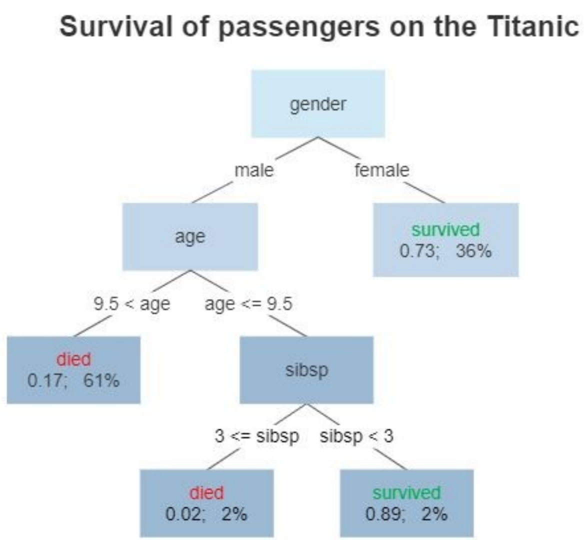
Decision trees

Vlad Gladkikh

IBS CMCM

Observations about an item are represented in the branches



https://en.wikipedia.org/wiki/Decision_tree_learning

https://www.guru99.com/r-decision-trees.html

Conclusions about the item's target value are represented in the leaves.

The questions are in the form of axis-aligned splits in the data

Each node in the tree splits the data into two groups using a cutoff value within one of the features.

Top-down

Choose a variable at each step that best splits the set of items

Metrics for the quality of the split generally measure the homogeneity of the target variable within the subsets.

Commonly used metrics: **Gini impurity**, **Information gain**, **Variance reduction**

Classification tree → the predicted outcome is the class to which the data belongs

Regression tree → the predicted outcome is a number

https://philippmuens.com/decision-trees-from-scratch

https://towardsdatascience.com/an-introduction-to-decision-trees-with-python-and-scikit-learn-1a5ba6fc204f

https://mlcourse.ai/articles/topic3-dt-knn/

Decision tree algorithms: ID3, C4.5, CART, Chi-square automatic interaction detection (CHAID), ...

Scikit-learn: only CART and C4.5

Algorithm	Splitting Metric	Pruning Method	Supports Classification and Regression?	Supports Multi-Class Splitting?
CART	Gini index	Cost complexity pruning	Both	No
C4.5	Information gain ratio	Error-based pruning	Both	Yes

Use decision trees for non-linear classification and regression tasks.

Perform pre-pruning by tuning various decision tree hyperparameters, like the maximum depth of the tree, to help reduce overfitting.

Perform various pruning methods such as reduced error pruning to further reduce the complexity of trees and minimize overfitting.

https://www.coursera.org/learn/build-decision-trees-svms-neural-networks

https://datascience.stackexchange.com/questions/10228/when-should-i-use-gini-impurity-as-opposed-to-information-gain-entropy

https://victorzhou.com/blog/gini-impurity/ https://victorzhou.com/blog/information-gain/

Advantages of ideal decision trees + real problems

Simple to understand and interpret: a white box model

If done correctly, otherwise you get an incomprehensible tree

Able to handle both numerical and categorical data.

Scikit-learn: only numerical (which framework works with categorical?)

H2O: http://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/algo-params/categorical_encoding.html

https://stackoverflow.com/questions/50740316/implementing-a-decision-tree-using-h2o

Matlab: how? R https://data-flair.training/blogs/r-decision-trees/

https://medium.com/data-design/visiting-categorical-features-and-encoding-in-decision-trees-53400fa65931

https://datascience.stackexchange.com/questions/52066/why-decision-tree-needs-categorical-variable-to-be-encoded

Requires little data preparation. E.g. no need to normalize data

Performs well with large datasets

Mirrors human decision making more closely than other approaches

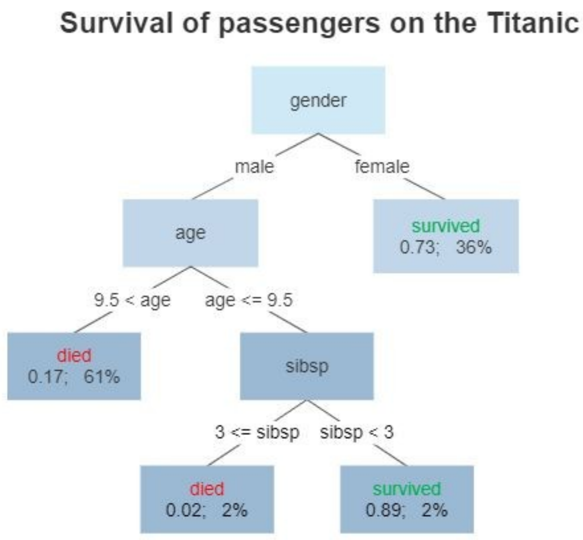
In built feature selection

But I would not trust it: more on that later

Decision trees can approximate any Boolean function e.g. XOR

It will fail to do so for most commonly used split metrics

A more or less ideal tree:



Interpretation:

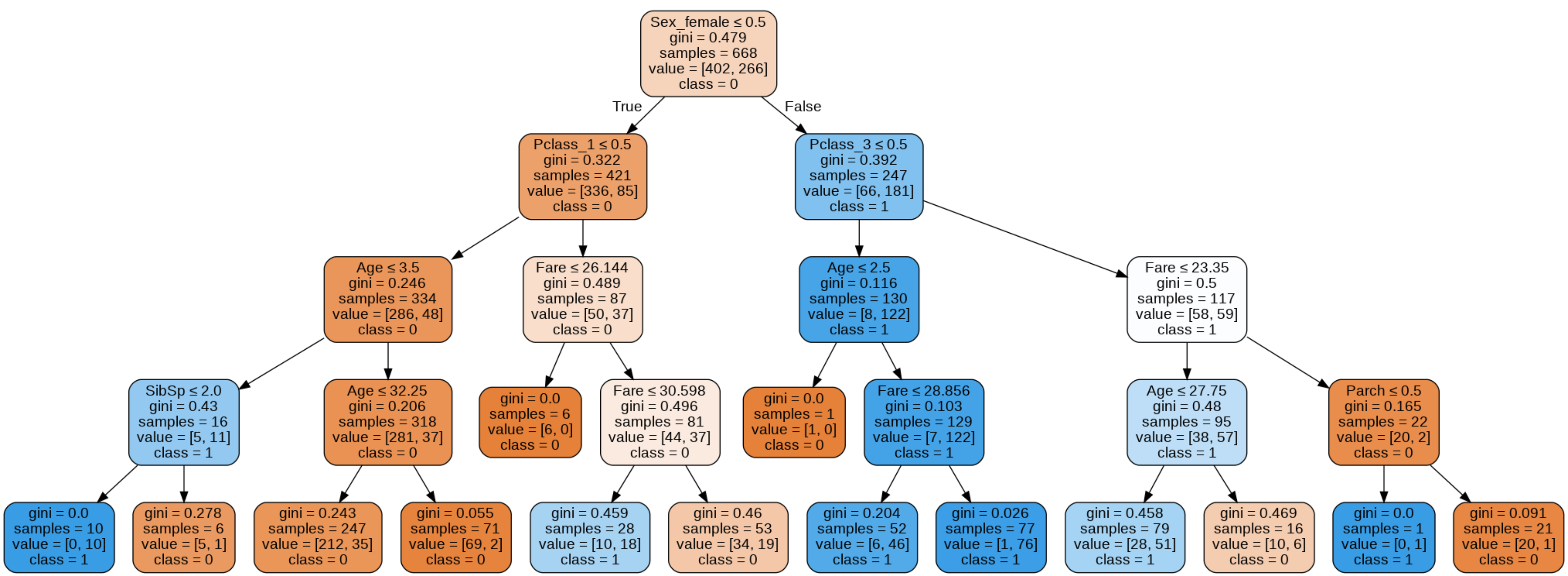
Your chances of survival were good if you were

1) a female or
2) a male younger than 9.5 y.o. with strictly less than 3 siblings.

https://en.wikipedia.org/wiki/Decision_tree_learning

But try it in scikit-learn, and you will probably get something like this:

https://www.coursera.org/learn/build-decision-trees-svms-neural-networks



Limitations

Decision trees can be very non-robust.

A small change in the training data can result in a large change in the tree and consequently the final predictions.

Learning an optimal decision tree is a global optimization problem.

Practical decision-tree learning algorithms are based on heuristics such as the greedy algorithm where locally optimal decisions are made at each node.

Decision-tree learners can create over-complex trees that do not generalize well from the training data.

For data including categorical variables with different numbers of levels, information gain in decision trees is biased in favor of attributes with more levels.

All paths from the root node to the leaf node proceed by way of conjunction, or AND.

In a decision graph, it is possible to use disjunctions (ORs)

http://users.monash.edu/~dld/Publications/2003/Tan+Dowe2003_MMLDecisionGraphs.pdf https://stats.stackexchange.com/questions/115509/how-to-split-a-decision-tree-when-information-gains-of-all-attributes-are-zero

How to split a decision tree when information gains of all attributes are zero?

Asked 6 years, 5 months ago Active 6 years, 5 months ago Viewed 2k times

3

2

The textbook tells us that we should choose an attribute with the maximum information gain to split a decision tree. My question is what if all information gains are zero? Should we stop splitting or we split the tree with all attributes?

An example of this question is $Y = a\ XOR\ b$. To determine the value of Y, the information gains of a and b are zero. How do we build a decision tree for this question?

machine-learning cart entropy

Share Cite Edit Follow Flag

asked Sep 15 '14 at 14:30
azure
31 2

It's March 2021 now

If the information gain is zero, there's no further purpose of splitting unless a combination of features yields information. I'm looking into that very question now: stats.stackexchange.com/questions/259176/...
– Brian Bien Jan 31 '17 at 17:25

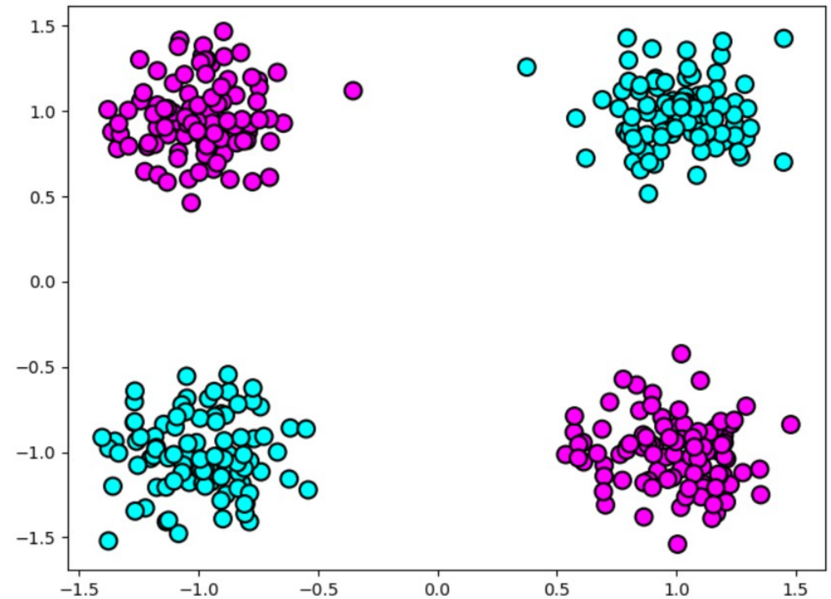
Add a comment

Start a bounty

Know someone who can answer? Share a link to this question via email, Twitter, or Facebook.

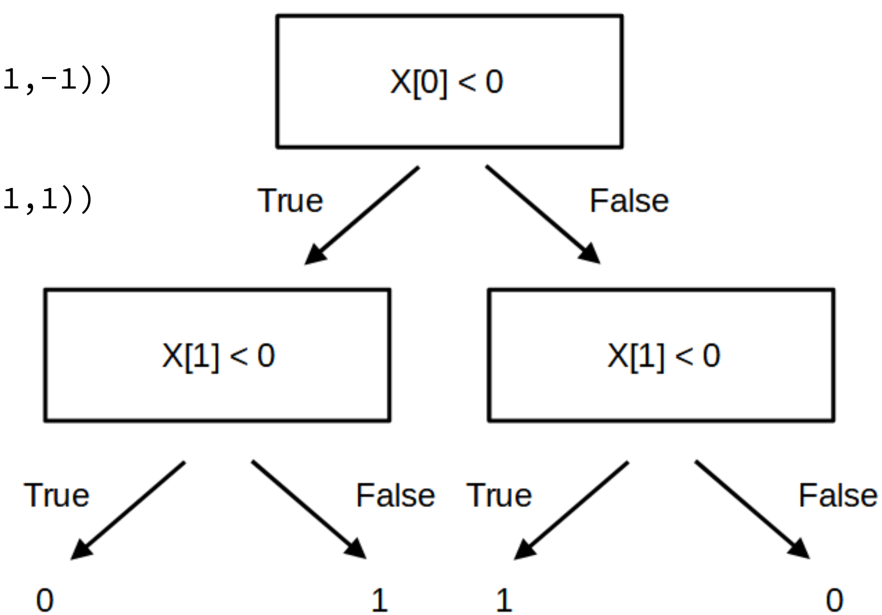
https://stats.stackexchange.com/questions/259176/can-decision-trees-look-multiple-levels-deep-when-selecting-features-to-maximize

XOR




```
import numpy as np
from matplotlib import pyplot as plt
import warnings
warnings.filterwarnings('ignore')

data_1 = np.random.normal(size=(100, 2), scale=0.2, loc=(-1,-1))
labels_1 = np.zeros(100)

data_2 = np.random.normal(size=(100, 2), scale=0.2, loc=(-1,1))
labels_2 = np.ones(100)

data_3 = np.random.normal(size=(100, 2), scale=0.2, loc=(1,-1))
labels_3 = np.ones(100)

data_4 = np.random.normal(size=(100, 2), scale=0.2, loc=(1,1))
labels_4 = np.zeros(100)

data = np.r_[data_1, data_2, data_3, data_4]
labels = np.r_[labels_1, labels_2, labels_3, labels_4]

plt.figure(figsize=(8,6))
plt.scatter(data[:, 0], data[:, 1], c=labels, s=100,
cmap='cool', edgecolors='black', linewidth=1.5);
plt.show()
```

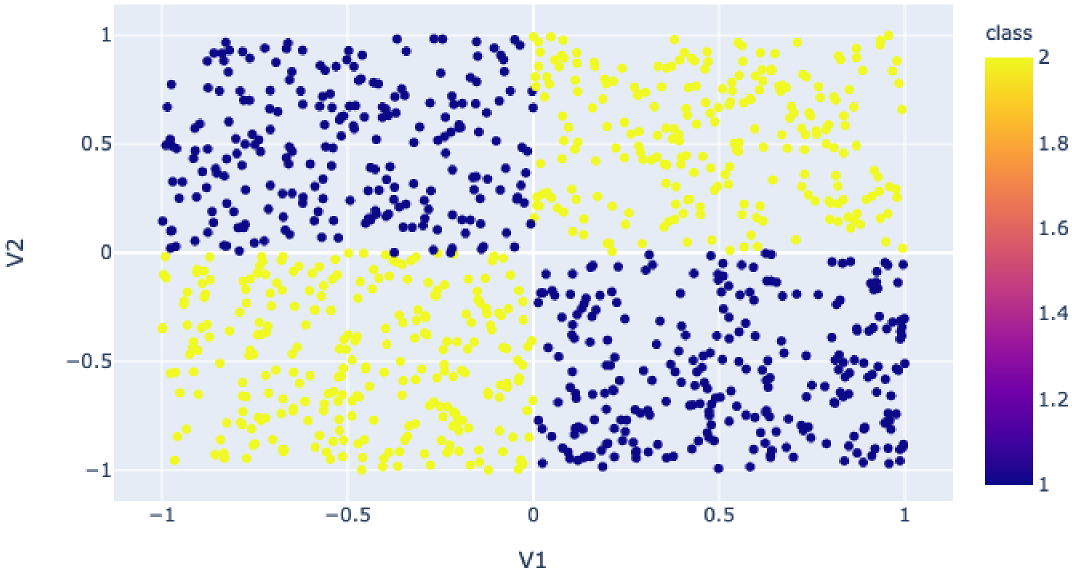
Ideally,



In reality...

https://ai.stackexchange.com/questions/21839/why-isnt-my-decision-tree-classifier-able-to-solve-the-xor-problem-properly

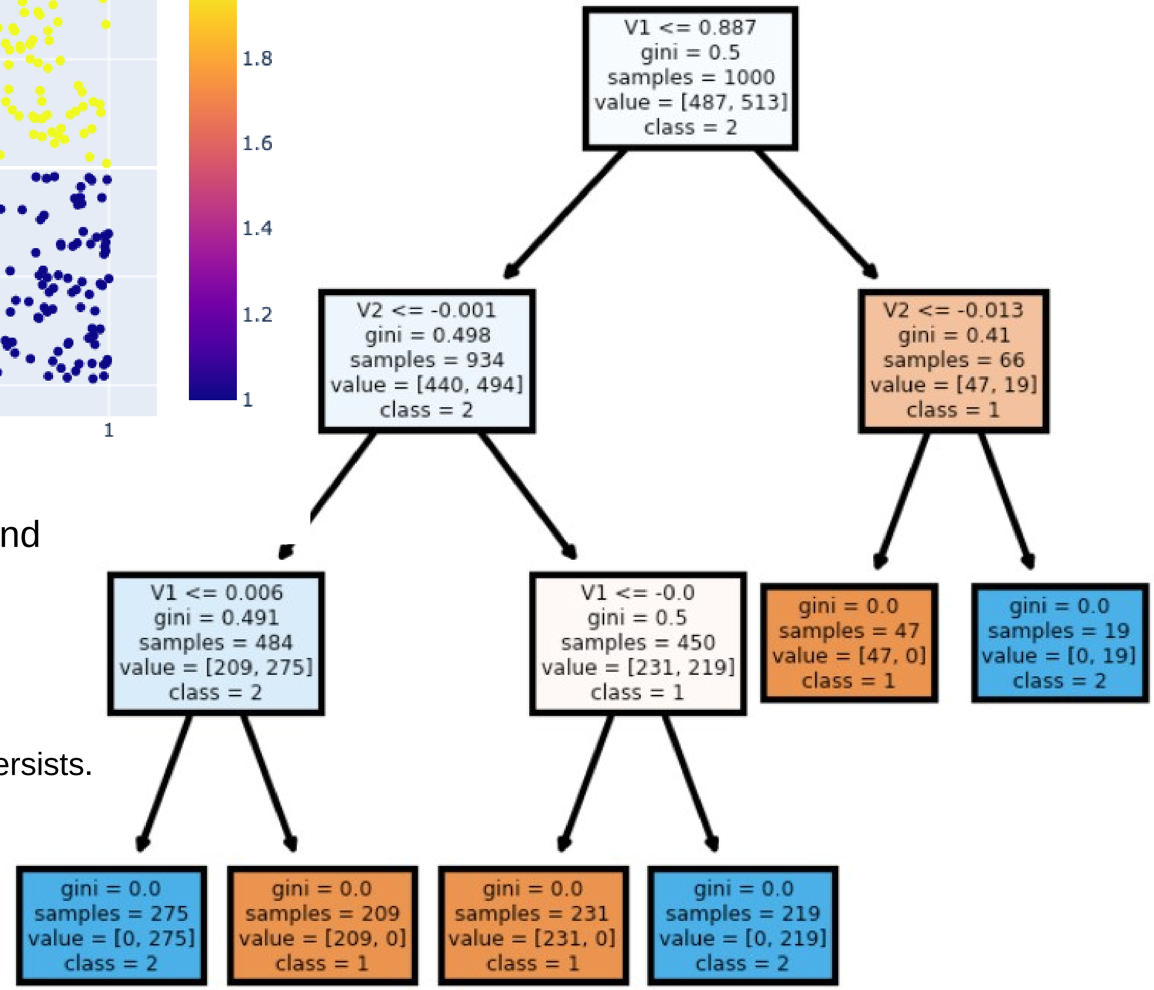
Why isn't my decision tree classifier able to solve the XOR problem properly?

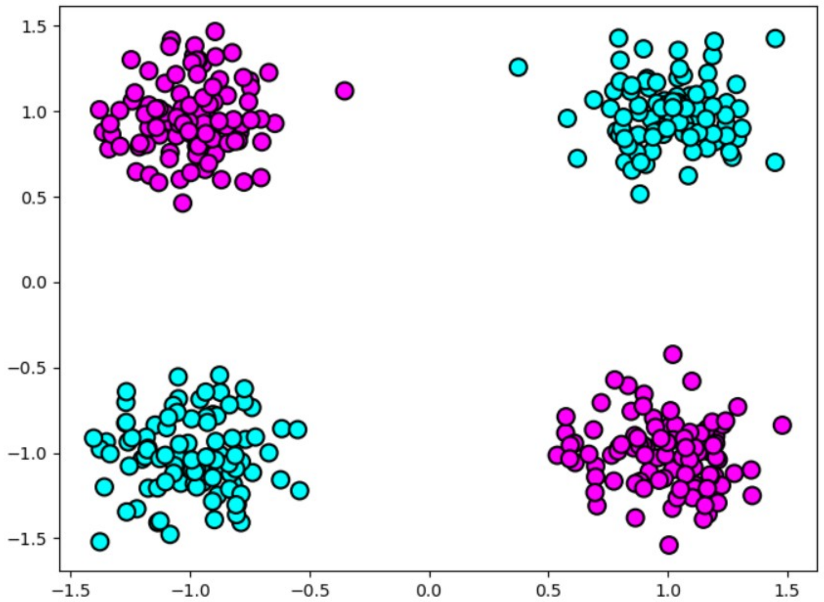


As I understand, the tree should have depth 2 and four leaves.

The first comparison is annoying, because it is close to the right x border (0.887).

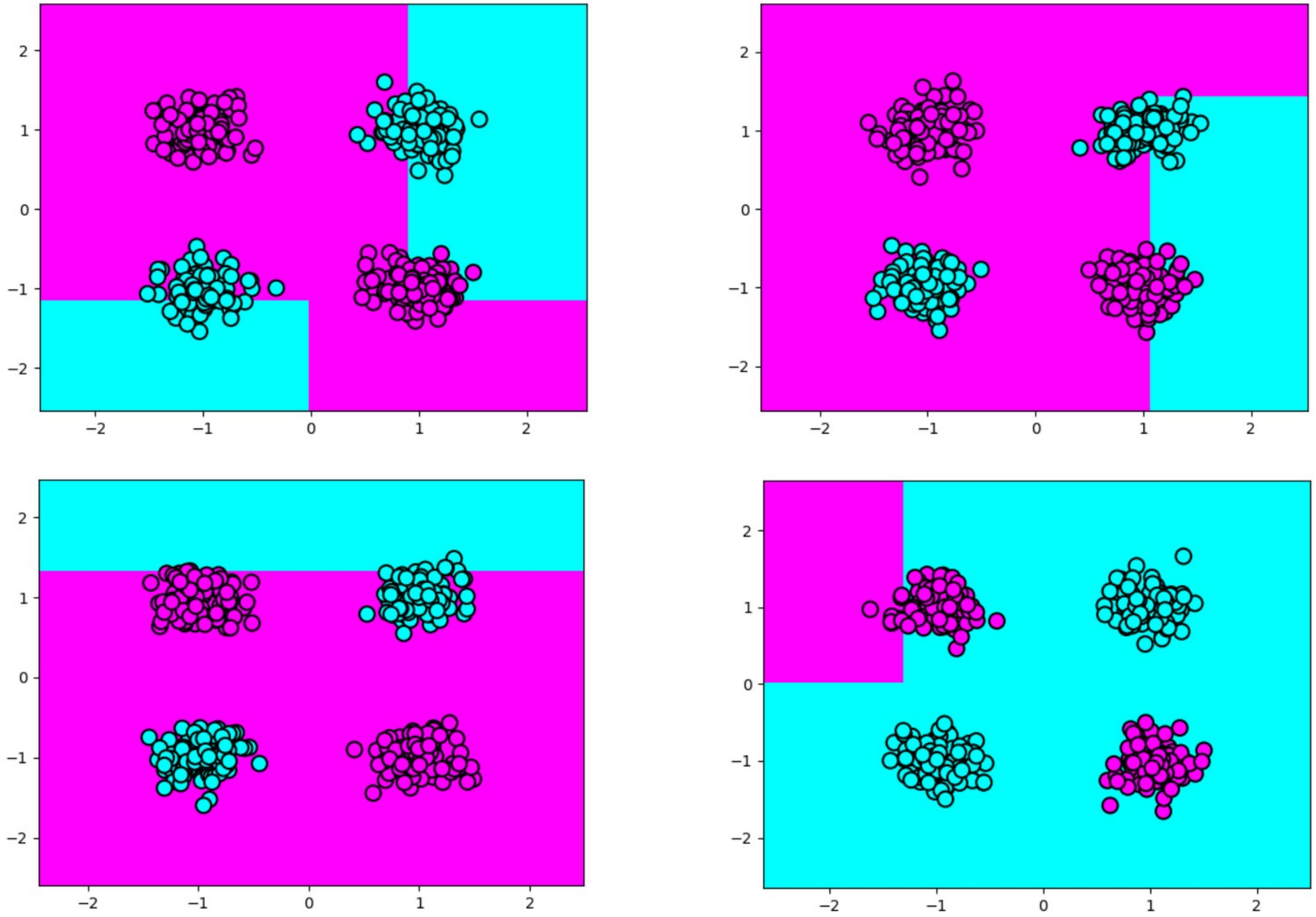
I've tried other parameterizations, but the same result persists.





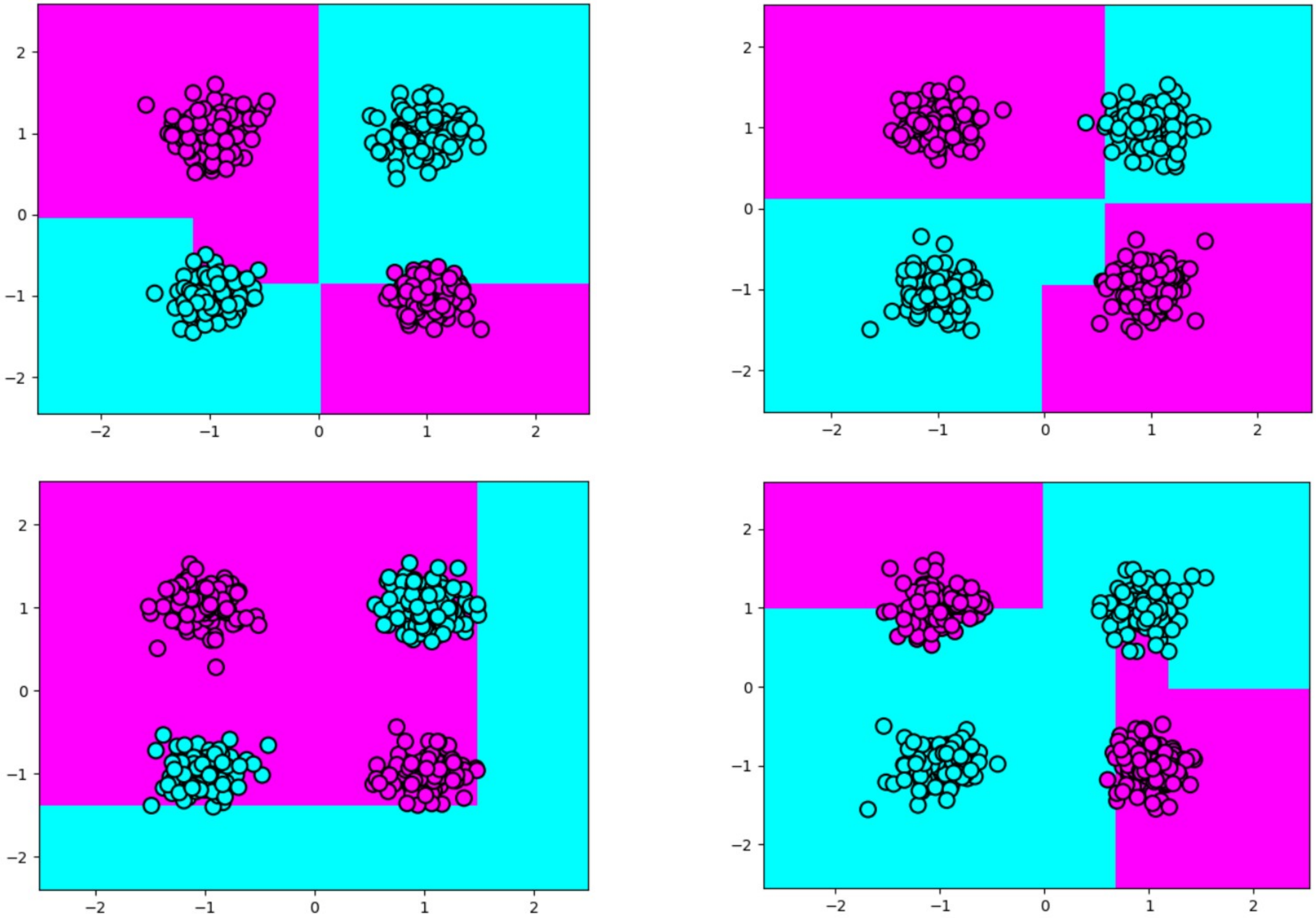

```
from sklearn.tree import DecisionTreeClassifier, plot_tree

def get_grid(data):
    x_min, x_max = data[:, 0].min() - 1, data[:, 0].max() + 1
    y_min, y_max = data[:, 1].min() - 1, data[:, 1].max() + 1
    return np.meshgrid(np.arange(x_min, x_max, 0.01), np.arange(y_min, y_max, 0.01))

clf = DecisionTreeClassifier(criterion='entropy', max_depth=2)

clf.fit(data, labels)

xx, yy = get_grid(data)
predicted = clf.predict(np.c_[xx.ravel(), yy.ravel()]).reshape(xx.shape)
plt.pcolormesh(xx, yy, predicted, cmap='cool')
plt.scatter(data[:, 0], data[:, 1], c=labels, s=100,
            cmap='cool', edgecolors='black', linewidth=1.5);
plt.show()

plt.figure(figsize=(12, 6))
plot_tree(clf, filled=False, fontsize=8)

plt.show()
```

Running this code many times...

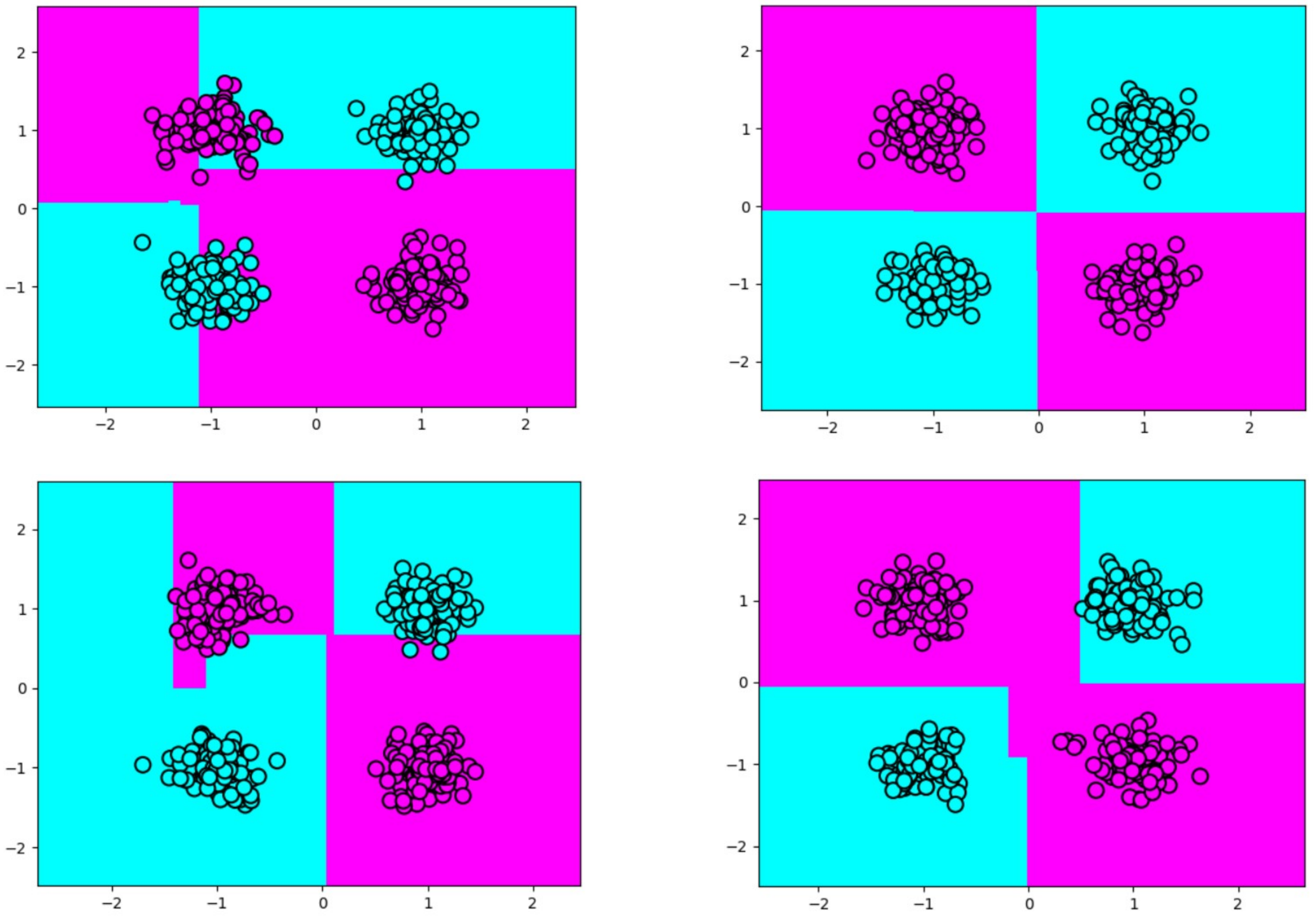


That's why I don't trust the estimates of feature importances based on decision trees.

`max_depth=3`

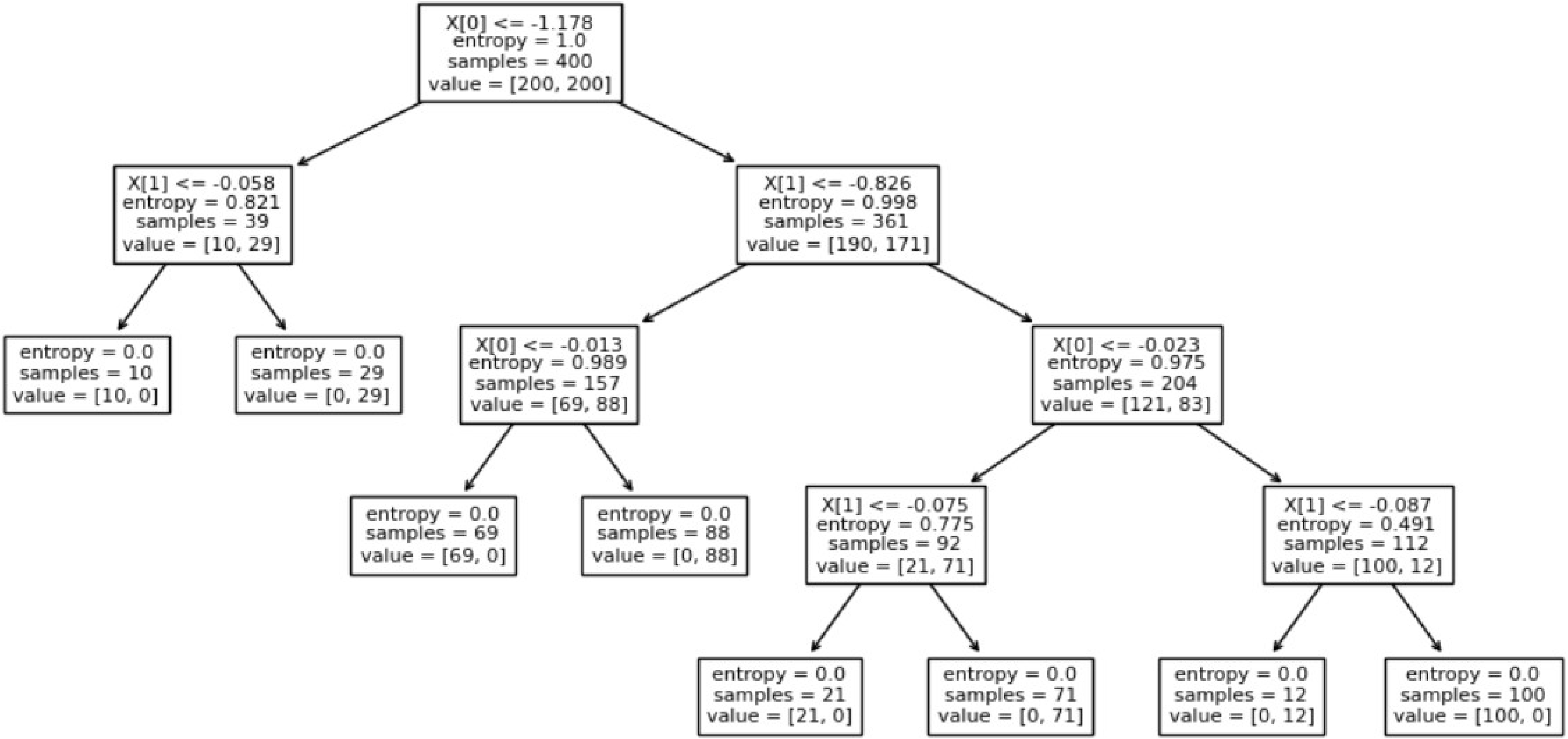


max_depth=4



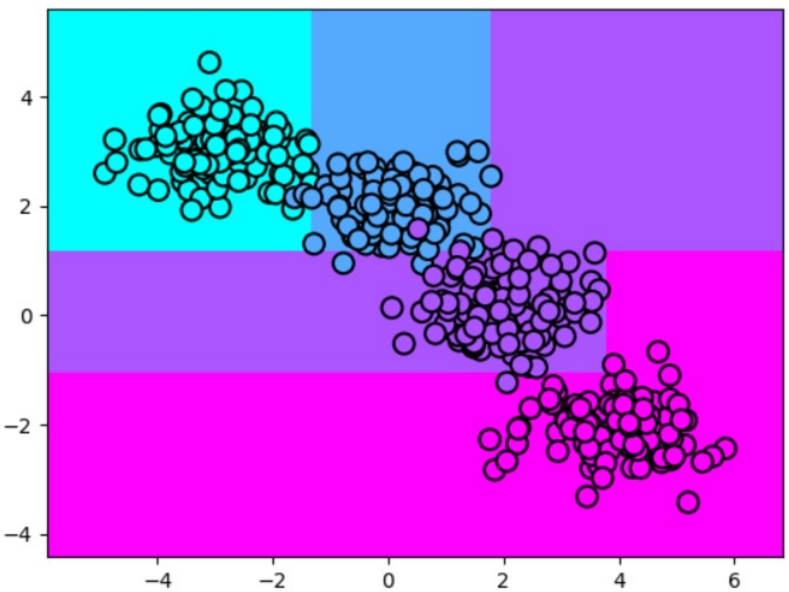
max_depth=4

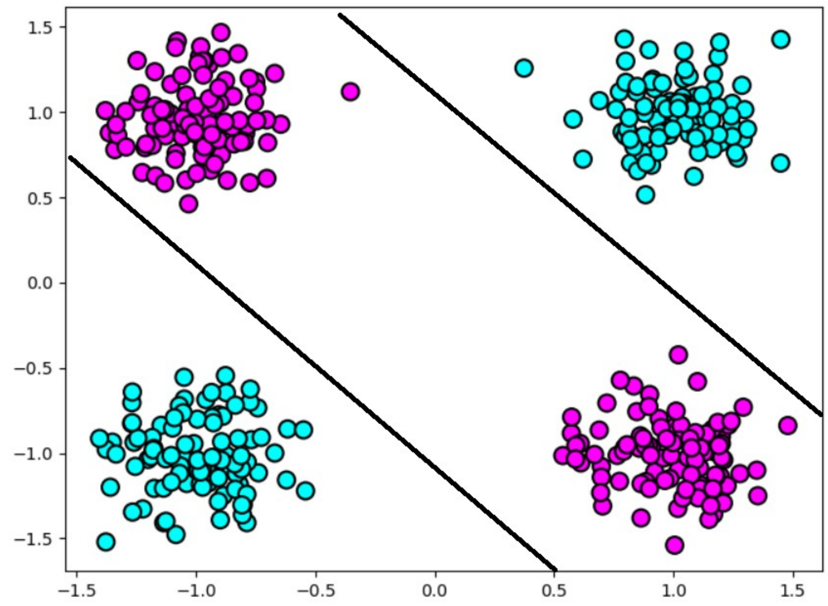
Can you decipher an XOR from this tree?

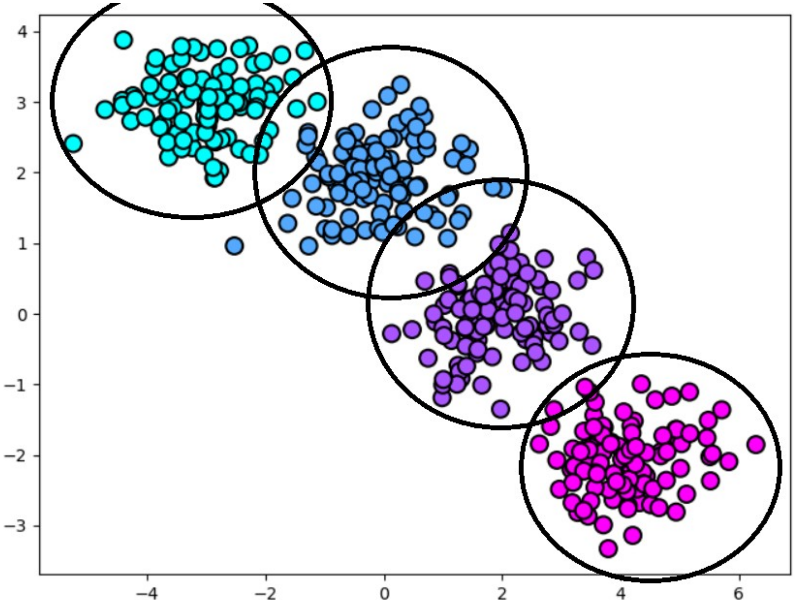


Another problem

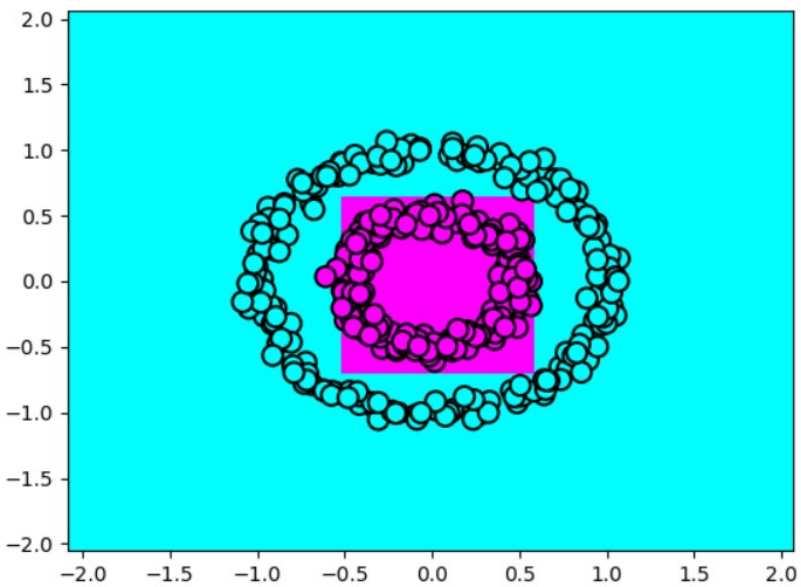
The class borders are parallel to the coordinate axes.

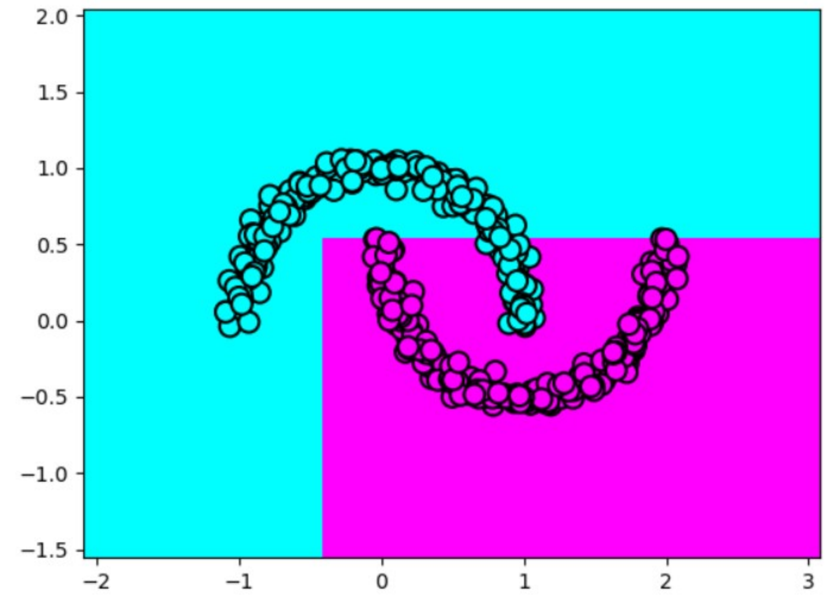


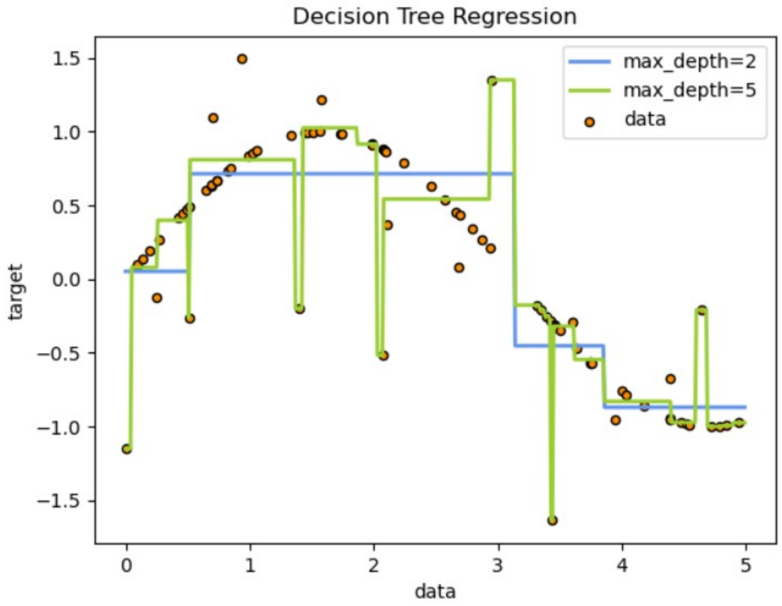


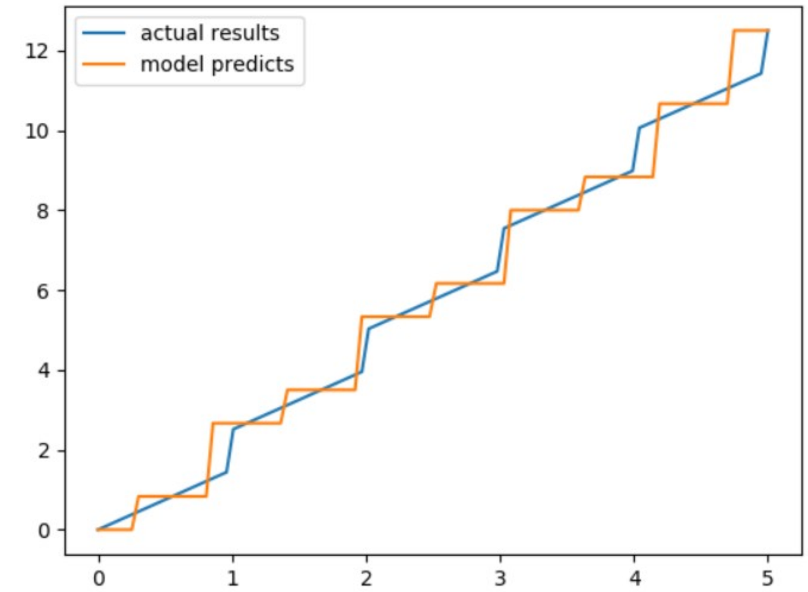


Cannot do these









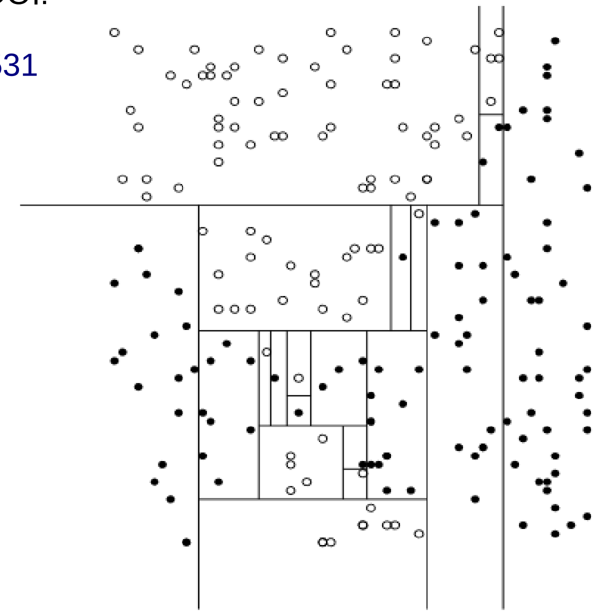
https://scikit-learn.org/stable/modules/tree.html

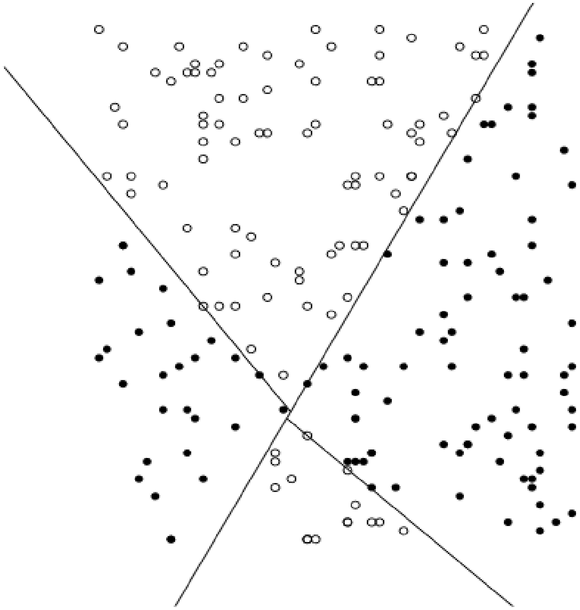
https://datascience.stackexchange.com/questions/65585/decision-tree-with-final-decision-being-a-linear-regression

Oblique decision trees (aka **multivariate**) - the goal is to find a combination of attributes with good discriminatory power.

D. Heath, S. Kasif and S. Salzberg, Induction of oblique decision trees", (1993).
https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.48.9208&rep=rep1&type=pdf

L. Rokach and O. Maimon, Top-down induction of decision trees classifiers-a survey (2005). DOI: 10.1109/TSMCC.2004.843247
https://ieeexplore.ieee.org/document/1522531





Simple tests may result in large trees that are hard to understand,
yet multivariate tests may result in small trees with tests that are hard to understand.

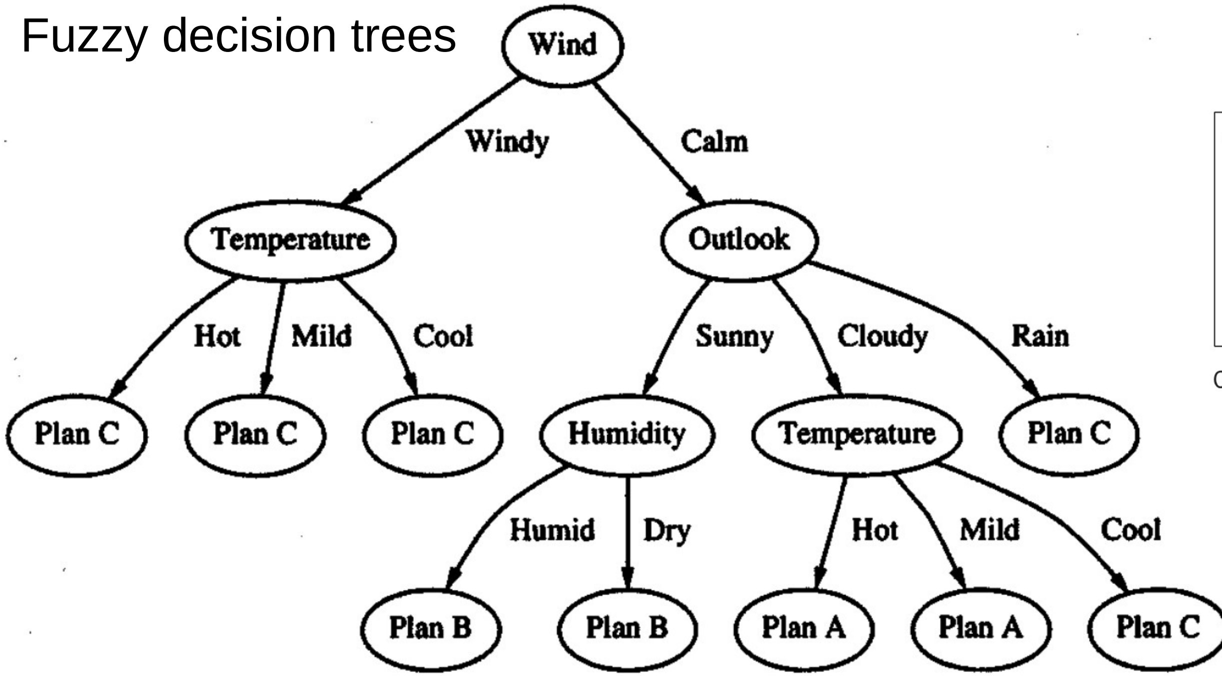
Brodley, C.E., Utgoff, P.E. Multivariate Decision Trees. Machine Learning 19, 45–77 (1995). https://doi.org/10.1023/A:1022607123649

C.T. Yildiz; E. Alpaydin, Omnivariate Decision Trees. (2001) DOI: 10.1109/72.963795 https://ieeexplore.ieee.org/document/963795

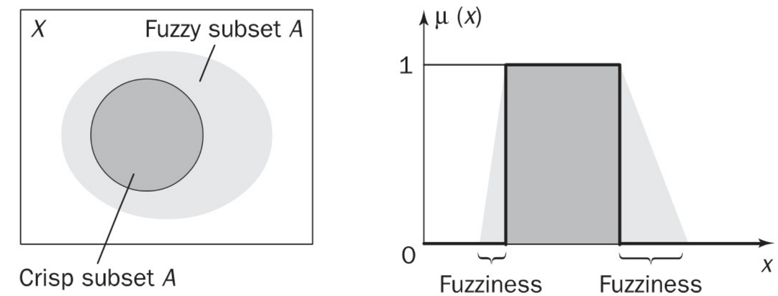
Barros et al. A bottom-up oblique decision tree induction algorithm (2011) DOI: 10.1109/ISDA.2011.6121697
https://ieeexplore.ieee.org/document/6121697

Magana-Mora, A., Bajic, V.B. OmniGA: Optimized Omnivariate Decision Trees for Generalizable Classification Models. Sci Rep 7, 3898 (2017). https://doi.org/10.1038/s41598-017-04281-9

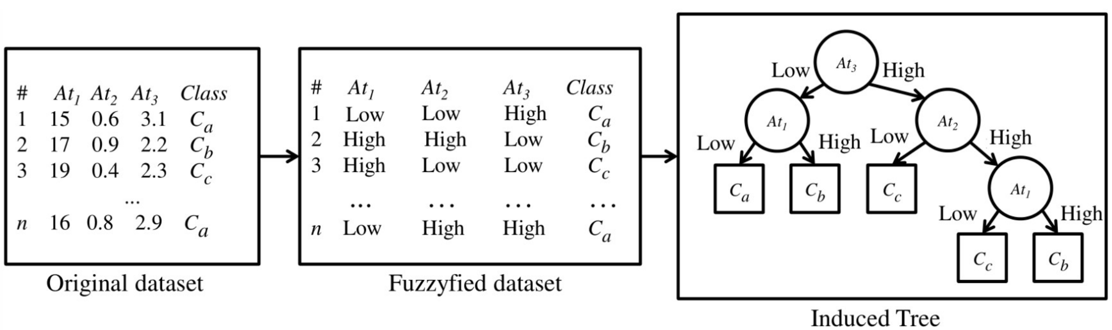
Fuzzy decision trees

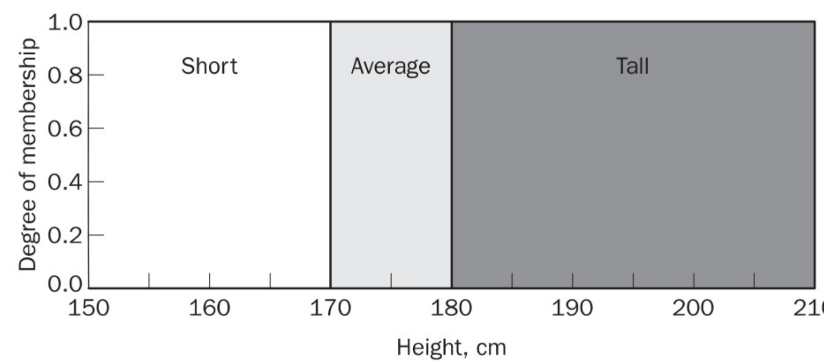


Fuzzy sets



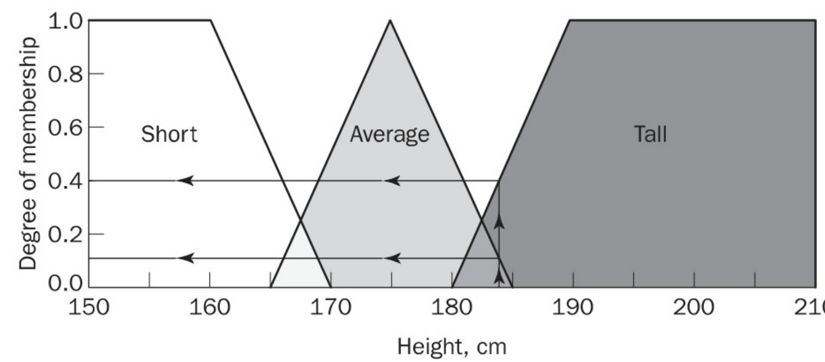
Michael Negnevitsky. Artificial Intelligence.
A Guide to Intelligent Systems. 3rd Ed. (2011)

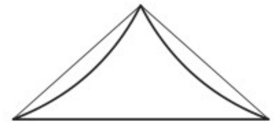
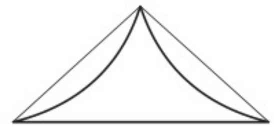
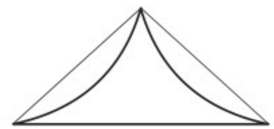
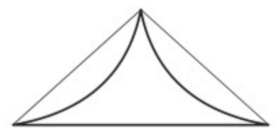
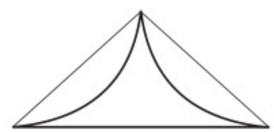
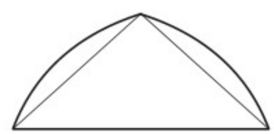
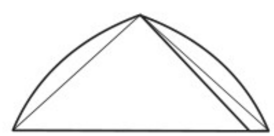
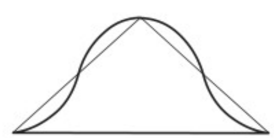




Fuzzy DT calculates a membership degree for the input values in each fuzzy set defining the attributes.

For a classic DT, whenever the input values are located in the decision frontiers, misclassification might occur.



Hedge	Mathematical expression	Graphical representati
A little	$[\mu_A(x)]^{1.3}$	
Slightly	$[\mu_A(x)]^{1.7}$	
Very	$[\mu_A(x)]^{2}$	
Extremely	$[\mu_A(x)]^{3}$	
Very very	$[\mu_A(x)]^{4}$	
More or less	$\sqrt{\mu_A(x)}$	
Somewhat	$\sqrt{\mu_A(x)}$	
Indeed	$2[\mu_A(x)]^2$ if $0 \leqslant \mu_A \leqslant 0.5$ $1-2[1-\mu_A(x)]^2$ if $0.5 < \mu_A \leqslant 1$	

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x)$$

$$\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)]$$

$$\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)]$$

Fuzzy rules

IF x is A
THEN y is B

Rule: 1
IF speed is fast
THEN stopping_distance is long

Rule: 2
IF speed is slow
THEN stopping_distance is short

Jan Łukasiewicz, 1930

Max Black, 1937

Lütfi Ələsgərzadə (Lotfi Zadeh), 1965

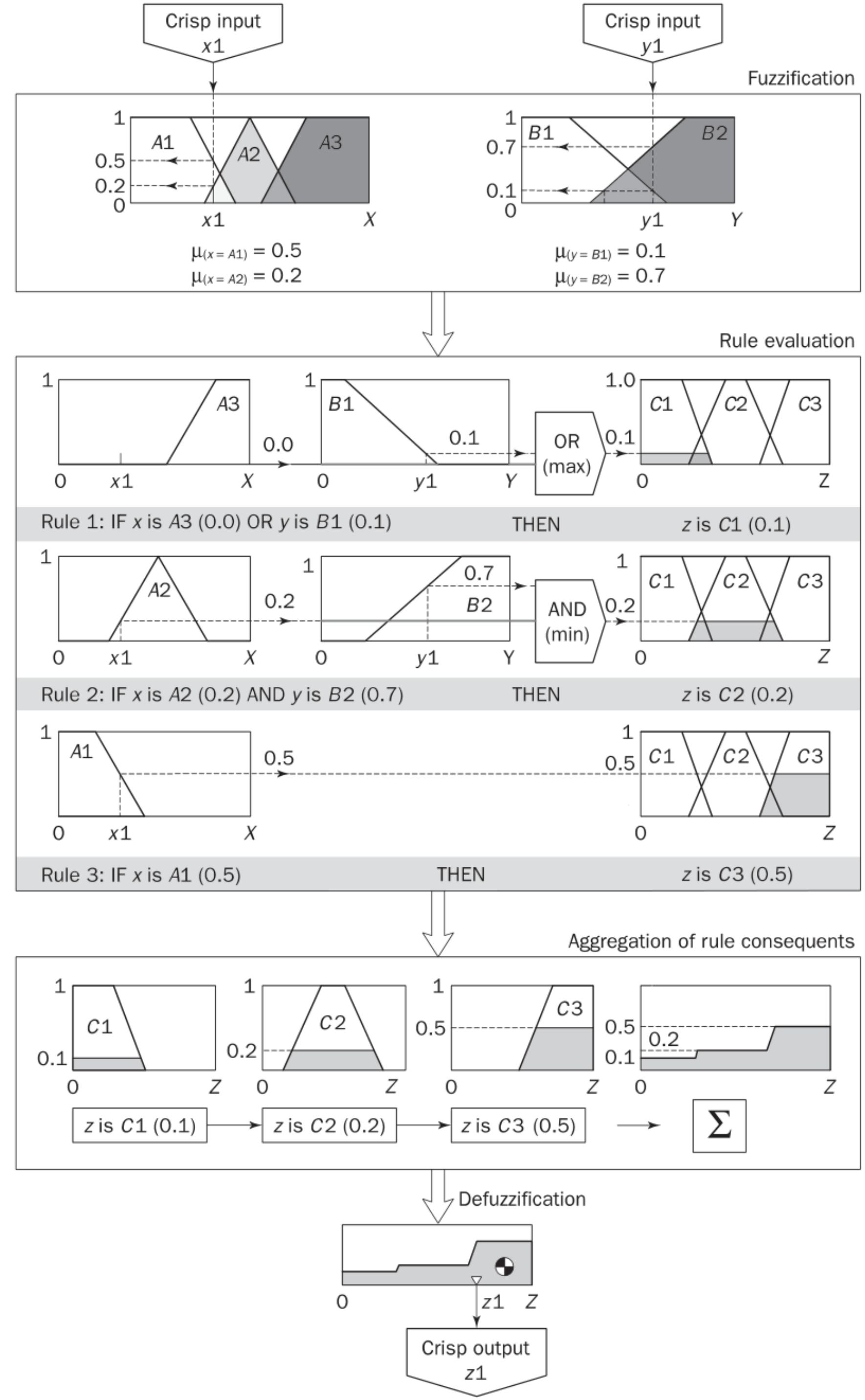
Mamdani-style fuzzy inference

Ebrahim Mamdani, 1975

Rule: 1
IF x is $A3$
OR y is $B1$
THEN z is $C1$

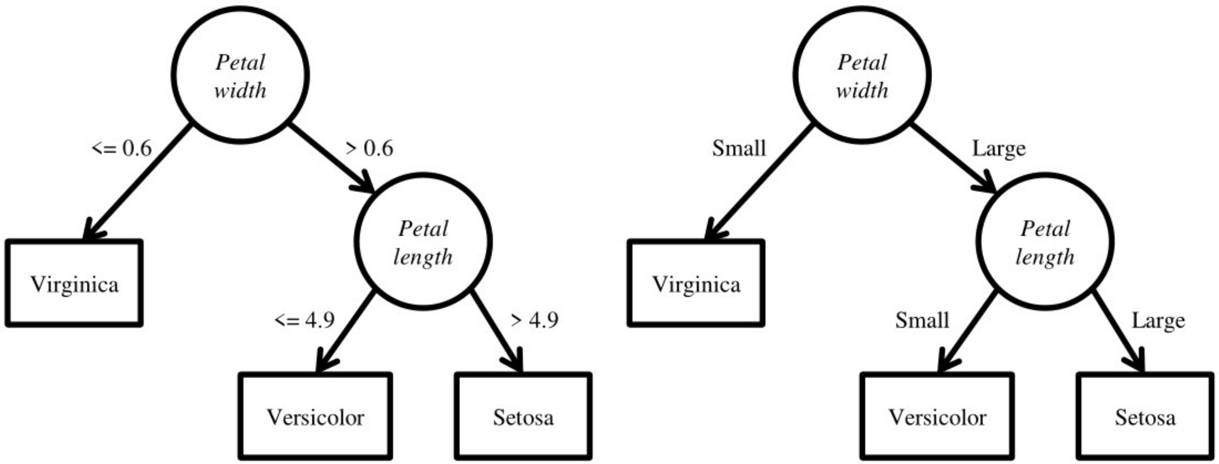
Rule: 2
IF x is $A2$
AND y is $B2$
THEN z is $C2$

Rule: 3
IF x is $A1$
THEN z is $C3$

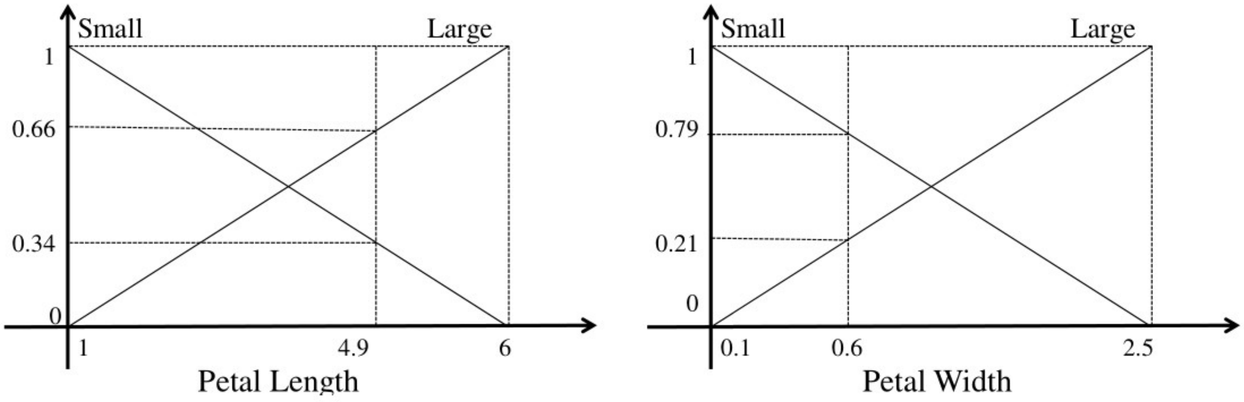


Cintra et al. A Fuzzy Decision Tree Algorithm Based on C4.5 Mathware & Soft Computing Magazine. Vol. 20 n. 1, 56 / 114
http://www.dimap.ufrn.br/~cbsf/pub/anais/2012/10000199.pdf

A classic(left) and a fuzzy (right) decision tree for the Iris dataset

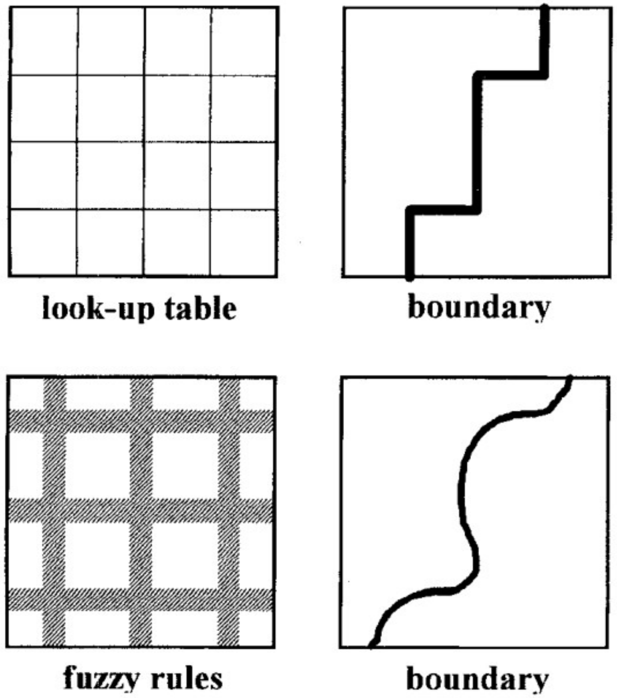


Fuzzy sets defining attributes Petal Length and Petal Width

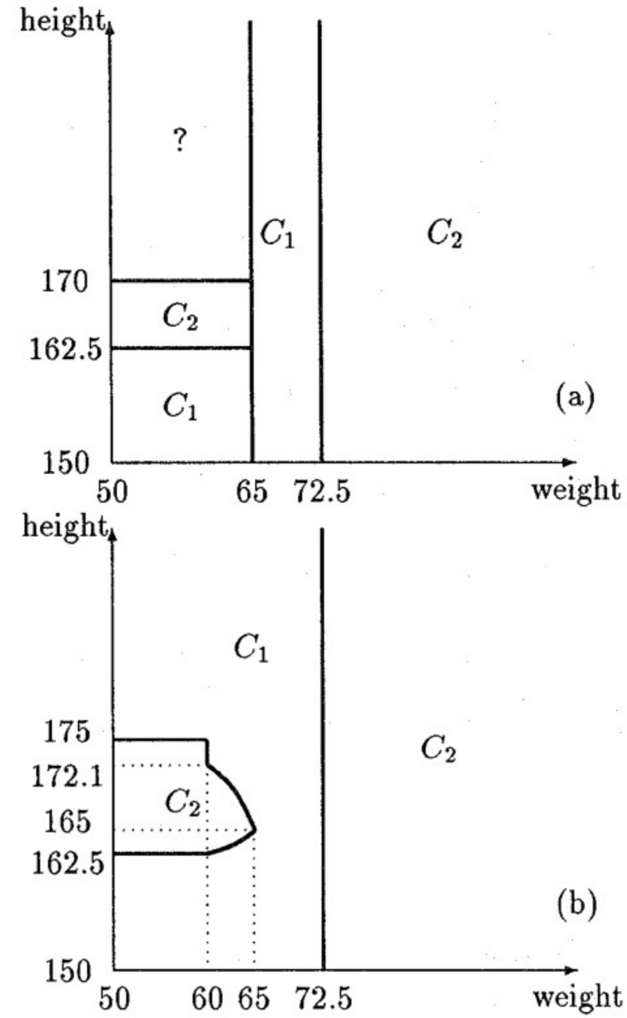


https://cran.r-project.org/web/packages/frbs/index.html

Riza et al. frbs: Fuzzy Rule-Based Systems for Classificationand Regression in R
https://www.jstatsoft.org/index.php/jss/article/view/v065i06/v65i06.pdf



Nozaki et al. Adaptive fuzzy rule-based classification systems.
DOI: 10.1109/91.531768
https://ieeexplore.ieee.org/abstract/document/531768



Zeidler et al. Fuzzy decision trees and numerical attributes. DOI: 10.1109/FUZZY.1996.552312 https://ieeexplore.ieee.org/document/552312

Fuzzy logic:

https://www.mathworks.com/help/fuzzy/an-introductory-example-fuzzy-versus-nonfuzzy-logic.html

https://www.mathworks.com/help/fuzzy/what-is-fuzzy-logic.html

https://www.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html

https://www.mathworks.com/help/fuzzy/fuzzy-inference-process.html

https://www.mathworks.com/help/fuzzy/types-of-fuzzy-inference-systems.html

https://www.mathworks.com/help/fuzzy/membership-function-gallery.html

https://www.mathworks.com/help/fuzzy/defuzzification-methods.html

https://fuzzytech.com/

The Fuzzy Lattice Reasoning

https://weka.sourceforge.io/packageMetaData/fuzzyLaticeReasoning/index.html

Fuzzy Unordered Rule Induction Algorithm

https://weka.sourceforge.io/packageMetaData/fuzzyUnorderedRuleInduction/index.html

Classifier for learning Functional Trees

https://weka.sourceforge.io/packageMetaData/functionalTrees/index.html

HotSpot learns a set of rules (displayed in a tree-like structure) that maximize/minimize a target variable/value of interest.

https://weka.sourceforge.io/packageMetaData/hotSpot/index.html

Decision tree learner based on imprecise probabilities and uncertainty measures.

https://weka.sourceforge.io/packageMetaData/JCDT/index.html

Joaquín Abellán and Serafín Moral. Building classification trees using the total uncertainty criterion. International Journal of Intelligent Systems 18.12 (2003) 1215-1225. doi: 10.1002/int.10143

Multi-objective evolutionary fuzzy classifier

https://weka.sourceforge.io/packageMetaData/MultiObjectiveEvolutionaryFuzzyClassifier/index.html

Jimenez, F., Sanchez, G. & Juarez, J.M. (2014). Multi-objective evolutionary algorithms for fuzzy classification in survival prediction. Artificial Intelligence in Medicine, 60(3), 197-219.

Binary-class alternating decision trees and multi-class alternating decision trees

https://weka.sourceforge.io/packageMetaData/alternatingDecisionTrees/index.html

Freund, Y., Mason, L.: The alternating decision tree learning algorithm. In: Proceeding of the Sixteenth International Conference on Machine Learning, Bled, Slovenia, 124-133, 1999.

Geoffrey Holmes, Bernhard Pfahringer, Richard Kirkby, Eibe Frank, Mark Hall: Multiclass alternating decision trees. In: ECML, 161-172, 2001.

Alternating Model Trees https://weka.sourceforge.io/packageMetaData/alternatingModelTrees/index.html

Eibe Frank, Michael Mayo, Stefan Kramer: Alternating Model Trees. In: Proceedings of the ACM Symposium on Applied Computing, Data Mining Track, 2015.

RIpple-DOwn Rule learner https://weka.sourceforge.io/packageMetaData/ridor/index.html

Brian R. Gaines, Paul Compton (1995). Induction of Ripple-Down Rules Applied to Modeling Large Databases. J. Intell. Inf. Syst. 5(3):211-228