



Decision Trees With Python

By Giulia Osti

25.05.2023 / #WAIpractice

Hello There!



Position

PhD candidate @ UCD

Research Topic

AI & digital curation/heritage

Education

former archaeologist (!)

Linkedin

godigitalcorner

Twitter

@semanticnoodles

Email

giulia.osti@ucdconnect.ie

Today's Takeaways

- Understand the limitations and potential of decision trees
- Become familiar with key concepts and mechanisms
- Learn how to build and evaluate decision trees in Python using scikit-learn
- Know alternative tools for tree building and visualisation

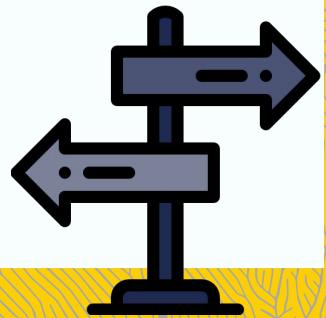
What are 'Decision Trees'?

A group of ML algorithms often referred to as CART (classification and regression trees).

They allow to solve decision problems by categorising object according to their decision features.

Use Cases:

- Product analysis
- Botany (!)
- Exploratory analysis



Potentials & Limitations



Advantages:

- Highly visual & explainable
- Almost zero data processing
- Reliable (you can validate them)
- Can handle mixed data
(categorical & ordinal)

Caveats:

- Sensitive to variance
- Careful reasoning on categorical variables
- One tree does not make a forest



Tools of the Trade



Python



Scikit-learn suite

[Main portal](#)

[Documentation](#)



Google Colab



Our Dataset



Alternatives To Python

WEKA



Fully interfaced ML free software
(no coding required) for ‘small’ datasets

[Main portal](#) | [Documentation](#)

[WEKA manual](#)



R Studio



Integrated Development Environment
free, multiplatform (supports python!)

[Main portal](#) | [User Guide](#)

[**rpart.plot package**](#) on CRAN



The screenshot shows the RStudio interface. The top menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Window, Help, and a date/time stamp. The main window has tabs for 'RStudio' and 'R'. The left pane contains a script editor with R code for extracting text from PDF files. The right pane shows the 'Environment' tab with a global environment and a 'Data' pane listing 'terms0'. The bottom pane shows the 'System Library' with various R packages listed.

```
16 headpdffiles_17
17 headpdffiles_23
18
19 # Define a discrete function
20 getTxt <- function(file.list){
21   for (pdf_file in file.list){
22     file_name <- file.path_sans_ext(basename(pdf_file))
23     pdf_text <- pdf_text(pdf_file)
24     extracted_text <- strsplit(pdf_text, collapse = "\n\n")
25     txt_file <- file(file_name, ext('.txt'))
26     txt_file <- paste0(txt_file, ".txt")
27     writeLines(extracted_text, txt_file)
28     cat("Text extracted from", pdf_file, "and saved as", txt_file, "\n")
29   }
30 }
31
32 # Run it for PS2017
33 getTxtpdffiles_17
34
35 # Run it for PS2023
36
37 (top Level :  
Console Background jobs :  
R 4.3.0 (2023-04-17))
```

R is a collaborative project with many contributors. Type 'contributors()' for more information and 'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or 'help.start()' for an HTML browser interface to help. Type 'q()' to quit R.

[Workspace loaded from ~/RData]

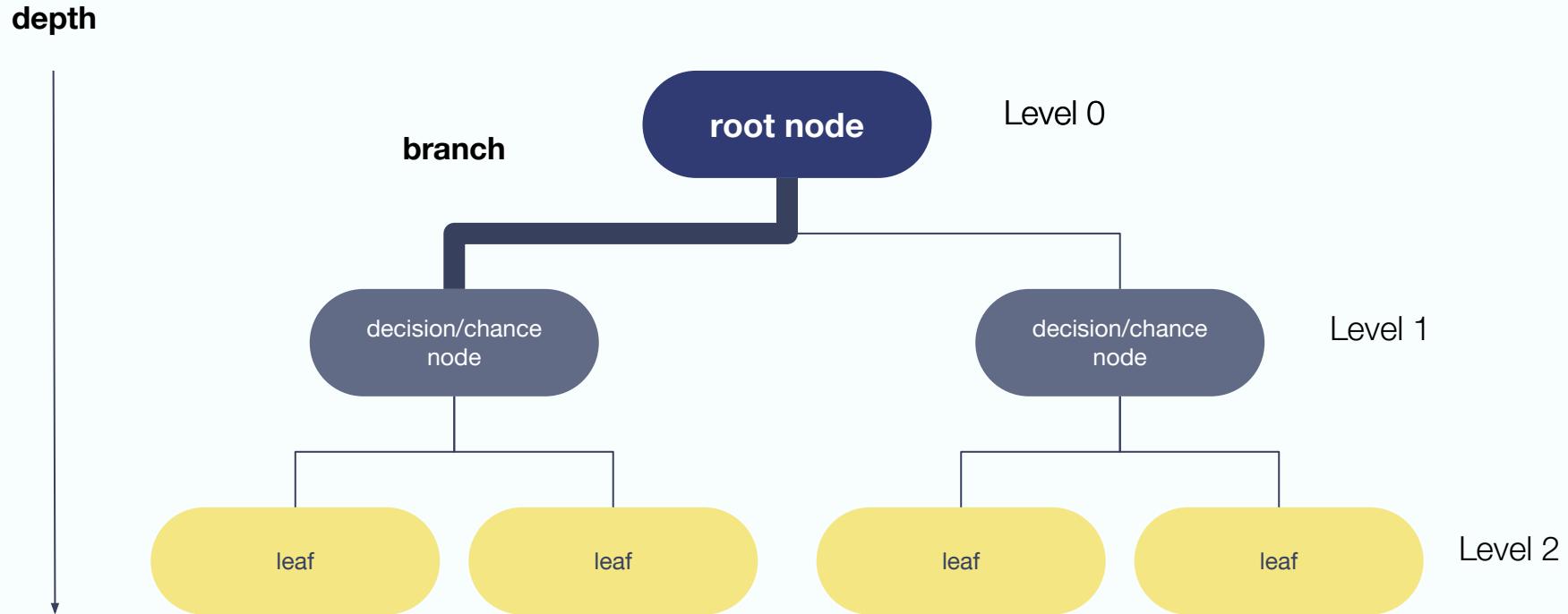
System Library	Description	Version
askpass	Safe Password Entry for R, C, and SSH	1.1
backports	Reimplementations of Functions introduced Since R-3.0.0	1.4.1
base	The R Base Package	4.3.0
base64enc	Tools for Base64 Encoding	0.1.3
bit	Classes and Methods for Fast Memory-Efficient Boolean	4.0.5
bit64	A S3 Class for Vectors of 64bit Integers	4.0.5
blob	A Simple S3 Class for Representing Vectors of Binary Data	1.2.4
boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-24.1
broom	Templating Framework for Report Generation	1.0-8
bslib	BSlib: Bootstrap Themes for shiny and	1.1.0
cachem	Cache R Objects with Automatic Pruning	1.0.8
callr	Call R from R	3.7.3
cellranger	Count Cell Ranges to Rows and Columns	7.3-21
class	Functions for Classification	7.3-11
cli	Helps for Developing Command Line Interfaces	3.6.1
dplyr	Read and Write from the System Clipboard	0.8.0
forcats	Tools for Working with Data: Character Analysis Extended	0.2.0
gridextra	Bringing ggplot2 to the Grid	2.1.4



A quick walk through some

Key Concepts

Anatomy of a Decision Tree



Essential Glossary



Root node

The topmost node in a decision tree. It represents the entire dataset and serves as the starting point for the tree's recursive partitioning process.

Pruning

The process of reducing the complexity of a decision tree by removing nodes and branches.

Chance node

A node in a decision tree that represents a situation where multiple outcomes are possible.

Weighting

Strategy to handle imbalanced classes, or assigning different weights to samples or classes.

Leaf

A terminal node in a decision tree. It represents the final prediction or outcome.

Depth

It represents the number of decisions or splits required to reach a prediction.



A Potential Use Case: Self-Tracking



About

A fictional dataset containing records of some behavioural habits related to productivity.

Task

Build a decision tree to get some insights about my focusing techniques of choice.

Spoiler: we are going to fail!

Focusing techniques

- Deep Work, almost hyperfocusing on a single task until it's done.
- Task Batching, completion of 1+ similar tasks jointly.
- Eat the Frog, run prioritised tasks depending on the difficulty.
- Zen to Done, just do it.





It's Python time, so

Let's Get Started

Conclusive remarks



- Decision trees work with ‘simple’ data. If you want to handle complexity you need to understand deeply how to weight and prune properly (doing some gardening).
- Make it simple for the machine to understand.
- Experiment with anything you are familiar with.
- Real world-like datasets bite, but do not give up!



Some extra resources



Here are some resources to get you started with:

- [Data wrangling with Python](#)
- [Decision Trees in Python](#)

- [Data wrangling with R](#)
- [Decision trees in R](#)

[Datacamp](#) and [Udemy](#) are actually offering nice courses on the topic - paid but for all the pockets!

Quick Recap

- Understand the limitations and potential of decision trees
- Become familiar with key concepts and mechanisms
- Learn how to build and evaluate decision trees in Python using scikit-learn
- Know alternative tools for tree building and visualisation



**Thank You
For Joining!**

Any Questions?