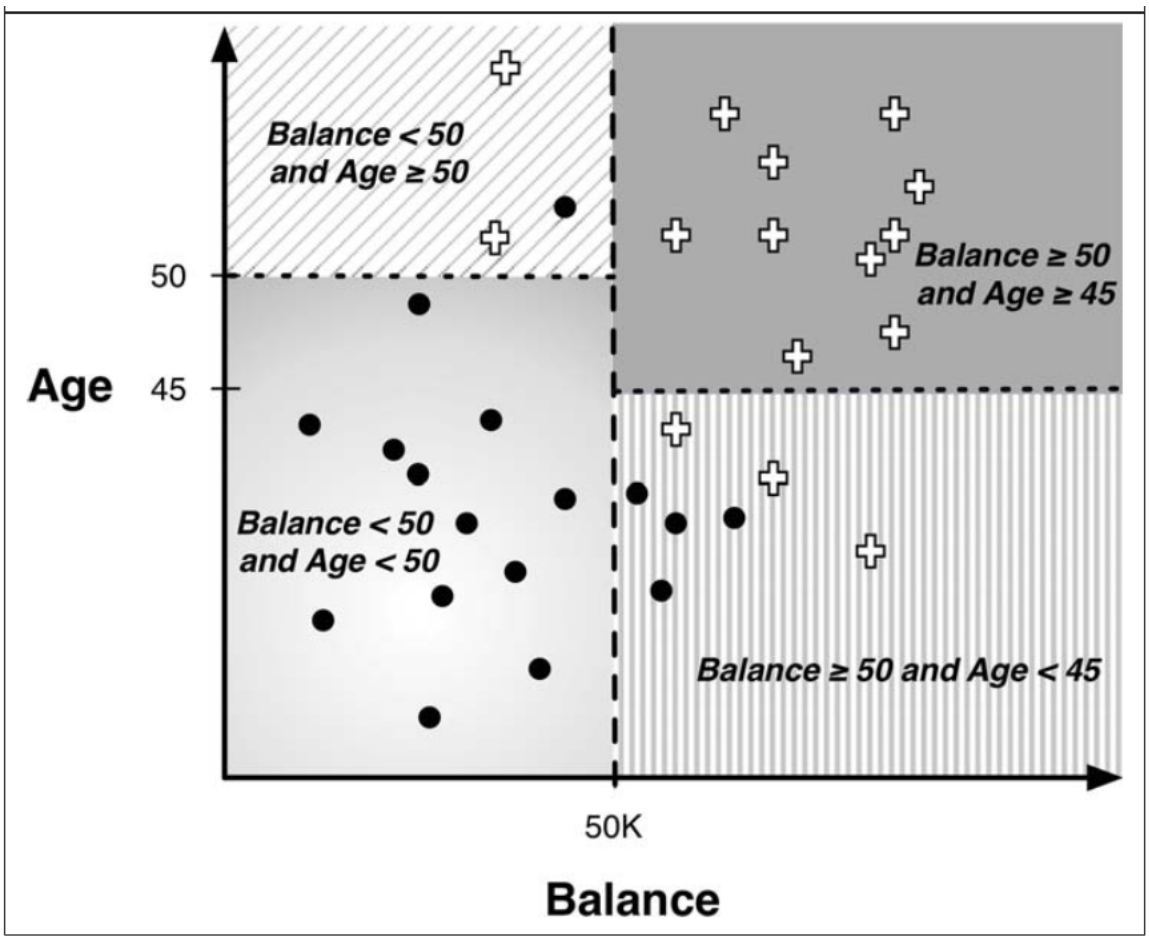
Live coding: decition trees

AH

1 Decision-trees: keywords

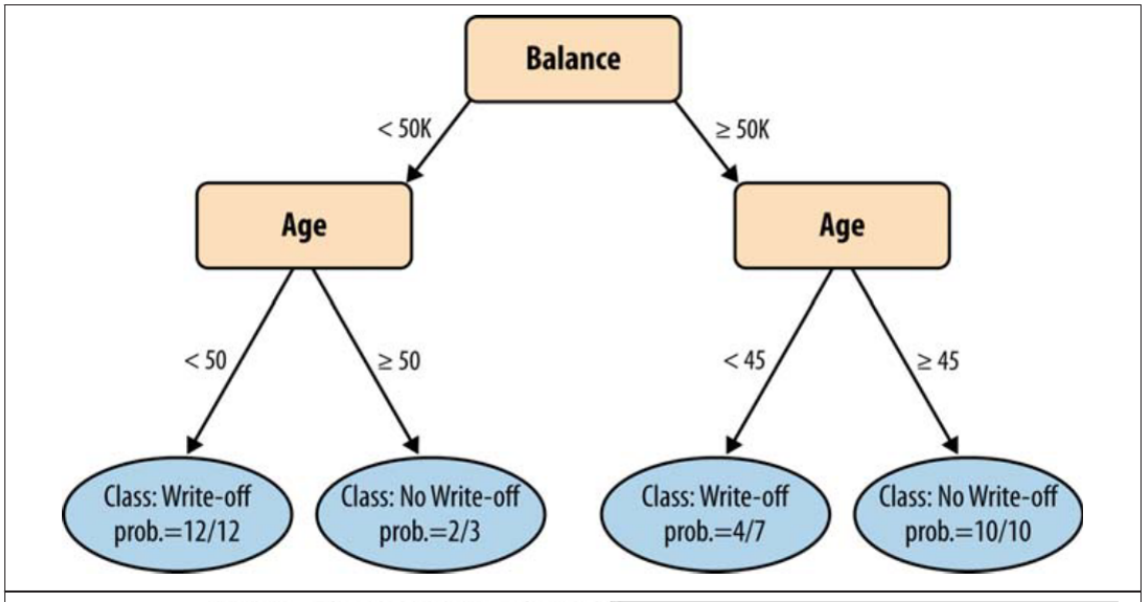
Discretization, iterated binary segmentation, *misclassification* ...



Decision regions appear

2 Decision-trees: more keywords

Purity, Entropy, Information Gain, root-to-leaf ...



---

3 Dataset: banknote authentication

Visit the UCI ML Repository.

banknote authentication Data Set

Download: Data Folder, Data Set Description

Abstract: Data were extracted from images that were taken for the evaluation of an authentication procedure for banknotes.

Data Set Characteristics:	Multivariate	**Number of Instances:**	1372	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	5	**Date Donated**	2013-04-16
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	220743

Source:

Owner of database: Volker Lohweg (University of Applied Sciences, Ostwestfalen-Lippe, volker.lohweg '@' hs-owl.de)
Donor of database: Helene DÃ¶rksen (University of Applied Sciences, Ostwestfalen-Lippe, helene.doerksen '@' hs-owl.de)
Date received: August, 2012

Wavelet transformation yields the following features:

- variance, skewness, curtosis and
- entropy of image.

One (integer) classification value: class

---

4 Implementing a decision tree

The Banknotes dataset, baseline code and model solutions are all available from the class channels: Moodle, GitHub, etc.

---

5 Objective A: write our own Gini function

1. inspect a Decision-tree baseline code
2. Lay out a function that segments the data according to the best Gini values available.:

Remember: Gini=0 is the best scenario

```
def get_split(dataset):
    b_index, b_value, b_score, b_groups = 999, 999, 999, None

    # TODO: Find the best possible place to split the dataset
    #
    # TODO: assign datapoints to 'left' and 'right' segments
    # using the test_split(index, value, dataset)
    # function.
    #
    # TODO: define a gini_index(groups, classes)
    # func. to construct a branch of the tree

    return {'index':b_index, 'value':b_value, 'groups':b_groups}
```

---

6 Objective a, cont'd

Compute Gini index for a split dataset

```
def gini_index(groups, classes):

    total_gini = 0.0

    # TODO : For each group, calculate its Gini index.

    return total_gini
```

A model solution for this exercise is available but please attempt your solution first.

If you are uncertain on how to write this type of function you can go directly to the model solution.

---

7 Objective b: write your own DT generator

Can you write Python functions that iteratively segment the data until you have a decision tree?