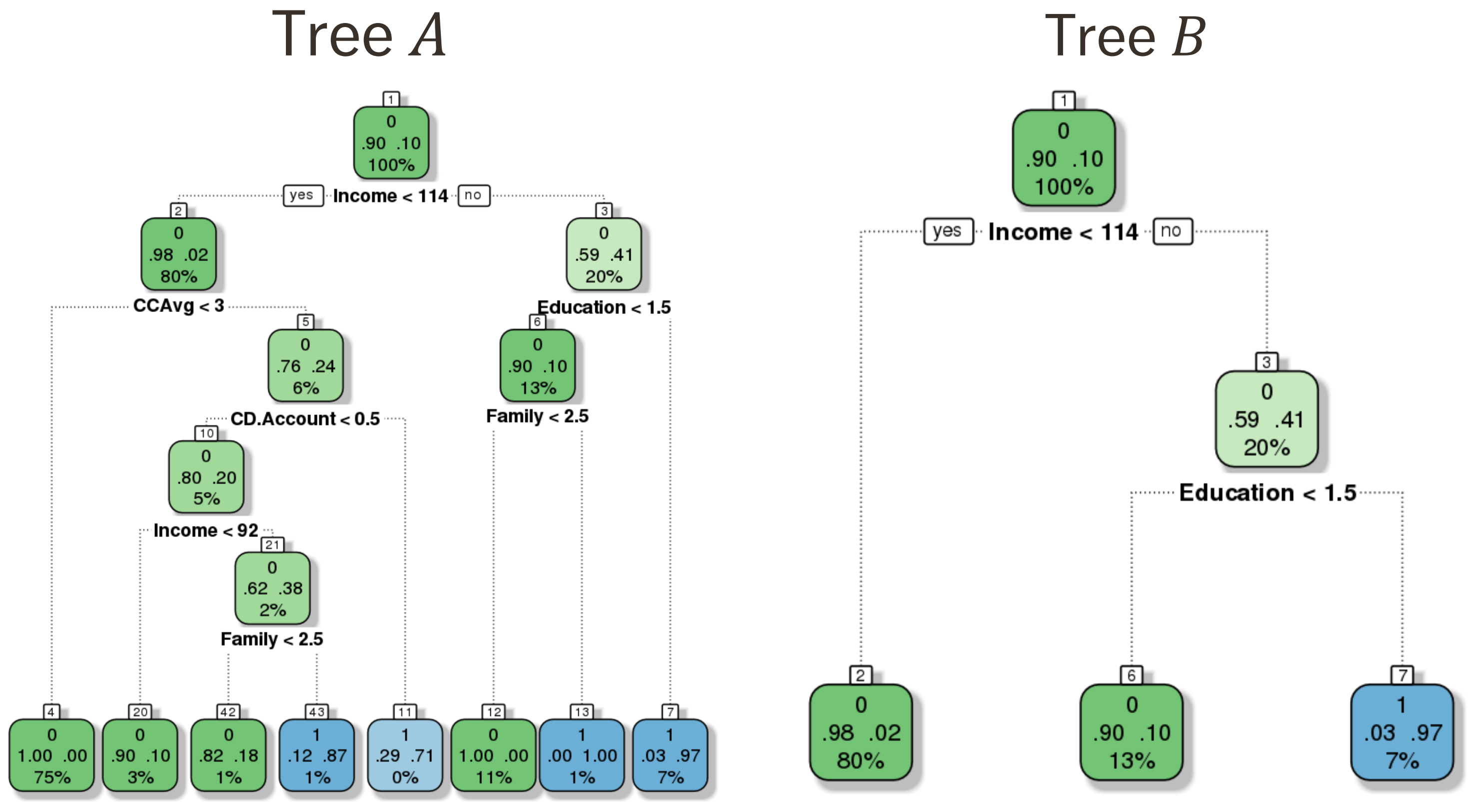
2. UniversalBank is looking at converting its **liability** customers (i.e., customers who only have deposits at the bank) into **asset** customers (i.e., customers who have a loan with the bank). In a previous campaign, *UniversalBank* was able to convert 9.6% of 5000 of its liability customers into asset customers. The marketing department would like to understand what combination of factors make a customer more likely to accept a personal loan, in order to better design the next conversion campaign.

   The dataset contains data on 5000 customers, including the following measurements: age, years of professional experience, yearly income (in $K), family size, value of mortgage with the bank, whether the client has a certificate of deposit with the bank, a credit card, etc.

Exercises

Decision Trees and Other Algorithms

2. (cont.) We build 2 decision trees on a training subset of 3000 records to predict whether a customer is likely to accept a personal loan (1) or not (0).



Exercises

Decision Trees and Other Algorithms

a. How many variables are used in the construction of tree A? Of tree B?

b. Is the following decision rule valid or not for tree A:

```
IF (Income ≥ 114) AND (Education ≥ 1.5)
THEN (Personal Loan = 1)?
```

c. Is the following decision rule valid or not for tree B:

```
IF (Income < 92) AND (CCAvg ≥ 3)
       AND (CD.Account < 0.5)
THEN (Personal Loan = 0)?
```

d. What prediction would tree A make for a customer with:

- yearly income of 94,000$USD (`Income = 94`),
- 2 kids (`Family = 4`),
- no certificate of deposit with the bank (`CD.Account = 0`),
- a credit card interest rate of 3.2% (`CCAvg = 3.2`), and
- a graduate degree in Engineering (`Education = 3`).

e. What about tree B?