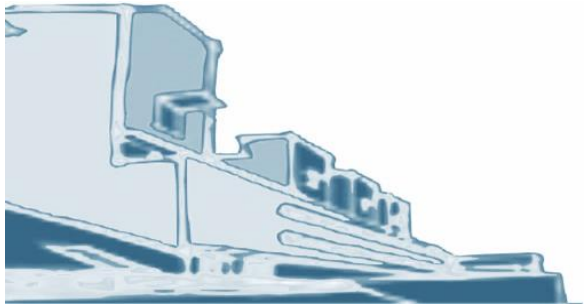
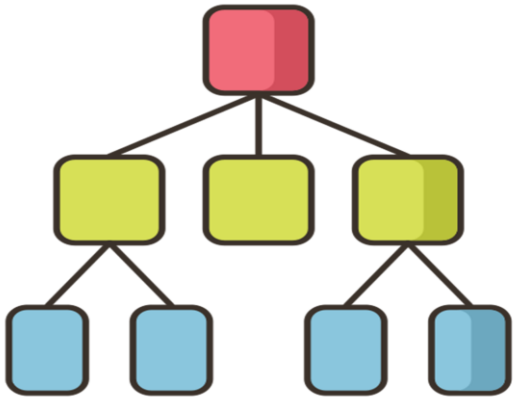
AC – Aprendizagem Computacional / Machine Learning

P3 – Decision Trees

Jorge Henriques
jh@dei.uc.pt

Departamento de Engenharia Informática
Faculdade de Ciências e Tecnologia





1 2 9 0
UNIVERSIDADE D COIMBRA

dei engenharia informática

Contents

- Decision Trees
  - Concepts
  - Build a DT by hand
  - Use python functionalities
    - Test different methods for splitting (Gain information, GINI)
    - Specific parameters
  - Evaluation the performance of the DT classifier
    - Sensitivity, specificity, F1score, …
  - **Interpretability** of a DT

- Decision Trees

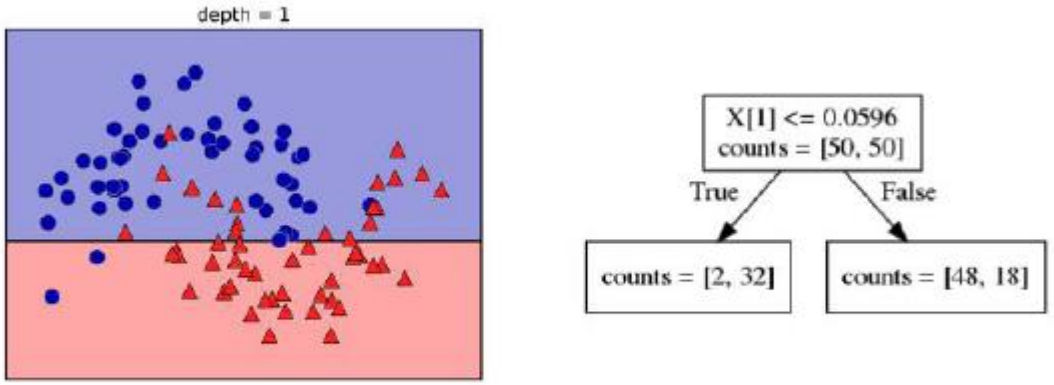


Figure 2-24. Decision boundary of tree with depth 1 (left) and corresponding tree (right)

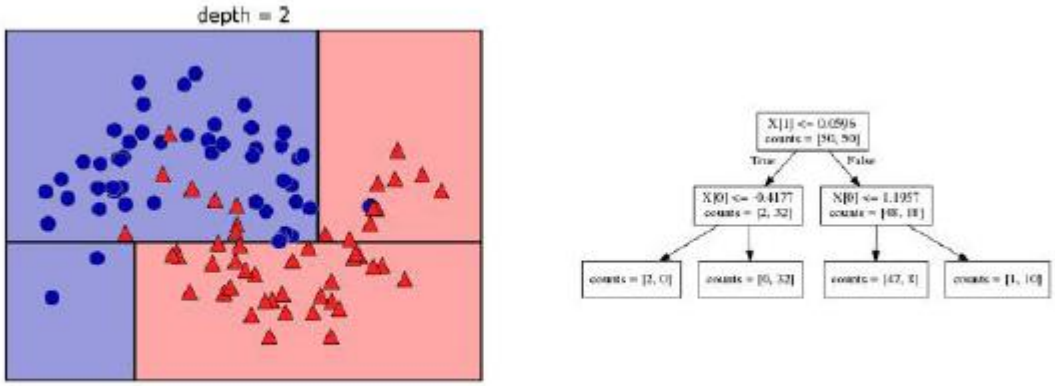


Figure 2-25. Decision boundary of tree with depth 2 (left) and corresponding decision tree (right)

Contents

- **Datasets**

  - 1| Quality of an Apartment
    - Inputs: Furniture {no,yes},
      #rooms {1,2,3,4},
      new kitchen {no,yes}
    - Output Acceptable {no,yes}

  - 2| Cardiac Risk
    - See (1. P1-Introduction)

Contents

- **1| Dataset: Apartment**

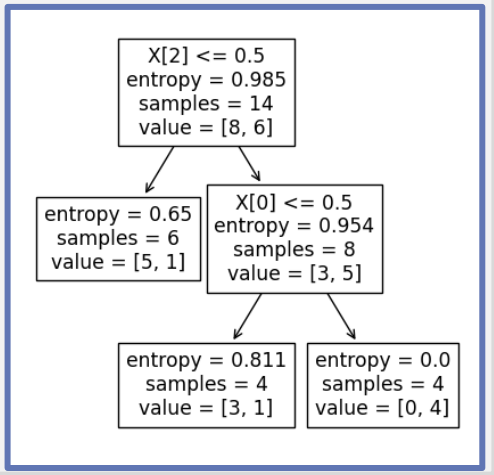
- 1.1| Build a decision tree using ID3
  - Perform the computations by hand
  - Split criterion
    - Gain information or Gini

Furniture	Nrooms	NewKitchen	Acceptable
No	1	Yes	No
Yes	1	No	No
Yes	1	Yes	Yes
No	2	Yes	Yes
Yes	2	No	No
Yes	2	Yes	Yes
No	2	No	No
Yes	3	No	No
No	4	Yes	No
Yes	3	Yes	Yes
Yes	4	No	Yes
No	3	Yes	No
No	4	No	No
Yes	4	Yes	Yes

- **1| Dataset: Apartment**
- **1.2| Interpretability: set of rules**

  - X[0] – Furniture X[1] – Nrooms X[2] - newKitchen

- If X[2] <= 0.5 quality = not acceptable
- If X[2] > 0.5 AND X[0] <=0.5 quality = not acceptable
- If X[2] > 0.5 AND X[0] >0.5 quality = acceptable



- If Furniture = No quality = not acceptable
- If Furniture = Yes AND newKitchen=No quality = not acceptable
- If Furniture = Yes AND newKitchen=Yes quality = acceptable

- **1| Dataset: Apartment**

- 1.3| Performance = ?
  - If Furniture = No — quality = not acceptable
  - If Furniture = Yes AND newKitchen=No — quality = not acceptable
  - If Furniture = Yes AND newKitchen=Yes — quality = acceptable

  - SE= ?
  - SP= ?
  - F1score ?

- **2| Dataset: cardiacRisk**
  - 2.1| Build the decision Trees
    - Use of scikitlearning

```
model = DecisionTreeClassifier(criterion='entropy',
                               splitter='best',
                               max_depth=3,
                               min_samples_split=4,
                               min_samples_leaf=2,
                               max_features=None,
                               random_state=42,
                               max_leaf_nodes=4
```

Parameters

- **Criterion** | how to split the tree
  > Gini, information gain (entropy)

- **Splitter** | strategy used to choose the split at each node of the tree.
  > best (default): considers all possible splits for all features and chooses the one that provides the best possible split
  > random: selects the best split from a random subset of features.

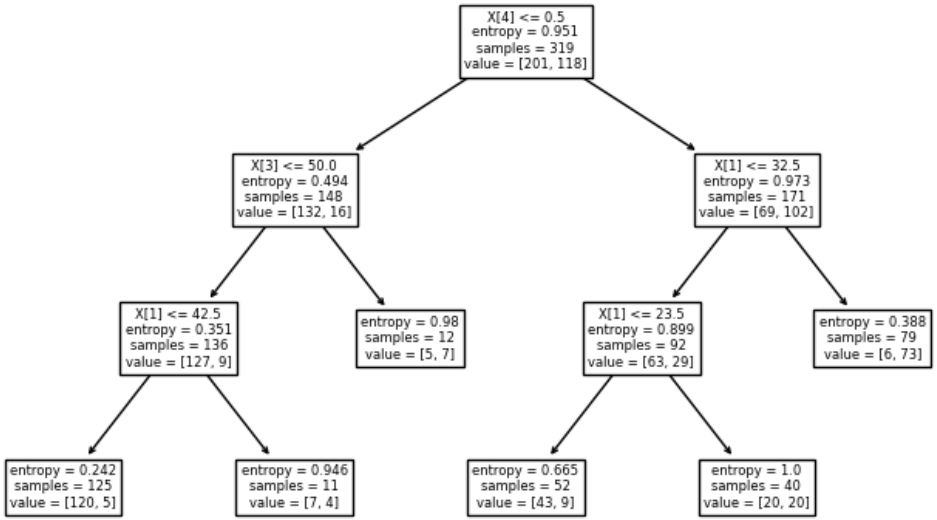
- **max_depth** | limits the maximum depth of the tree
- **min_samples_split** | minimum number of samples required to split an internal node.
- **min_samples_leaf** | minimum number of samples that a leaf node (terminal node) must have.
- **max_features** | number of features to consider when looking for the best split at each node
- **random_state** | randomness involved in various processes of the algorithm, ensuring reproducibility of results (**42 is a seed**)
- **max_leaf_nodes** | controls the maximum number of leaf nodes in a decision tree.

- **2| Dataset: cardiakRisk**
  - 2.2| Train/ test the DT

```
Xtrain, Xtest, Ttrain, Ttest = train_test_split(X,T,test_size = 0.3,
random_state = 42)

model = model.fit(Xtrain, Ttrain)
Ytrain= model.predict( Xtrain)

plt.figure(figsize=(5,5))
plot_tree(model)
plt.show()
```

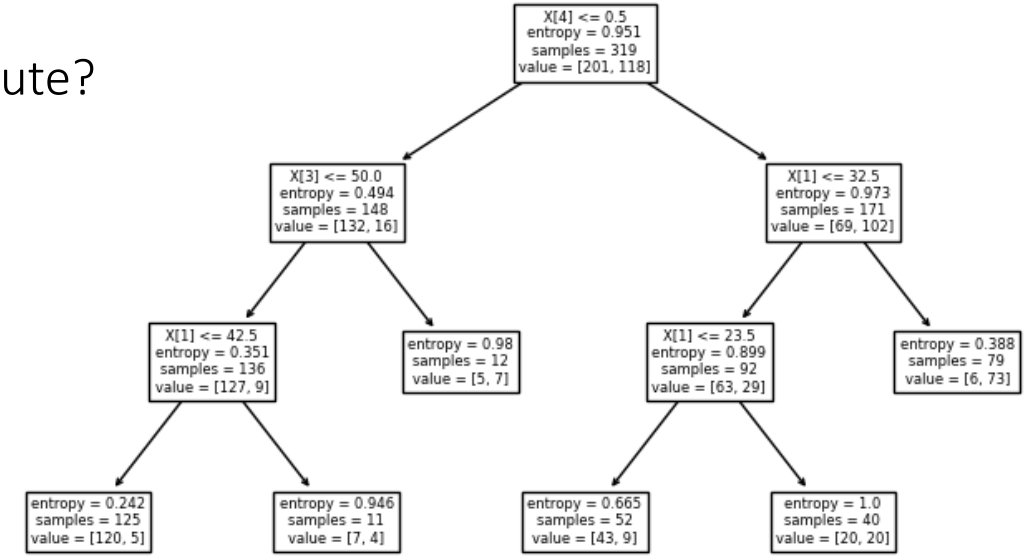


- **2| Dataset: cardiakRisk**
  - 2.3| performance
  - Performance
    - SE, SP, F1score ??

```
cm    = confusion_matrix(Ttrain, Ytrain)
TN, FP, FN, TP = cm.ravel()
SE    = TP/(TP+FN)
SP    = TN/(TN+FP)
```

- **2| Dataset: cardiakRisk**
  - 2.4| Interpretability of the DT

- Can we generate clinical knowldge ?
- Importance
  - Rules ?
  - Most importante atribute?
  - X[4] – ST
  - X[3] – HR
  - X[1] – Age



Contents

4| Conclusions

- Decision Trees
  - Build by hand the DT
  - Build using Scikit functionalities
  - Splitting technique (information gain / Gini)
  - Rules - interpretability
  - Train/test to cardiacRisk dataset
  - Performance

4| Conclusions

- Improvements
  - Any other idea ?

- CardiacRisk, Questions
  - Can we derive a "***clinical guideline***" ?
  - Are the rules aligned with clinical practice / knowledge?
  - Tradeoff rules / performance
  - Number of rules – interpretability ?