CLASSIFICATION I
NEAREST NEIGHBOURS

Dr. Lablanche Pierre-Yves
African Institute for Mathematical Sciences

WHAT IS THIS FLOWER ?

WHAT IS THIS FLOWER ?

Setosa ?

Versicolor ?

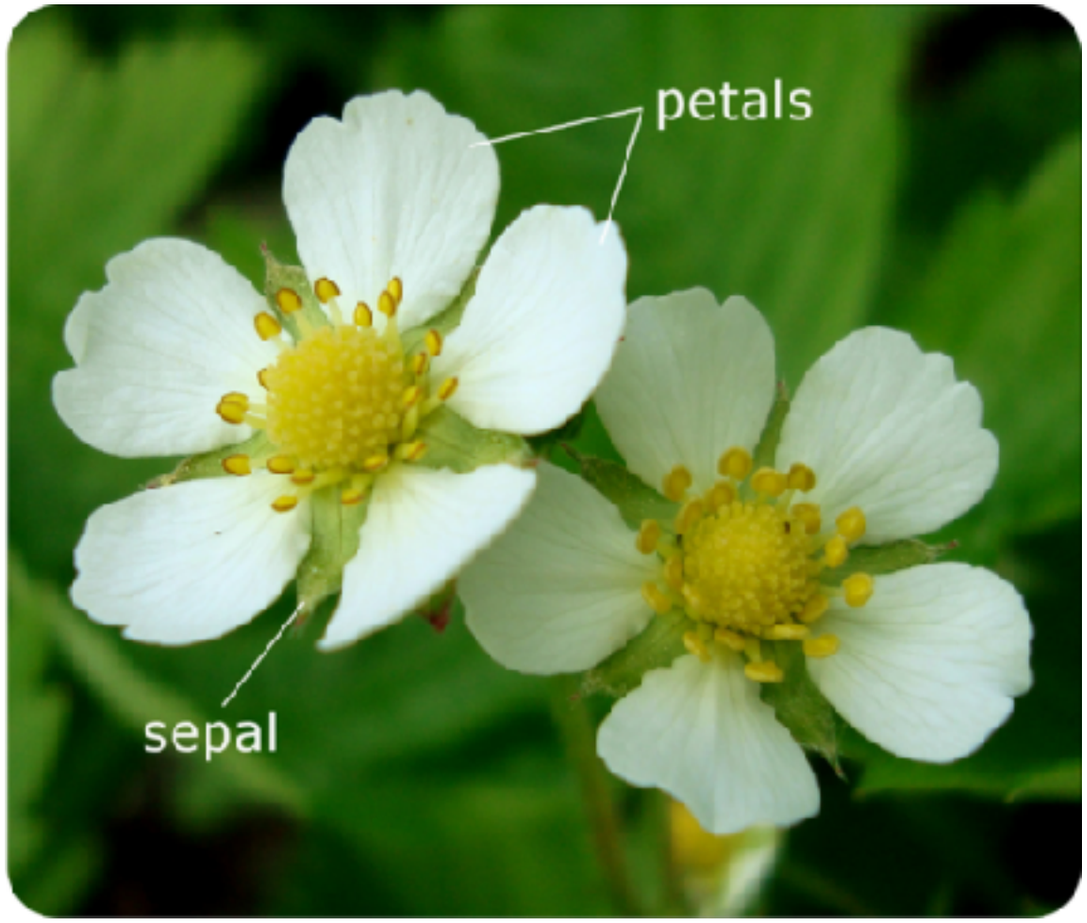
Virginia ?

WHAT WOULD YOU DO TO FIND THE NAME OF THE FLOWER ?

IRIS FEATURES

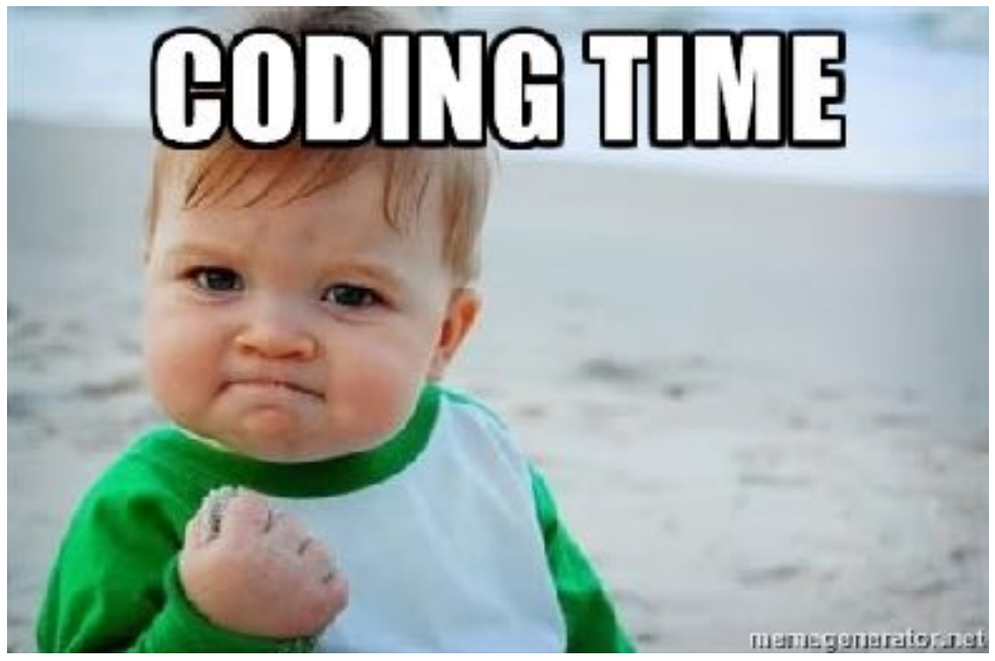
Setosa, Versicolor and Virginica can be differentiated based on four **features** :

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width



CODING TIME (1)

- Open python/ipython or an empty jupyter notebook
- Import the *iris* dataset from scikit
- Explore and Visualise the data set
- Try to find the best feature(s) to classify flowers



IRIS FEATURES

Setosa, Versicolor and Virginica can be differentiated based on four **features** :

- Sepal Length
- Sepal Width
- Petal Length
- Petal Width

We can investigate this 4 dimensions feature space and assign the label of the closest flower to our unknown flower…

… or vote over the K closest flowers

K-Nearest Neighbours

K-NEAREST NEIGHBOURS

How it works :

1. Find the K nearest neighbours of a new object
2. Vote (Classification) or Average (Regression) over the K nearest neighbours and assign the result to the new object.

K-NEAREST NEIGHBOURS

- Several choices of searching algorithm…
  …very often **k-d tree** or **ball tree**

- Metric is important!

Most common Minkowski degree p=2 (aka Euclidean distance):

$$d(x, y) = d(y, x) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$

More details in lecture 3

K-NEAREST NEIGHBOURS

- Pros

  + Powerful and fast for classification and regression

  + Simple to implement and use

- Cons

  - Does not like high dimension space
  (curse of dimensionality)

  - Chose your metric accordingly to the data set

CODING TIME (2)

- Import the **`KNeighborsClassifier`** from sklearn
- Train it (**`.fit()`** method) on the iris data
- Use **`predict`** and **`predict_proba`** on new values
- Repeat with different parameters value