Decision Trees

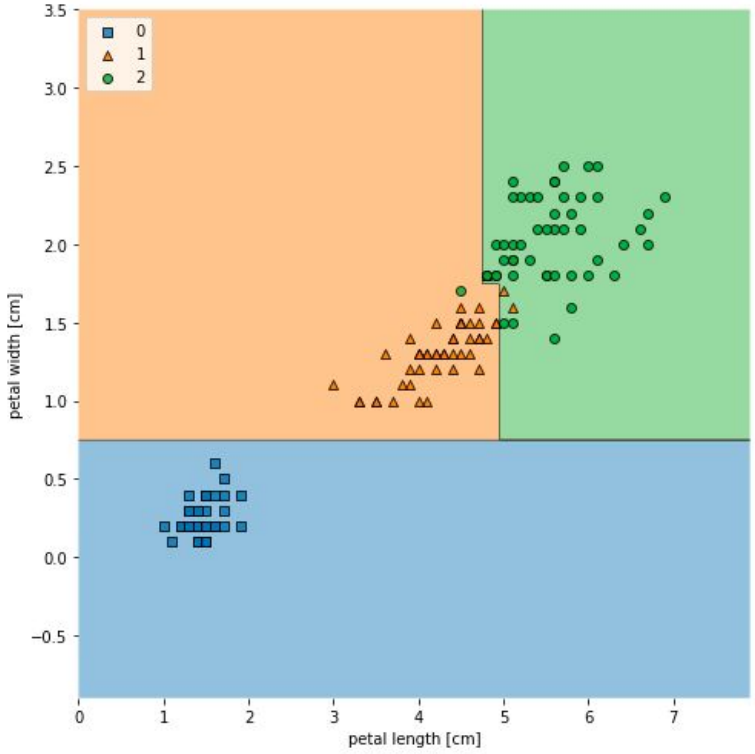
1. Decision Tree Overview
2. Training Decision Trees
3. Pros and Cons

Decision Tree Overview

How Decision Trees Work

- The basic idea of a decision tree is to partition our data into rectangular regions and represent this partition as a tree structure
- We want to make this partition so that each section is as similar as possible

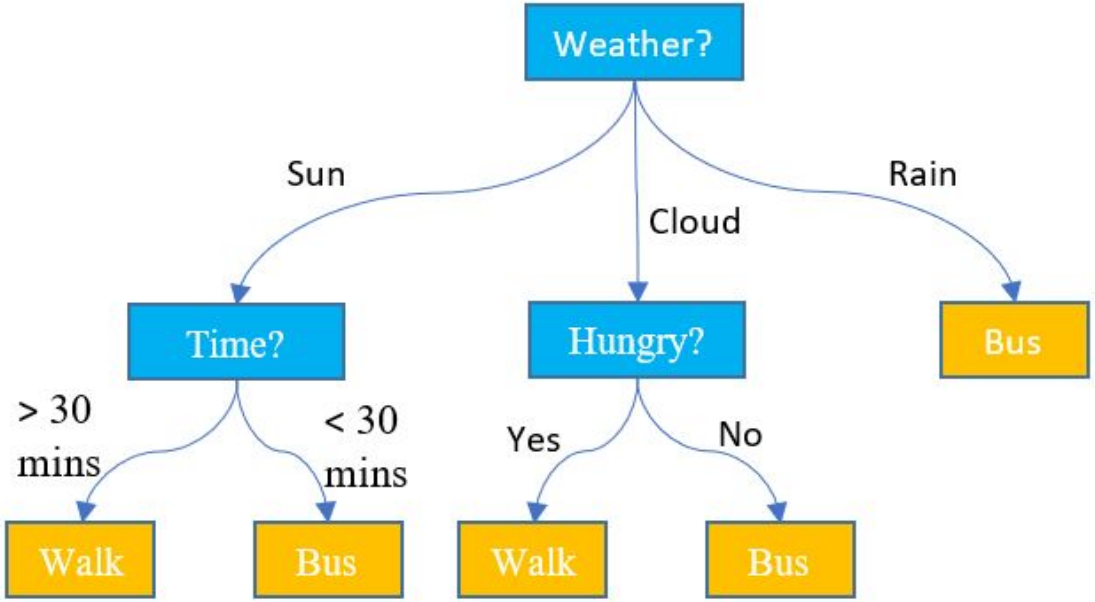
Partition Example



Decision Tree Prediction

1. Start from the root node of the tree
2. Follow the edges according to the features of our sample
3. If we reach the bottom (i.e. come across a leaf node), then we stop
4. Output the prediction at that leaf node

Decision Tree Prediction Example



Training the Decision Trees

Decision Tree Training

- The standard algorithm is the CART algorithm, which boils down a simple recursion:
  1. Find a feature and a threshold of that feature.
  2. Make a split based on the threshold to form two sub-datasets
  3. Recursively apply step 1 and 2 to both of the splitted data sets until a stopping criteria is met

Finding the Right Split

- A reasonable split is found by trying to minimize the impurity of the split
  - We can try a bunch of splits and then pick the split that results in the lowest impurity
- We can think of the impurity as a measure of how mixed the sub-data is from the split.
  - We want the splits to be as un-mixed as possible.

Stopping Criterions

- Maximum depth of tree
- Maximum number of leaf nodes
- Minimum number of samples in each leaf

Pros and Cons of Decision Trees

Benefits of Decision Trees

- Decision trees are universal approximators
- Very interpretable
- Fast to predict with

Downsides of Decision Trees

- Decision trees are sensitive to:
  - Small variations in the training data
  - Operations on the data (such as data rotation)

Questions to Answer

1. What is the approximate depth of a decision tree trained (without restrictions) on a training set with 1 million instances?
2. Why would having no stopping criterion lead to overfitting?
3. What hyperparameters should you change in the decision tree if it is overfitting / underfitting? How should you change those hyperparameters?
4. Can you come up with some examples where transforming the data has a dramatic influence on the resulting trained decision tree?
5. How can we know which features are more important for the decision tree model?
6. What are some potential problems of very large trees?