

LDSSA

SLU11 - Tree-based Models

October 1st and 2nd, 2022

Before we start

Ask and vote for questions through <https://www.sli.do/> with the code:

#7815174





1. Introduction

Overview

Objective: build an intuition on different tree-based models and techniques

We will cover:

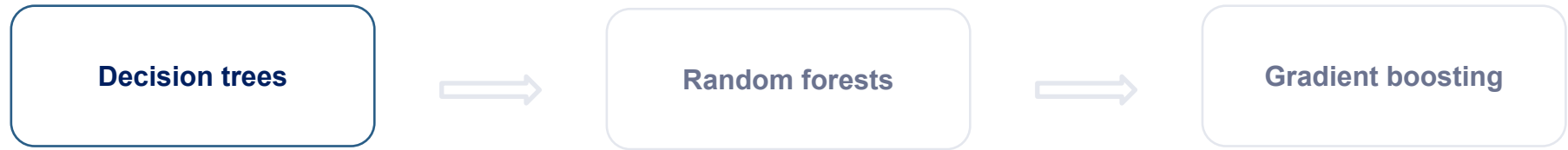
- Rule-based systems
- Decision trees
- Ensemble learning: Bagging (random forests) and Gradient Boosting

Motivation

- Tree-based models represent knowledge as a set of rules
- Ensemble methods:
 - Combine basic models to create more powerful models
 - Avoid overfitting
- Tree-based models are very widely used and very flexible



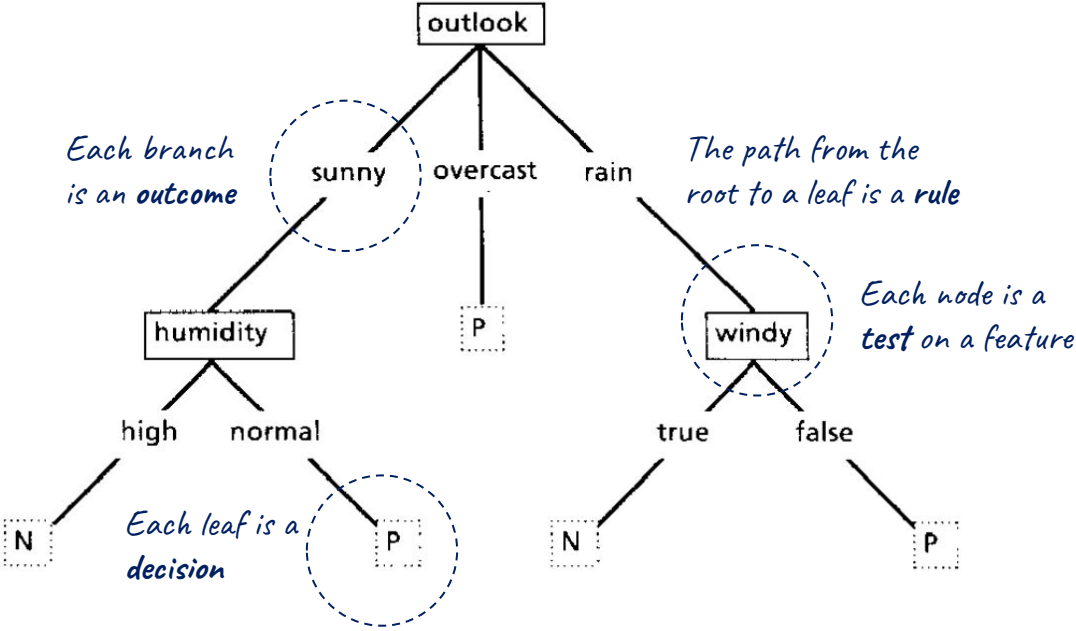
2. Topic Explanation



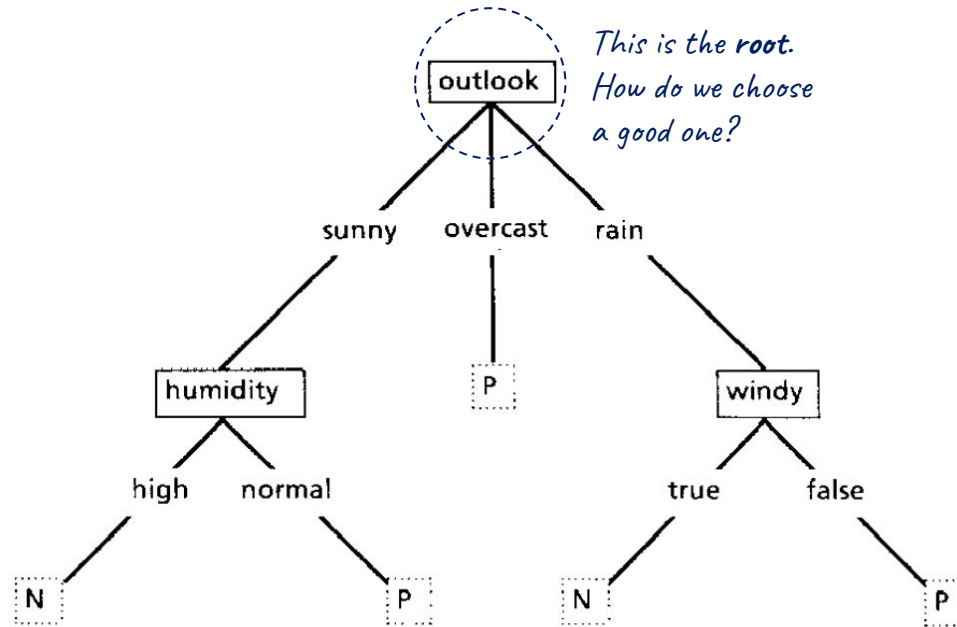
Decision Trees

First 5 rows from the data table:

	Outlook	Temperature	Humidity	Windy	Class
0	sunny	hot	high	false	0
1	sunny	hot	high	true	0
2	overcast	hot	high	false	1
3	rain	mild	high	false	1
4	rain	cool	normal	false	1



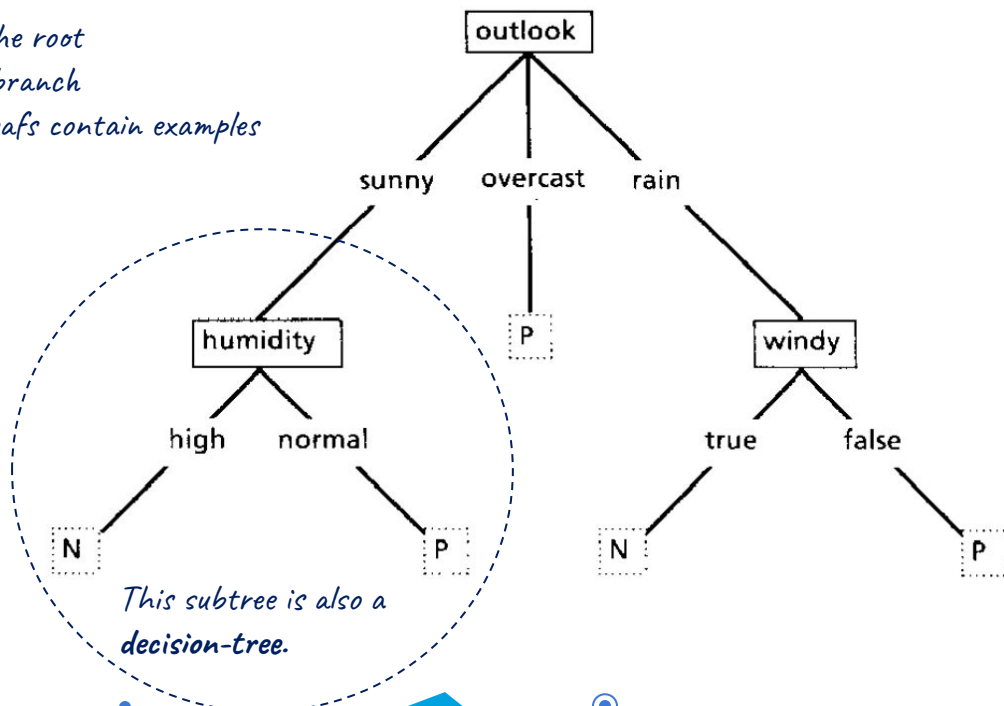
What would be the best root?



Building a decision tree

Algorithm overview:

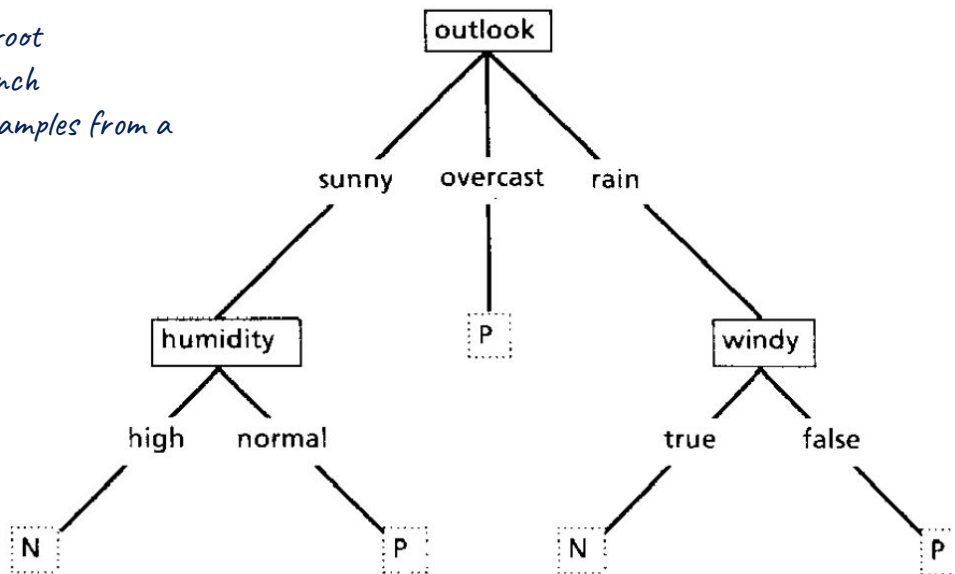
1. Pick the best attribute as the root
2. Create a sub-tree for each branch
3. Repeat recursively until all leafs contain examples from a single class



What could go wrong, really?

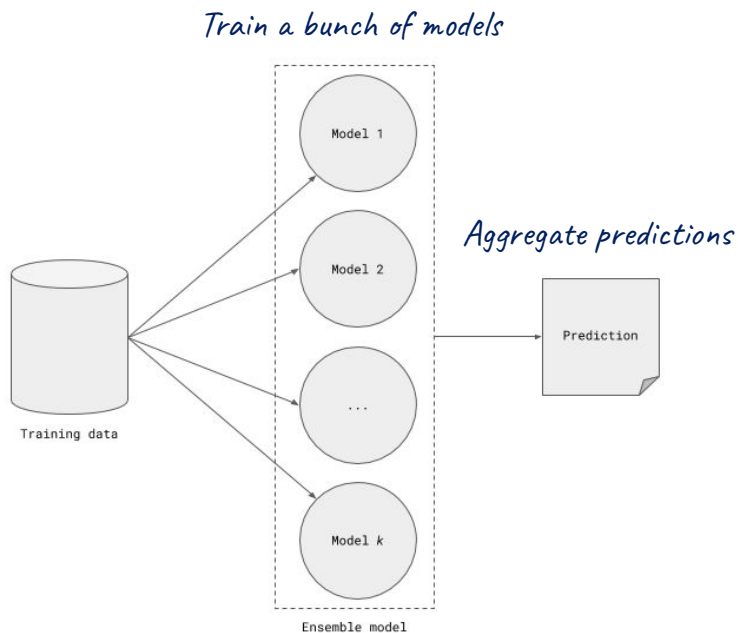
Algorithm overview:

1. Pick the best attribute as the root
2. Create a sub-tree for each branch
3. Repeat until all leafs contain examples from a single class

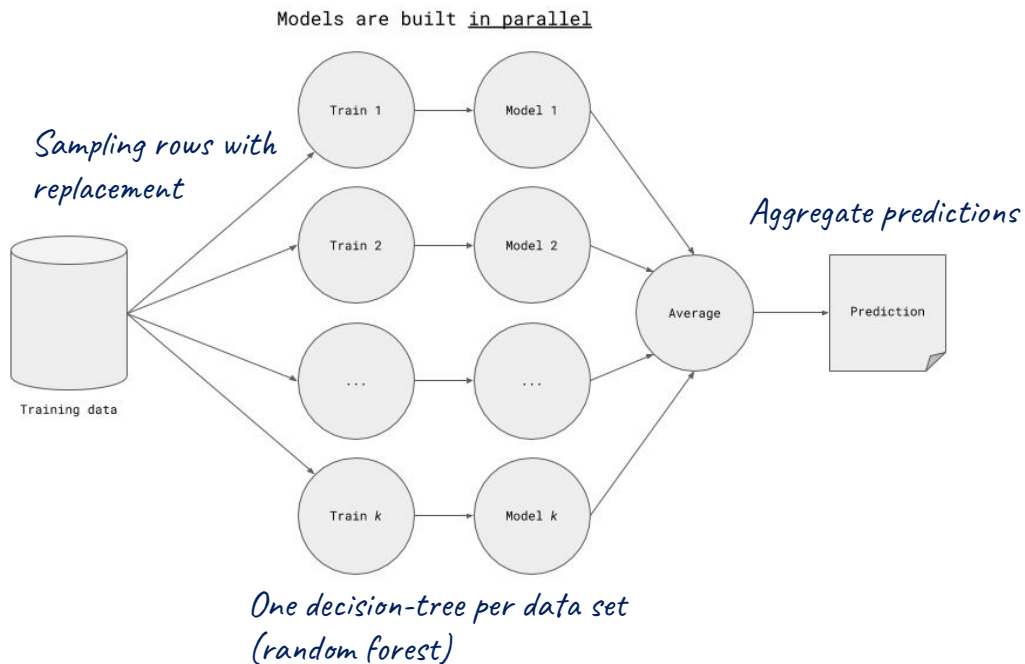




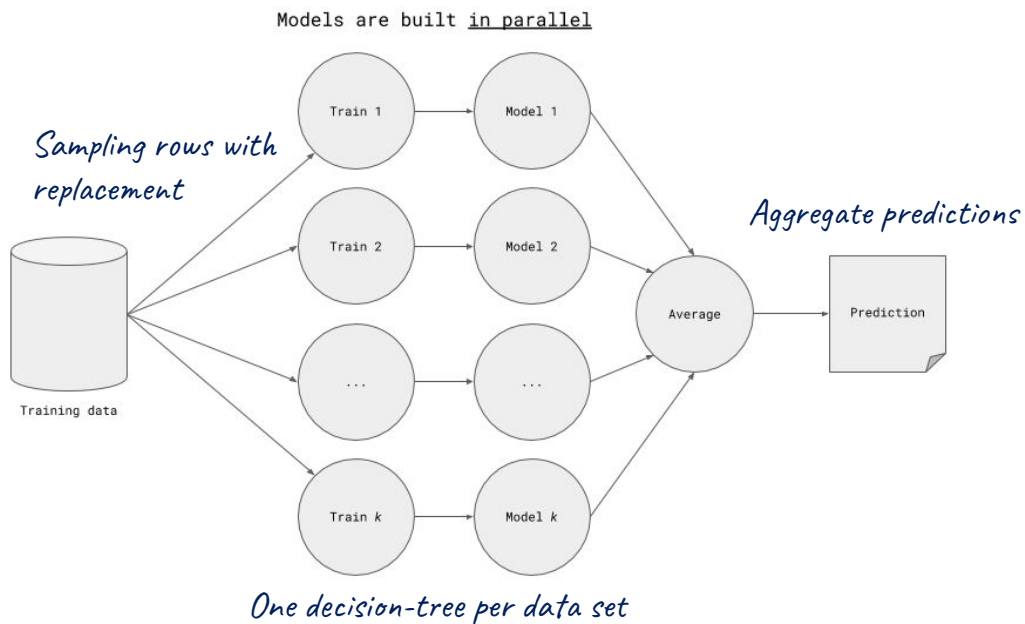
Many minds are better than one



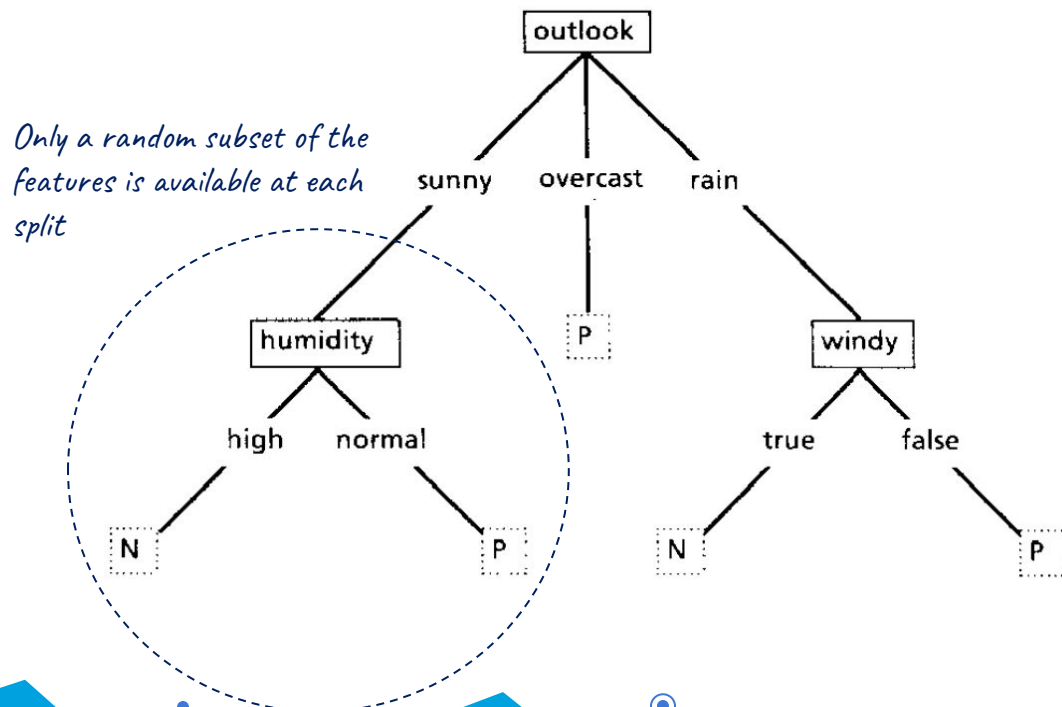
Bootstrap aggregating or bagging



With random feature selection

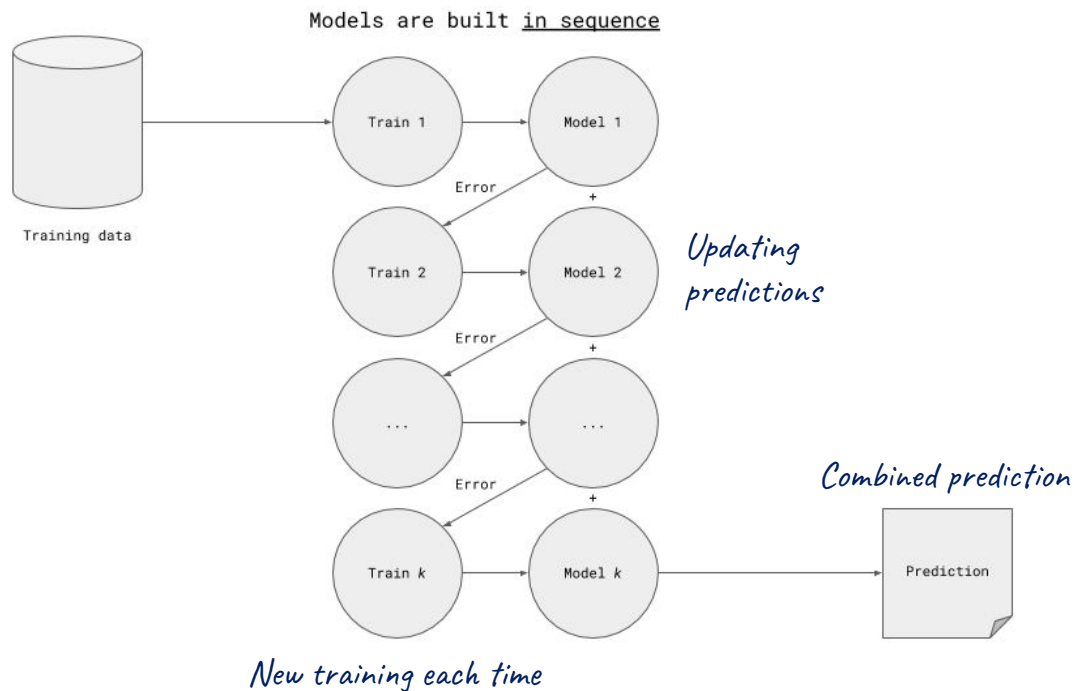


With random feature selection

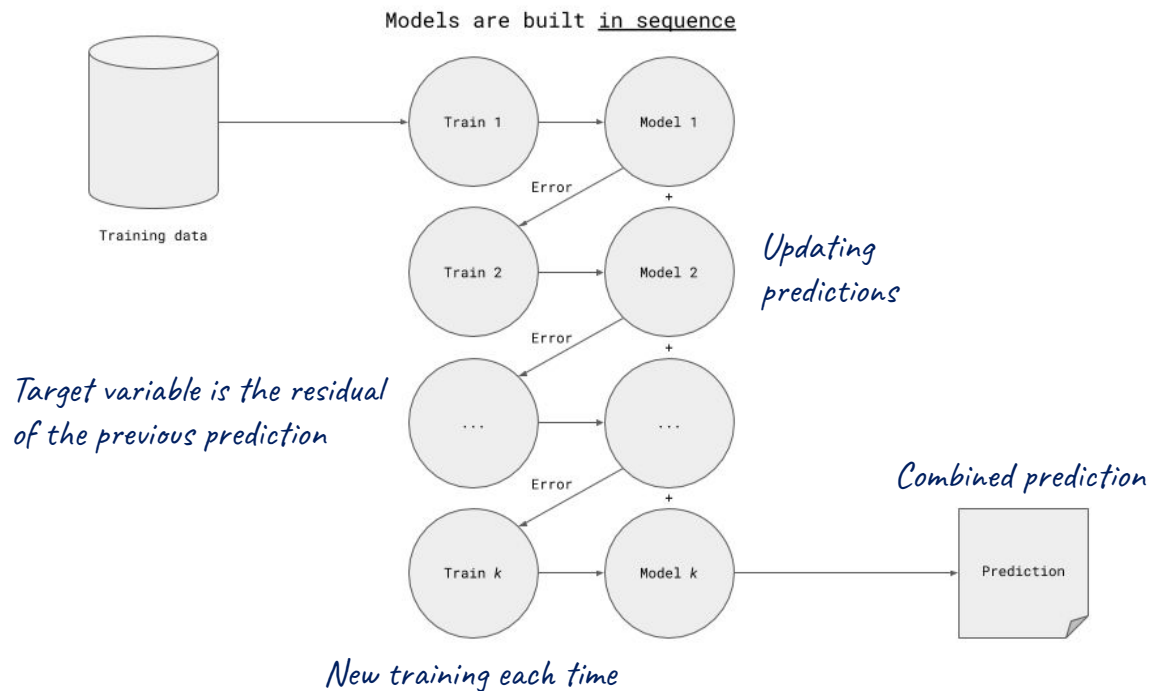




Each tree uses the previous tree



Fitting on pseudo-residuals





3. Recap

Recap

- Decision-trees are a way to learn sets of rules from a training set, but **prone to overfitting**
- Ensemble learning techniques can be used to combine the predictions of several models, and reduce overfitting
- Random forests use row sampling and column sampling to build several trees in parallel and **control overfitting**
- Gradient boosting build shallow trees sequentially to **reduce the bias** of the combined estimator, by making each tree specialize in correcting past errors
- Both random forests and gradient boosting improve the performance of decision-trees, at the expense of interpretability



4. Q&A

#7815174