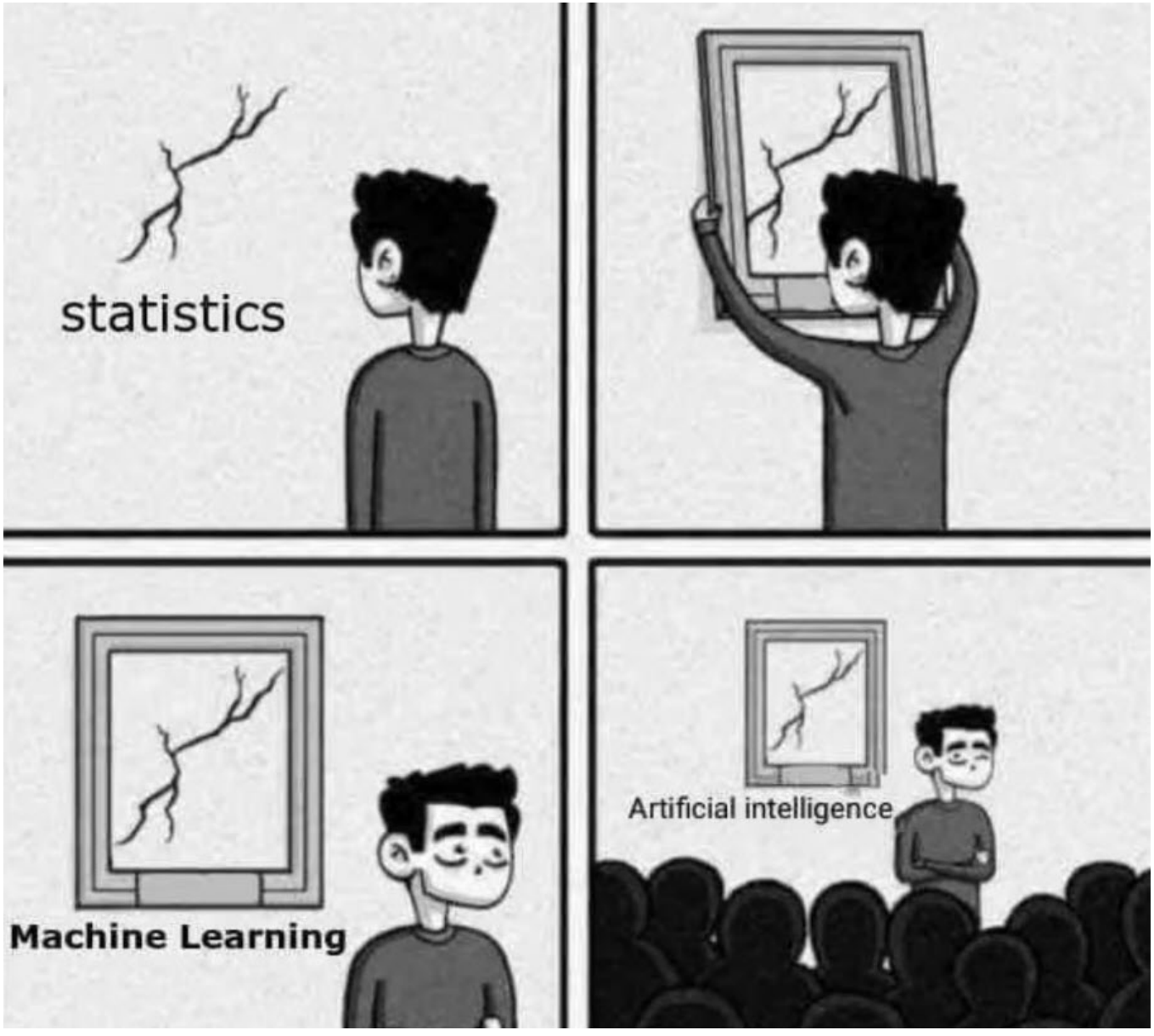
# MOD500 Decision Analysis with Artificial Intelligence Support

Enrico Riccardi[1]

Department of Energy Resources, University of Stavanger (UiS).[1]

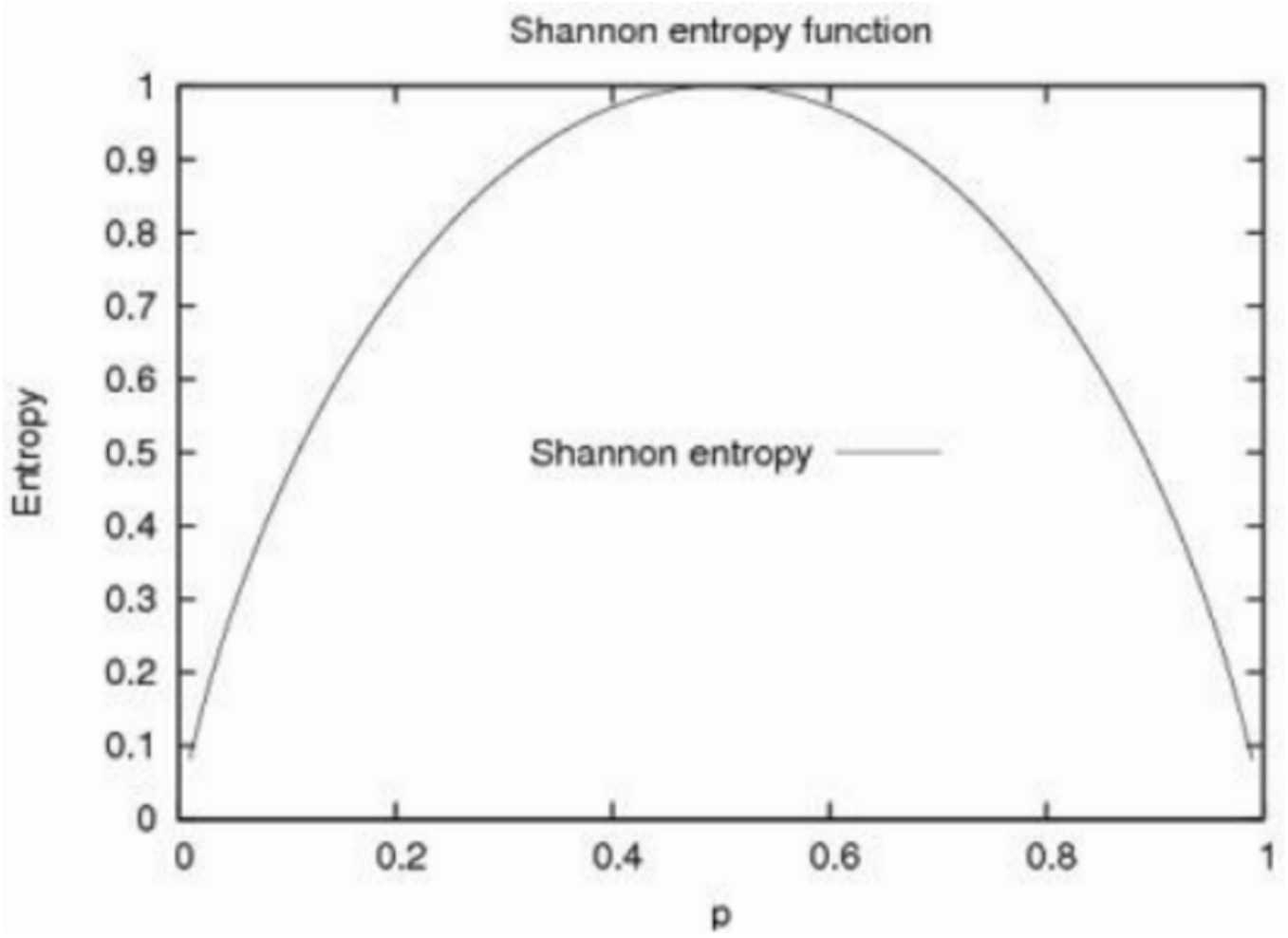
Oct 21, 2024

# Recaps



# Shannon's entropy shape

$$H_X = -\sum_i (p_i) log(p_i)$$



# Task assignment [2]

Code a discrete probability distribution in Python

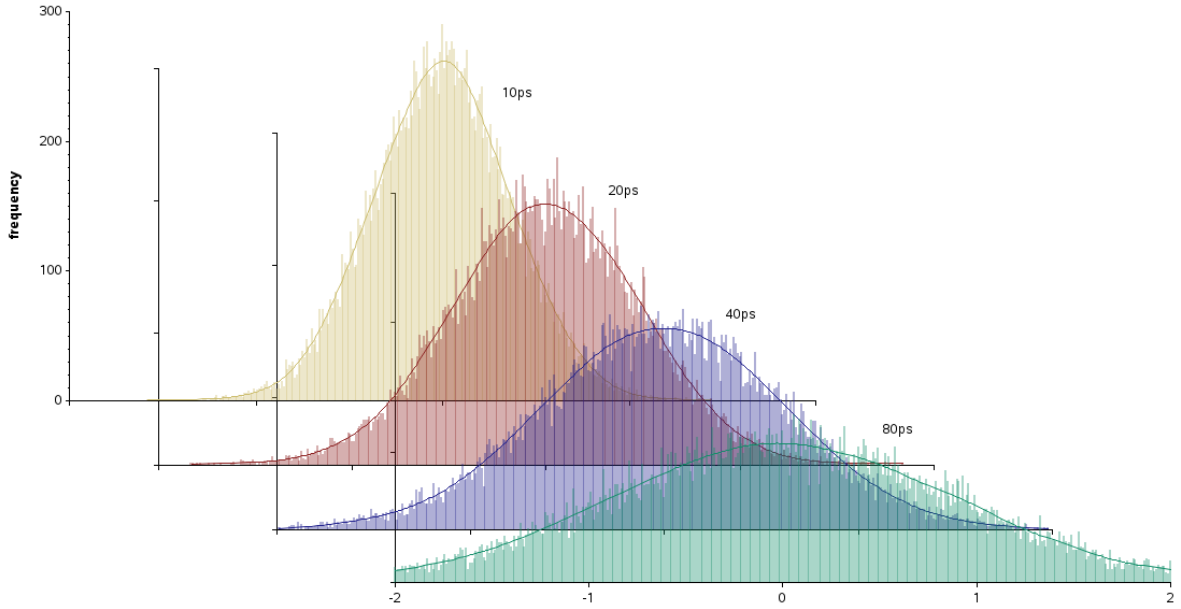
Calculate the Mean and Standard Deviation

How to get an experiment out of this distribution?

Calculate Shannon's Entropy as a function of the number of experiments

# Comparing information

A distribution can be the sum of multiple distributions



# Kullback–Leibler divergence (information gain)

How to compare multiple information sources?

$D_{KL} = \sum_i (p_i) log(\frac{p_i}{q_i})$

- It is a divergence, as it is asymmetric
- It is NOT a metric

# Kullback–Leibler divergence (information gain)

How to compare multiple information sources?

$D_{KL} = \sum_i (p_i) log(\frac{p_i}{q_i})$

- It is a divergence, as it is asymmetric
- It is NOT a metric

# Task assignment [3]

Compare the initial discrete probability distribution with the appriximate ones computed as function of the number of experiments.

Calculate Kullback–Leibler divergence for different number of experiments

Find the distribution that minimise K-L divergence

# Decision trees learning

**Terminology confusion: Decision trees**

2 different domains use the same name for 2 different methods

We here use now the Machine Learning definition

Two types of decision tree (learning)

- Classification trees (our focus here)
- Regression trees

# Decision trees learning

**Terminology confusion: Decision trees**

2 different domains use the same name for 2 different methods

We here use now the Machine Learning definition

Two types of decision tree (learning)

- Classification trees (our focus here)
- Regression trees

# Decision trees learning

**Terminology confusion: Decision trees**

2 different domains use the same name for 2 different methods

We here use now the Machine Learning definition

Two types of decision tree (learning)

- Classification trees (our focus here)
- Regression trees

# Decision trees learning

**Terminology confusion: Decision trees**

2 different domains use the same name for 2 different methods

We here use now the Machine Learning definition

Two types of decision tree (learning)

- Classification trees (our focus here)
- Regression trees

# Decision trees learning

**Terminology confusion: Decision trees**

2 different domains use the same name for 2 different methods

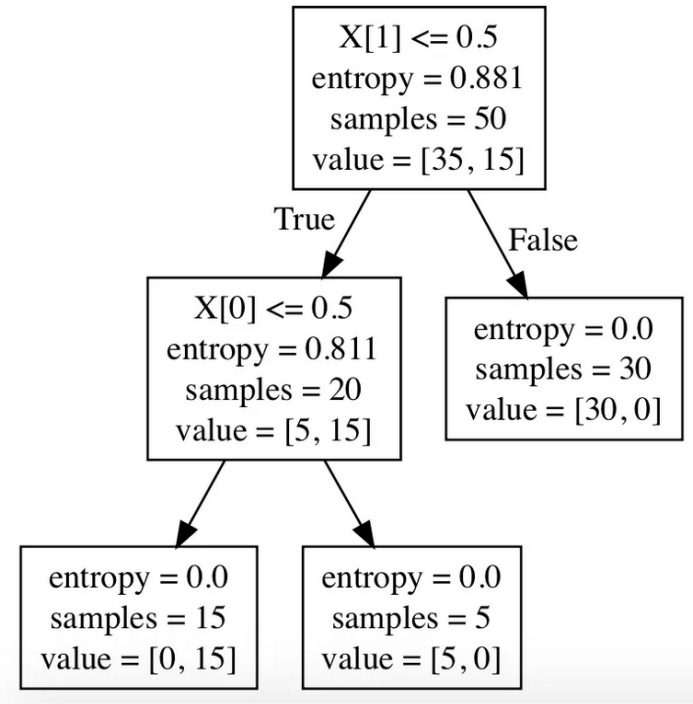
We here use now the Machine Learning definition

Two types of decision tree (learning)

- Classification trees (our focus here)
- Regression trees

# Decision trees learning

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features

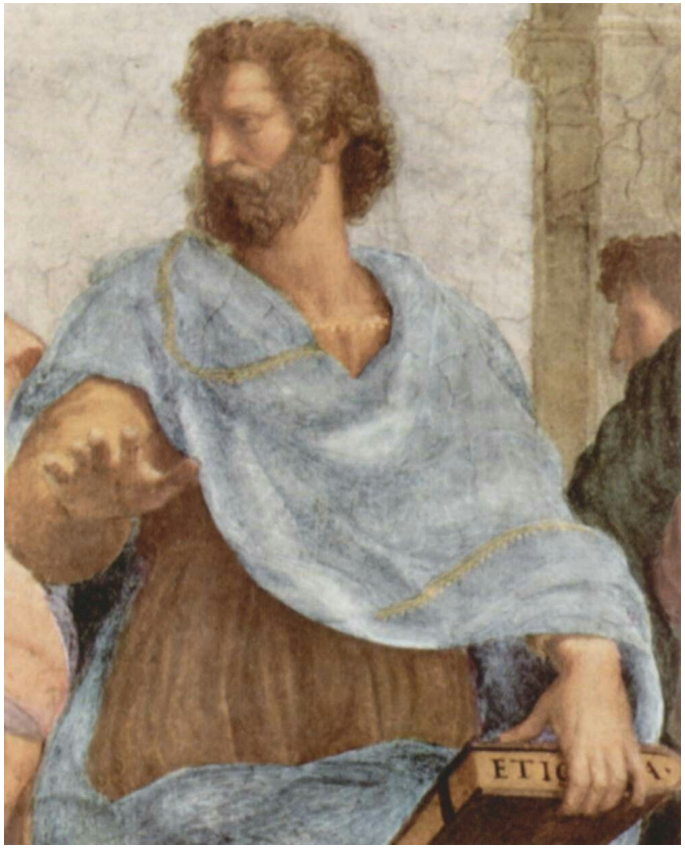


# Decision trees

New? Well, the name came from...

William Belson, in 1959

BUT the basic idea is even older (Aristotle's book on Categories)

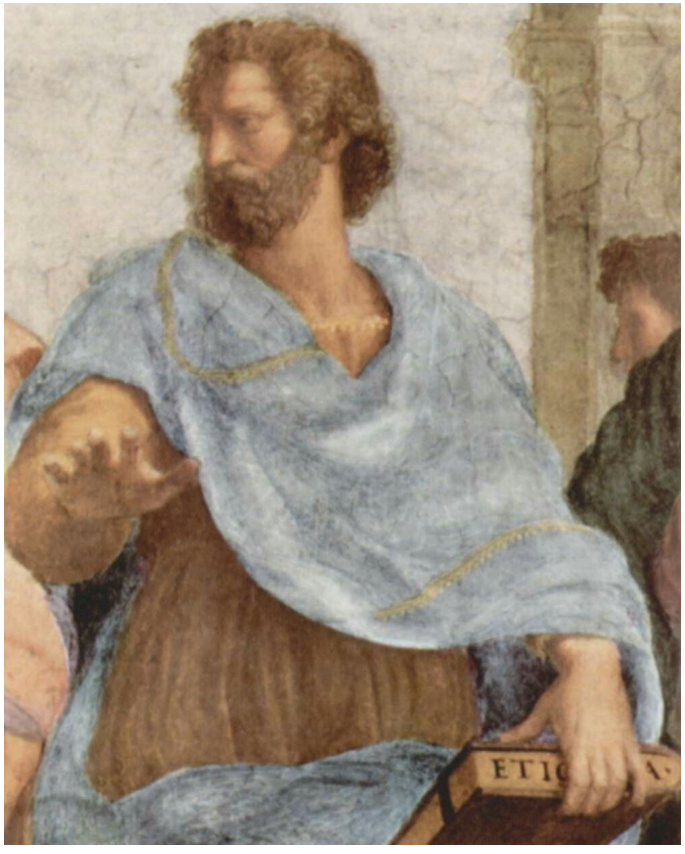


# Decision trees

New? Well, the name came from...

William Belson, in 1959

BUT the basic idea is even older (Aristotle's book on Categories)

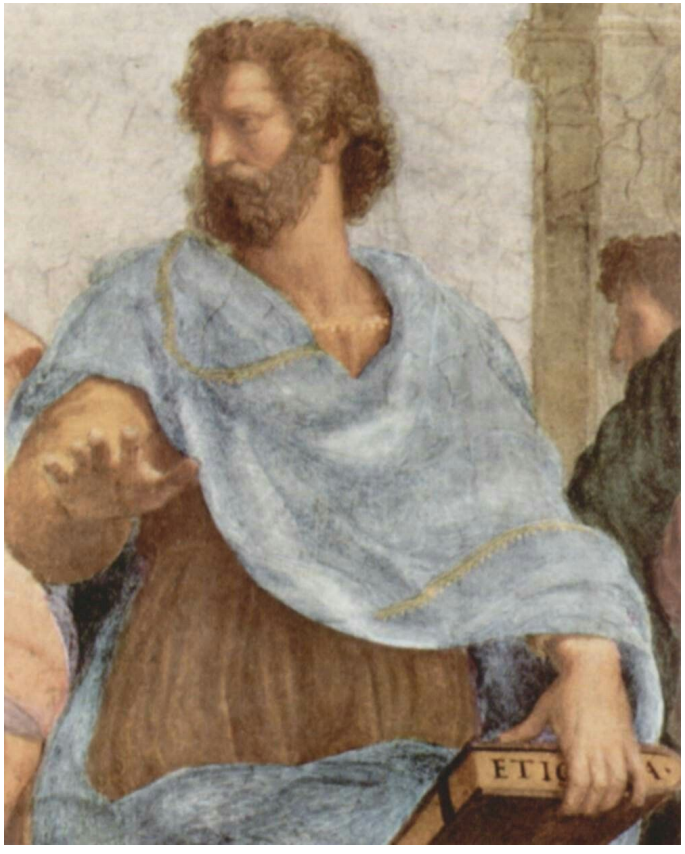


# Decision trees

New? Well, the name came from...

William Belson, in 1959

BUT the basic idea is even older (Aristotle's book on Categories)



# Decision trees

It is a simple model for supervised classification

Each decision nodes performes a Boolean test (binary split version)

They are build out of DATA!

At each split, we perform the slip that reduce entropy the most.

**REMINDER**

We need to provide a label!

# Decision trees

It is a simple model for supervised classification

Each decision nodes performes a Boolean test (binary split version)

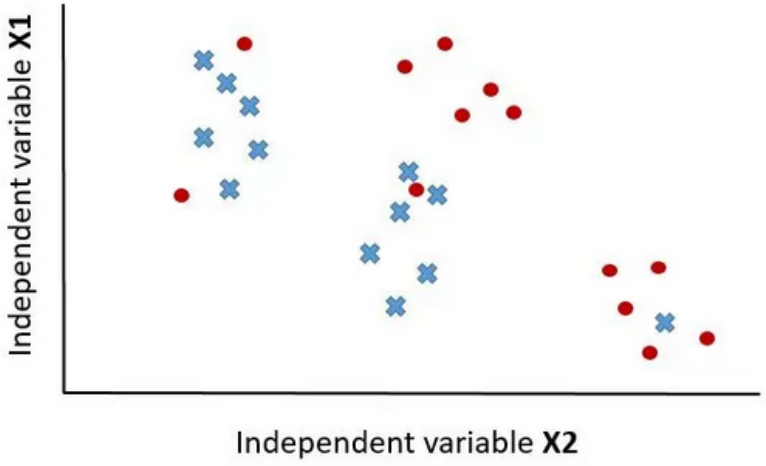
They are build out of DATA!

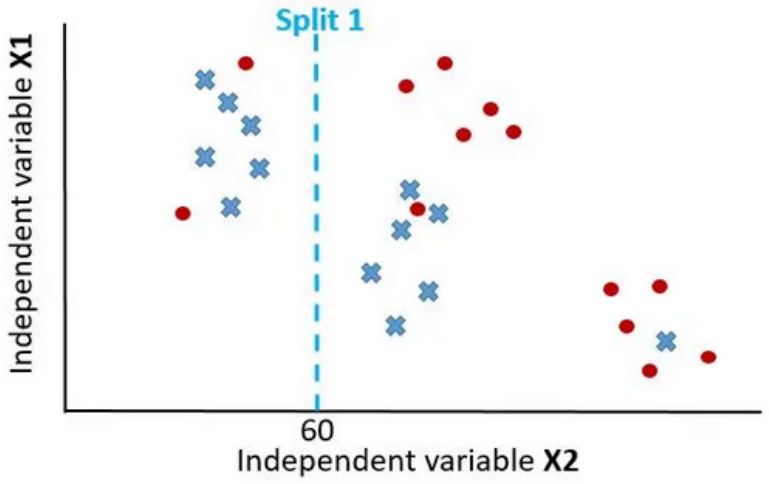
At each split, we perform the slip that reduce entropy the most.

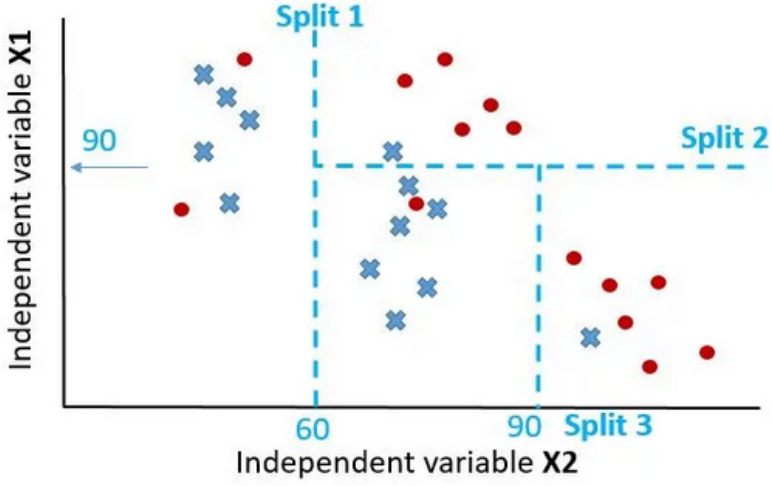
**REMINDER**

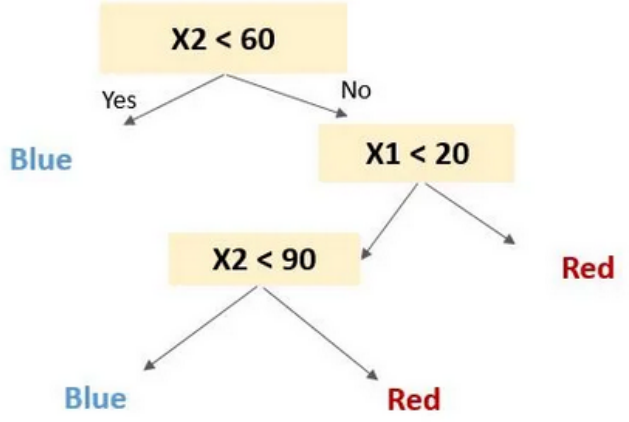
We need to provide a label!

# Decision tree outcome



# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees

## Pseudo-code

- Compute the entropy of each feature (myopic approach)
- Pick the feature with the maximum entropy
- For each value of the selected feature, compute the entropy of the new population
- Compute the Information Gain by splitting the dataset
- Repeat for the number of desired splits

# Decision trees in Python

```
"""
MOD500 tutorial: Decision tree minimal example

"""
import numpy as np
from matplotlib import pyplot as plt

from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier, plot_tree

iris = load_iris()
X = iris.data
y = iris.target

clf = DecisionTreeClassifier(max_leaf_nodes=10,
                             criterion='entropy')
clf.fit(X, y)

plot_tree(clf, proportion=True, filled=True)

plt.show()
```

# Tutorial [4]

Generate (at least) 4 different probability distributions

Make a meaningful label, and then make a decision tree from the data generated

(Use the given template to sort out Python programming part if you need)