Decision Trees

February 13, 2020

Introduction

- We will now study a Machine Learning tool : Decision Trees
- They can be used for **classification** and for **regression**.

Problem statement

- We have a dataset of samples, that have several **features**. Each sample also has a class.

Problem statement

- We have a dataset of samples. Each sample also has a class. Each sample also has several **features**.
- For instance we study two types of fishs : the possible classes are **tuna** and **salmon**.
- Each fish has two features : its weight and its length.

Problem statement

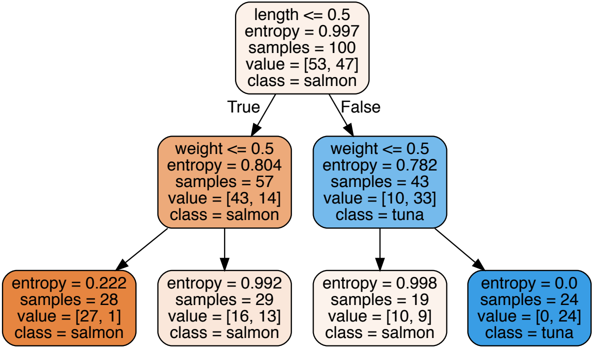
- We have a dataset of samples. Each sample also has a class. Each sample also has several **features**.
- For instance we study two types of fishs : the possible classes are **tuna** and **salmon**.
- Each fish has two features : its weight and its length.
- The question is : are we able to **predict the class by looking at the features ?**

Problem statement

- The **Decision Tree** is a classifier that we will build from the data that will help us to **predict the class of a new datapoint**.

Problem statement

- When the tree is built, it will look like this. Let us analyze what this means :



Building a tree

- We are interested in a method that would automatically build the tree for us.

Building a tree

- We are interested in a method that would automatically build the tree for us.
- Let us try to design such a method.

Segmentation variable

- We start from a simple tree with only one node.

Segmentation

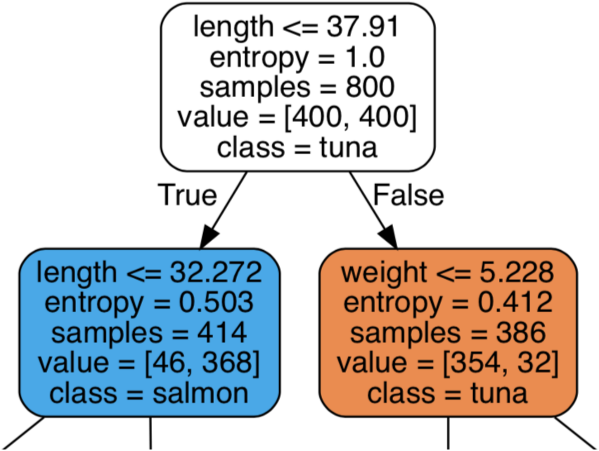
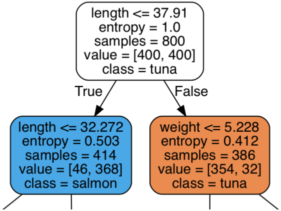


Figure: segmentation

Segmentation variable

- We start from a simple tree with only one node.
- 

- We want to find the feature that helps us to predict the class of the fish with most certainty.

Segmentation variable

- We start from a simple tree with only one node.
- We want to find the feature that helps us to predict the class of the fish with most certainty.
- We then need a measure of the informativeness of the feature on the class.

Segmentation variable

- We start from a simple tree with only one node.
- We want to find the feature that helps us to predict the class of the fish with most certainty.
- We then need a measure of the informativeness of the feature on the class.
- There are several possible measures :
  - Gini factor
  - Information gain
  - Misclassification probability

Notion of Entropy

- Let us discuss the concept of entropy and information.

Notion of Entropy

- Let us discuss the concept of entropy and information.
- Let X be a random variable, that can take the values a_k with probability p_k.

Notion of Entropy

- Let us discuss the concept of entropy and information.
- Let X be a random variable, that can take the values a_k with probability p_k.
- Its entopy is then

$$H = -\sum_{i=1}^{n} p_k \log p_k \tag{1}$$

Entropy

Exercice :
What is the sign of the entropy ?
What are its maximum and minimum values ?

Entropy

- Usually the logarithm in base 2 is used.
- Minimum value : $H = 0$ (deterministic random variable)
- Maximum value : $H = \log n$ (uniform random variable with n values)

Entropy

- Usually the logarithm in base 2 is used.
- Minimum value : $H = 0$ (deterministic random variable)
- Maximum value : $H = \log n$ (uniform random variable with n values)

Remark :the value of the entropy does not depend on the values taken by the random variable, but only on the distribution.

Dataset

- Let us look at our dataset

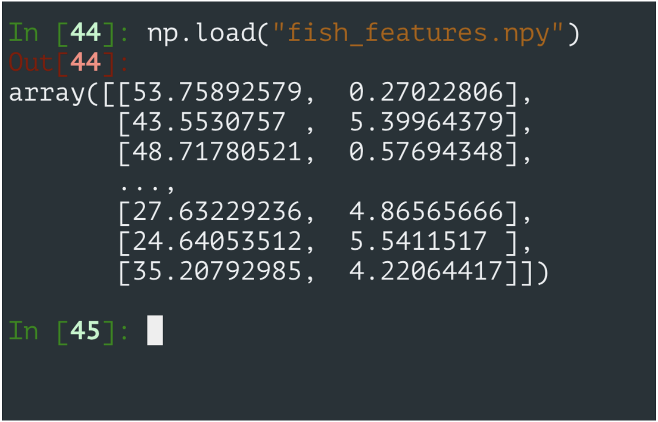
Dataset

- Let us look at our dataset
- We have a database of 800 fishes (tunas and salmon).

Dataset

- Let us look at our dataset
- We have a database of 800 fishes (tunas and salmon).
- the features of the fishes are stored in **numpy arrays**.

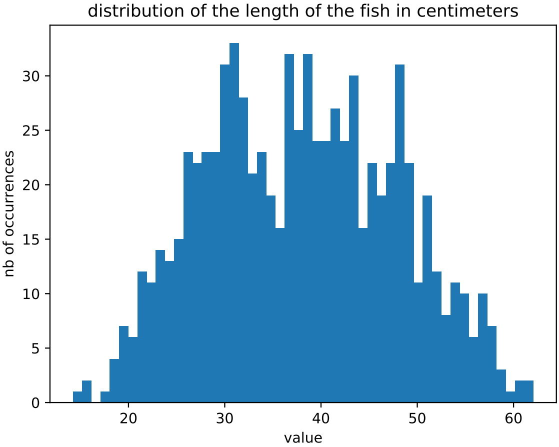
Dataset : ipython demo



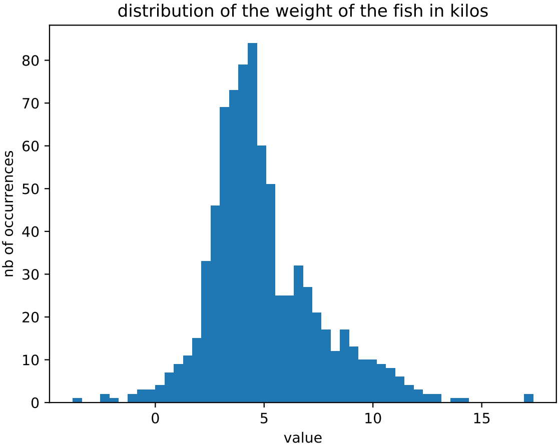
```
In [44]: np.load("fish_features.npy")
Out[44]:
array([[53.75892579,  0.27022806],
       [43.5530757 ,  5.39964379],
       [48.71780521,  0.57694348],
       ...,
       [27.63229236,  4.86565666],
       [24.64053512,  5.5411517 ],
       [35.20792985,  4.22064417]])

In [45]: 
```

Dataset : histograms



Dataset : histograms



Visualization

What other visualization could we make ?

Exercise 1

- Given the number n of datapoints, what is the maximal number of nodes in the tree ?

Exercise 1

- Given the number n of datapoints, what is the maximal number of nodes in the tree ?
- What is then the **prediction cost ?**

Implementation

- We will use **sklearn** to build decision trees.
- `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`
- `https://scikit-learn.org/stable/modules/tree.html`

Implementation

- **pip install sklearn**
- We will also need
  - **numpy**
  - **matploblib**

Exercise 2

- Use the file **fish_tree.py** in order to build your decision tree and plot it.
- Try to use different depths.

Exercise 3

- Uncomment the end of the file **fish_tree.py** in order to predict the class for new fishes.

Exercise 4

- Use the file **fish_blurred_dataset.py** in order to modify the dataset by **adding a new feature to the fishes**.

Exercise 5

- Use the file **fish_tree_pruned.py** in order to build a new decision tree but with a relevant number of nodes.
- You can use the documentation `https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html`
- You can modify :
  - the distributions
  - the value of the parameters **min_samles_split** and **min_impurity_decrease**

Exercise 6

- We can apply what we learned to a famous dataset, the **iris dataset**.
- please use the file **iris.py** in order to build several decision trees with different number of nodes, by changing the specifications given to sklearn.

Discussion

- The design of a decision tree has many variants.

Discussion

- The design of a decision tree has many variants.
- Sometimes the rule used to predict the variable at a leaf node is not the **majority rule**.

Discussion

- The design of a decision tree has many variants.
- When the variable to predict is continuous, we build a **regression** tree.
- Sometimes the rule used to predict the variable at a leaf node is not the **majority rule**.

Overfitting

- Overfitting can easily happen with a decision tree

Overfitting

- Overfitting can easily happen with a decision tree
- To handle it, **pruning** is often performed. It consists in **removing nodes from the tree**.

Overfitting

- Overfitting can easily happen with a decision tree
- To handle it, **pruning** is often performed. It consists in **removing nodes from the tree** :
  - **pre pruning :** while building the tree, we choose not to split some nodes
  - **post pruning :** after building the tree, we remove some nodes.
  - in **Exercise 5** we used prepruning.