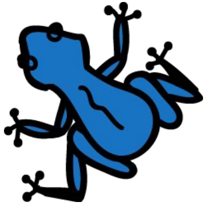
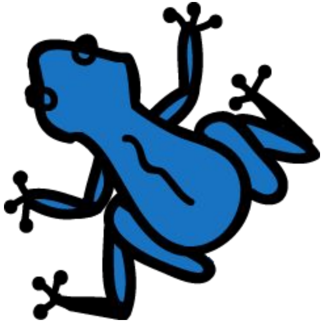
# Classification of frog calls

Can we infer the specie of a frog just from its call ?





# Summary

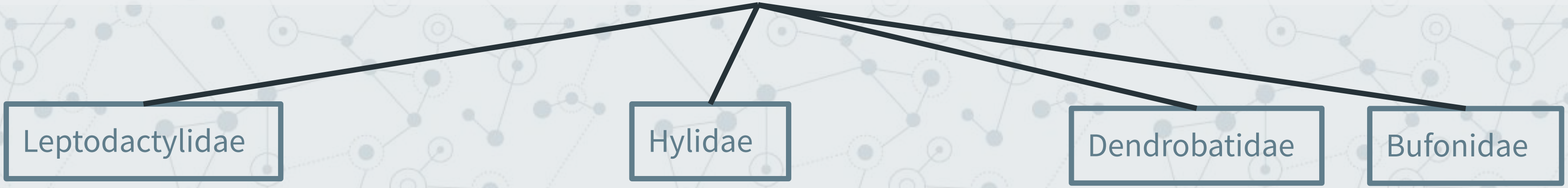


# Summary



# The Frogs

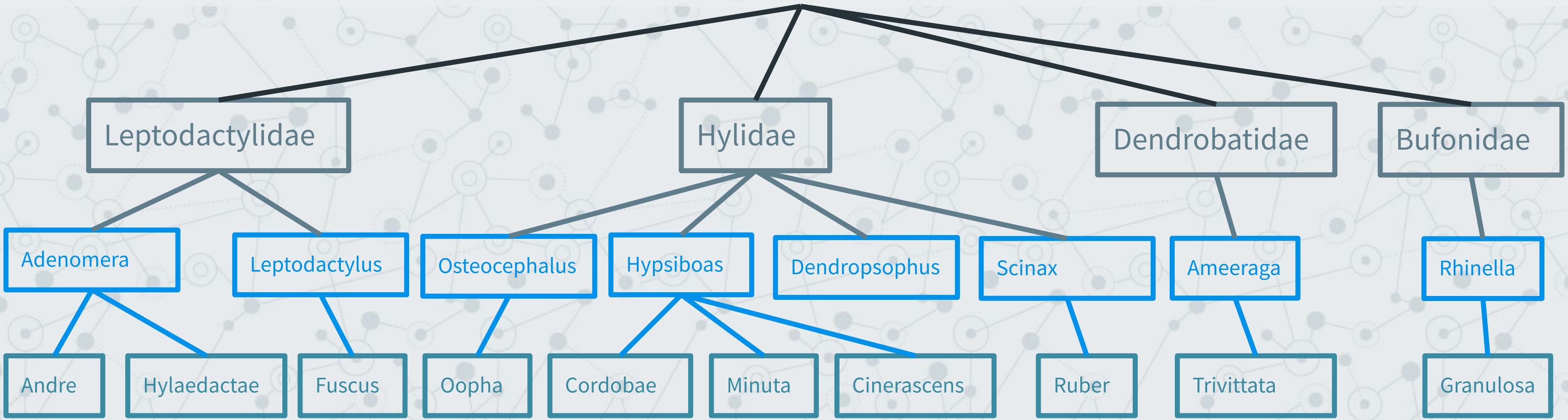
# The Frogs



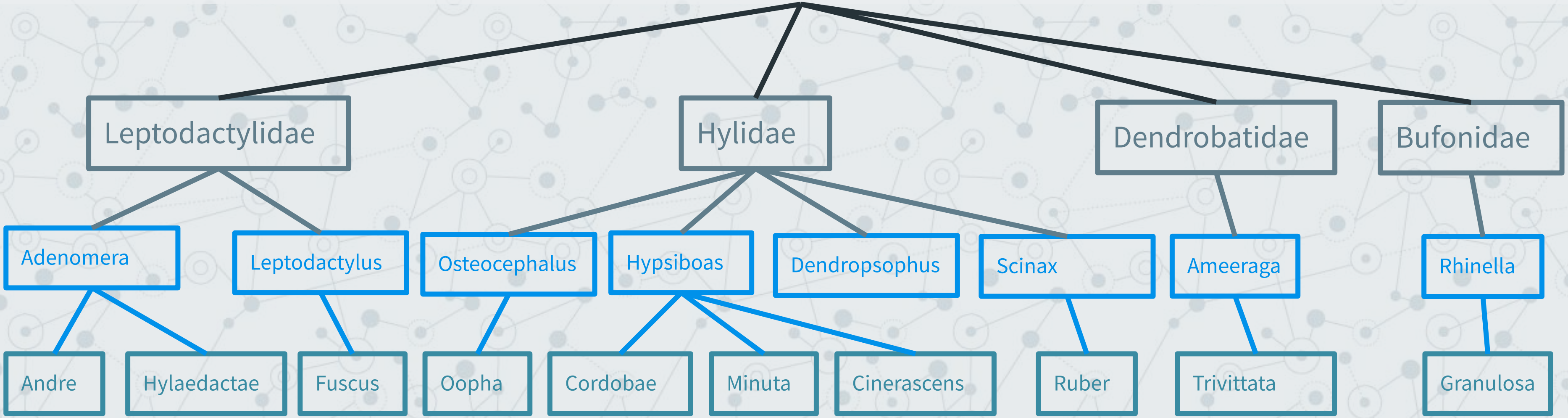
# The Frogs

- Leptodactylidae
  - Adenomera
  - Leptodactylus
- Hylidae
  - Osteocephalus
  - Hypsiboas
  - Dendropsophus
  - Scinax
- Dendrobatidae
  - Ameeraga
- Bufonidae
  - Rhinella

# The Frogs


Leptodactylidae
Hylidae
Dendrobatidae
Bufonidae
Adenomera
Leptodactylus
Osteocephalus
Hypsiboas
Dendropsophus
Scinax
Ameeraga
Rhinella
Andre
Hylaedactae
Fuscus
Oopha
Cordobae
Minuta
Cinerascens
Ruber
Trivittata
Granulosa

# The Frogs





**Look at these cute froggy bois !**

# The Frogs

- **For each specie, 6 instances** of **one** syllable of their call
- The audio of the syllable is transformed into **MFCCs**

Andre | Hylaedactae | Fuscus | Oopha | Cordobae | Minuta | Cinerascens | Ruber | Trivittata | Granulosa




**Look at these cute froggy bois !**

## What are MFCCs ?

- MFCCs stands for Mel-Frequency Cepstral Coefficients
- They are commonly used in sound classification

**What does this mean ?**
**How much information is retained from the original audio ?**

## How are MFCCs obtained ?

1. Decompose the signal into components with Fourier Transform
2. Project the powers of the  components onto the mel scale
3. Take the logs of the powers at each of the mel frequencies.
4. Take the discrete cosine transform of the list of mel log powers, as if it were a signal.
5. The MFCCs are the amplitudes of the resulting spectrum.

# What are MFCCs ?

- MFCCs are very common because they represent **tone**
- This aspect of sound is very useful to interpret speech

# How much of the information is retained ?

We converted the MFCCs into an **audio signal**

Each sound represents a **syllab**, a component of a frog call:

Adenomera Hylaedactylus

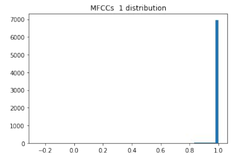
AdenomeraAndre

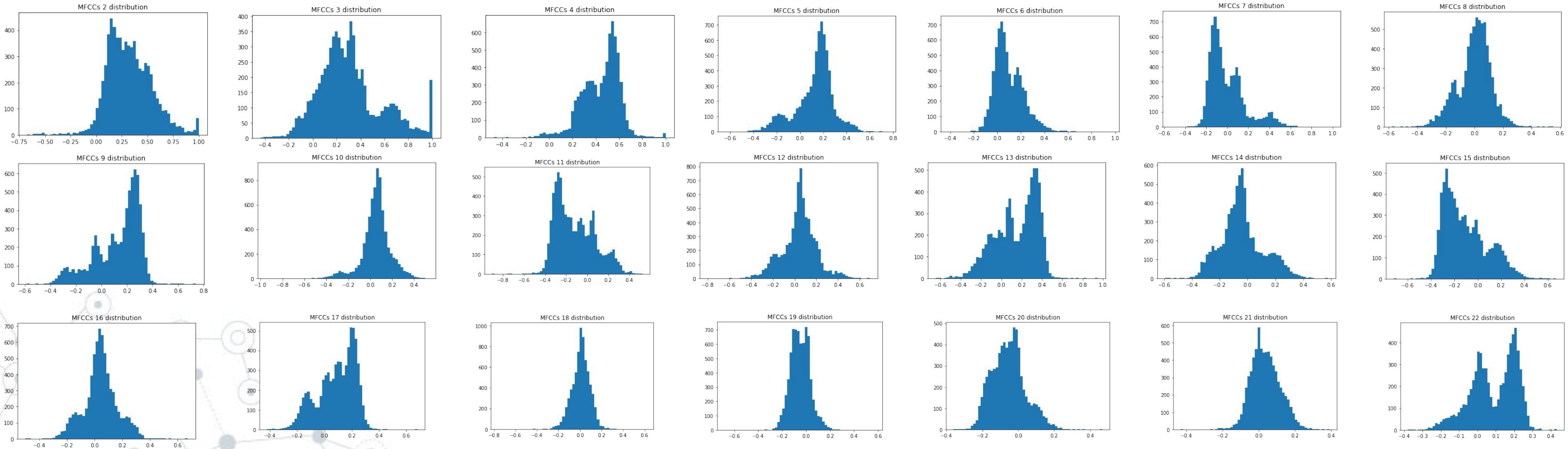
HypsiboasCordobae

**A lot of characteristics of the signal are lost in the process. Do we have enough  to classify species ?**

# How much of the information is retained ?

- The MFCC 1 has been removed because it was too homogeneous.



# Summary

## 1. Presentation of the dataset

- 1.1 Overview
- 1.2. What are MFCCs ?
- 1.3. Characteristics of the dataset
- 1.4. Normalisation

## 2. Classification

- 2.1. The state-of-the-art algorithms
- 2.2. How do they work ?
- 2.3. Using Linnaeus taxonomy

# Three state-of-the-art algorithms

## Random Forest

**Advantages**

- Really fast to set up
- Versatile
- Rarely over-fitting.
- RF is very effective performance wise.

**Fast, simple and flexible tool to develop**

# Three state-of-the-art algorithms

## Random Forest

**Advantages**

- Really fast to set up
- Versatile
- Rarely over-fitting.
- RF is very effective performance wise.

**Fast, simple and flexible tool to develop**

## Support Vector Machine

**Advantages**

- Often used with MFCCs
- Rarely over-fitting
- More effective in high dimensional spaces.

**Drawbacks**

- Not suitable for large datasets

**Particularly suited for sound applications**

# Three state-of-the-art algorithms

## Random Forest

**Advantages**

- Really fast to set up
- Versatile
- Rarely over-fitting.
- RF is very effective performance wise.

**Fast, simple and flexible tool to develop**

## Support Vector Machine

**Advantages**

- Often used with MFCCs
- Rarely over-fitting
- More effective in high dimensional spaces.

**Drawbacks**

- Not suitable for large datasets

**Particularly suited for sound applications**

## K-NN

**Advantages**

- It has no assumptions to be used.
- Very easy to implement for multi-class problem.

**Drawbacks**

- Not suitable in high dimensional spaces.

**It has no training period**

# Random Forest

- **Step 1**: Create a bootstrap dataset:

N experiments

| j | 1 | 2 | j-1 | j+1 | p |
|---|---|---|---|---|---|
| $x_0^j$ | $x_0^1, \ldots, x_0^{j-1}, x_0^{j+1}, \ldots, x_0^p$ | | | | |
| $x_1^j$ | $x_1^1, \ldots, x_1^{j-1}, x_1^{j+1}, \ldots, x_1^p$ | | | | |
| $\vdots$ | $\vdots$ | | | | |
| $x_{N-1}^j$ | $x_{N-1}^1, \ldots, x_{N-1}^{j-1}, x_{N-1}^{j+1}, \ldots, x_{N-1}^p$ | | | | |
| $x_N^j$ | $x_N^1, \ldots, x_N^{j-1}, x_N^{j+1}, \ldots, x_N^p$ | | | | |

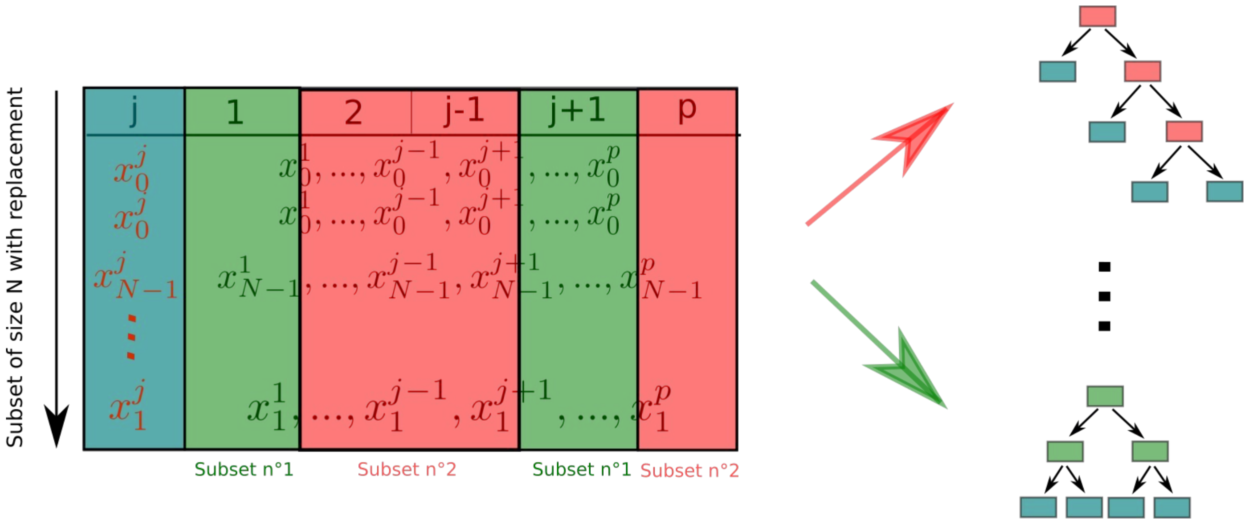
$\longrightarrow$

Subset of size N with replacement

| j | 1 | 2 | j-1 | j+1 | p |
|---|---|---|---|---|---|
| $x_0^j$ | $x_0^1, \ldots, x_0^{j-1}, x_0^{j+1}, \ldots, x_0^p$ | | | | |
| $x_0^j$ | $x_0^1, \ldots, x_0^{j-1}, x_0^{j+1}, \ldots, x_0^p$ | | | | |
| $x_{N-1}^j$ | $x_{N-1}^1, \ldots, x_{N-1}^{j-1}, x_{N-1}^{j+1}, \ldots, x_{N-1}^p$ | | | | |
| $\vdots$ | | | | | |
| $x_1^j$ | $x_1^1, \ldots, x_1^{j-1}, x_1^{j+1}, \ldots, x_1^p$ | | | | |

# Random Forest

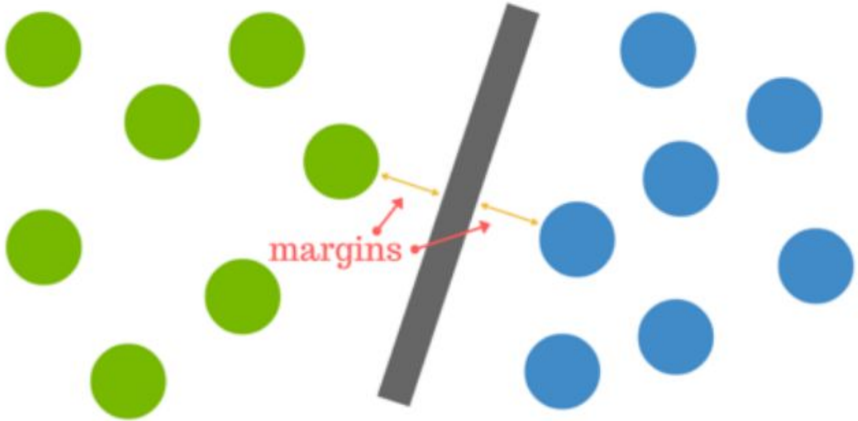
- **Step 2**: Create decision trees from subsets of the dimensions of the dataset




- **Step 3**: Given a new sample, we take the label predicted by the **majority** of the trees

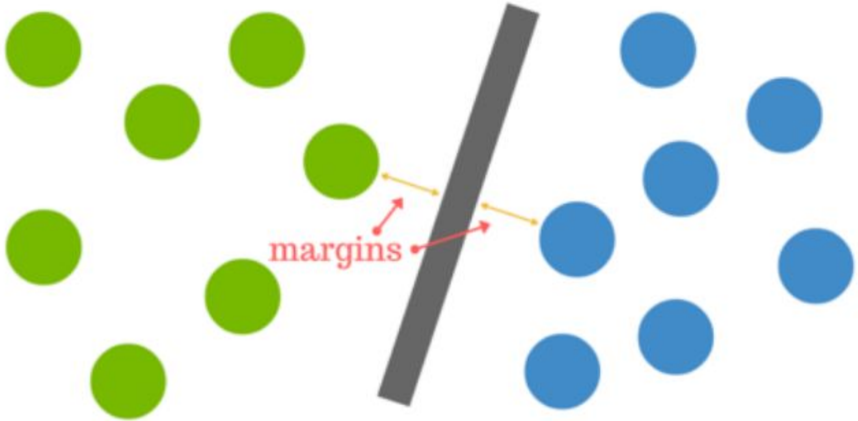
# Support Vector Machine

- Goal: optimise the hyperplane that separate our datas by optimizing the margin between the individuals and the hyperplane.



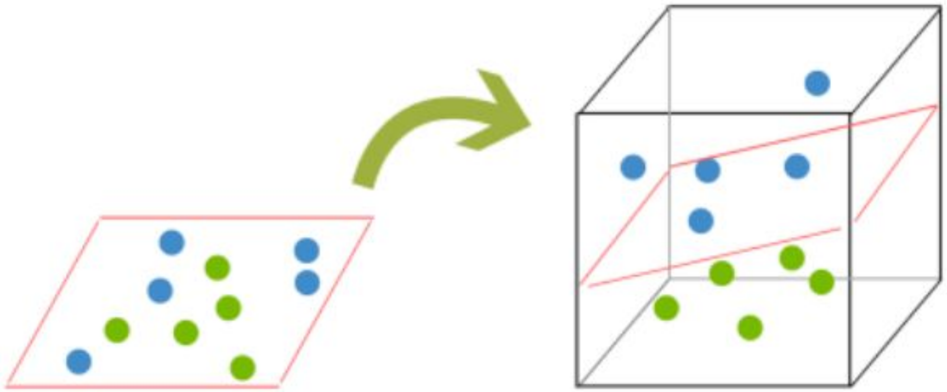
# Support Vector Machine

- If datas are nearly linearly separable, the algorithm will make a trade-off between minimization of the margin and minimization of the classification errors.

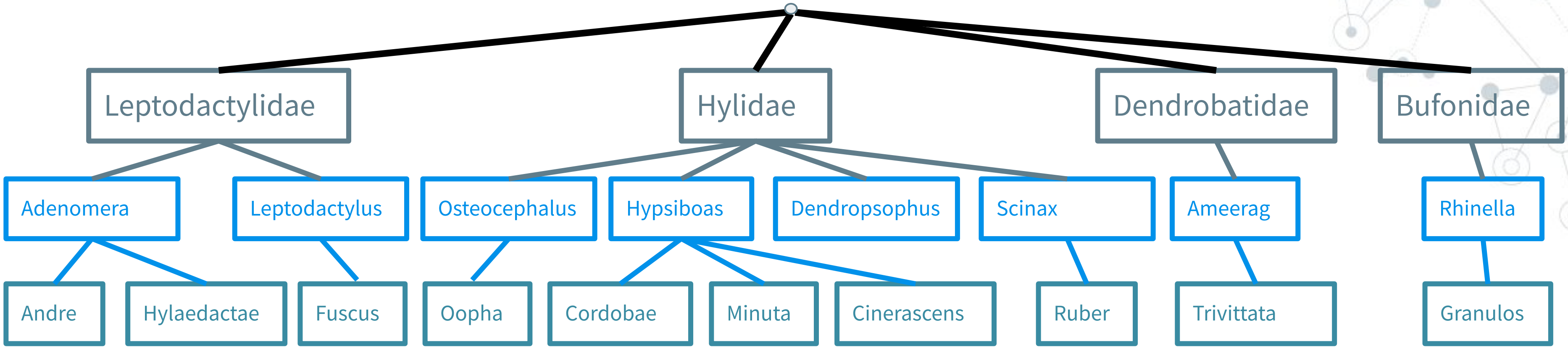


# Support Vector Machine

- If datas are far from linearly separable, it will redefine the space of description in a space with more dimensions.

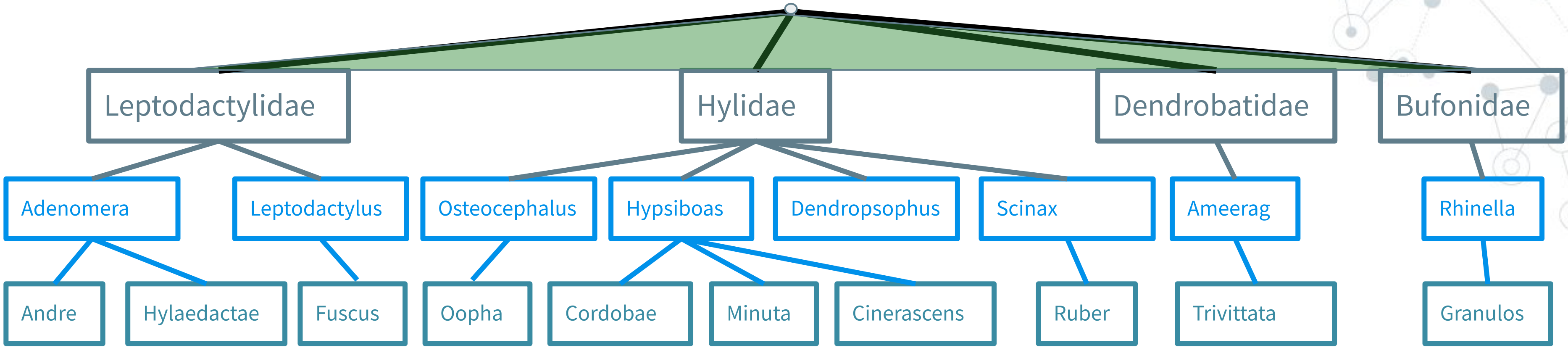


# Improvements using Linnaeus taxonomy




## How does it work ?

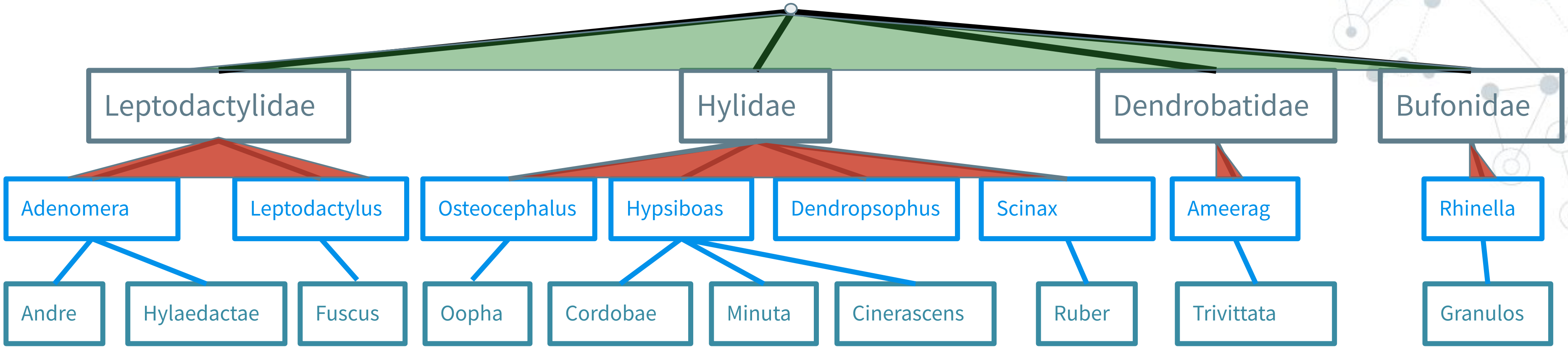
# Improvements using Linnaeus taxonomy




## How does it work ?

A first classifier trained on families will tell the family of the individual.

# Improvements using Linnaeus taxonomy

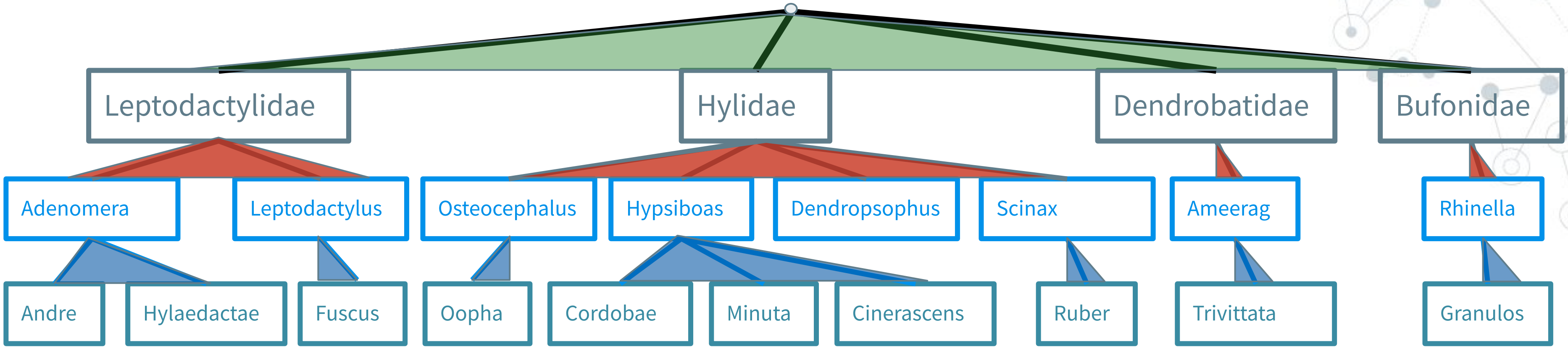



## How does it work ?

A first classifier trained on families will tell the family of the individual.

A second classifier is used, trained only on this family, predicting the genus.

# Improvements using Linnaeus taxonomy




## How does it work ?

A first classifier trained on families will tell the family of the individual.

A second classifier is used, trained only on this family, predicting the genus.

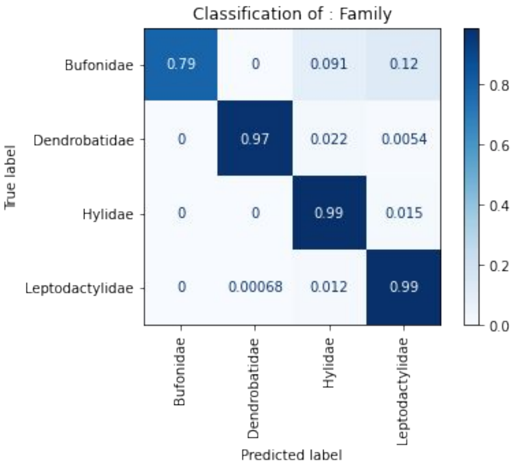
A third classifier is used, trained only on this genus, predicting the specie.

# Summary

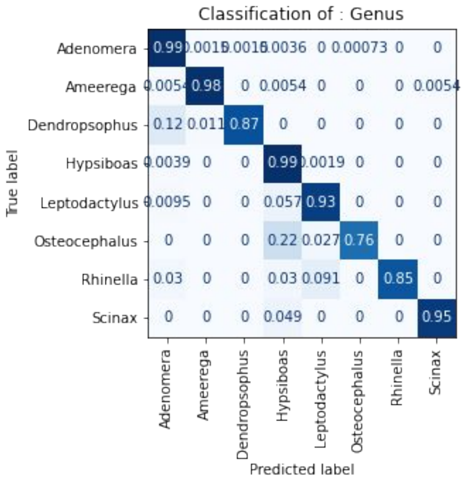


# Random Forest

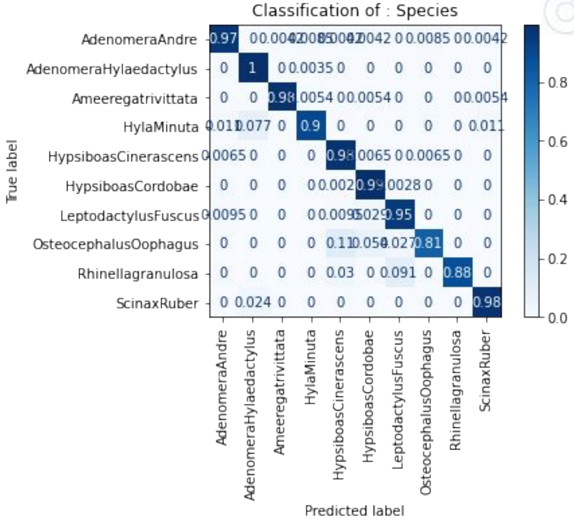
Regardless of the level of classification, RF classifier has pretty good results, with some issues.




Confusion Matrix at the family level



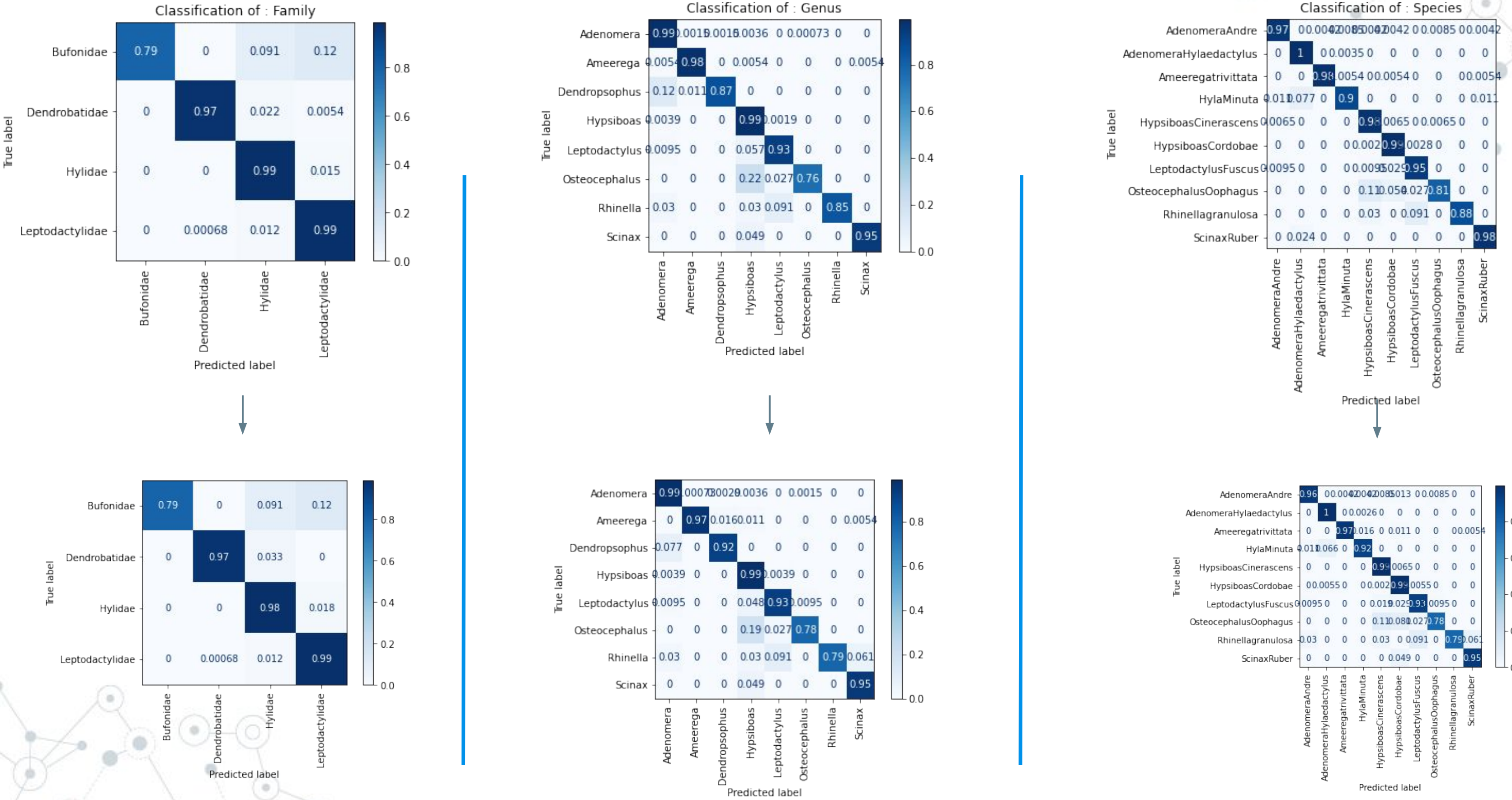

Confusion Matrix at the genera level




Confusion Matrix at the specie level

**Accuracy : 0.953**

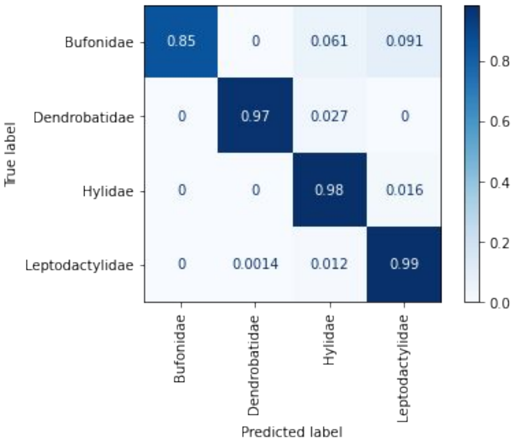
# Hierarchical Random Forest



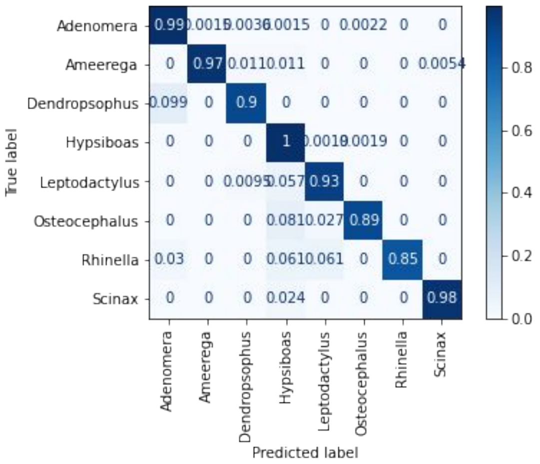

**Accuracy : 0.974**

# SVM

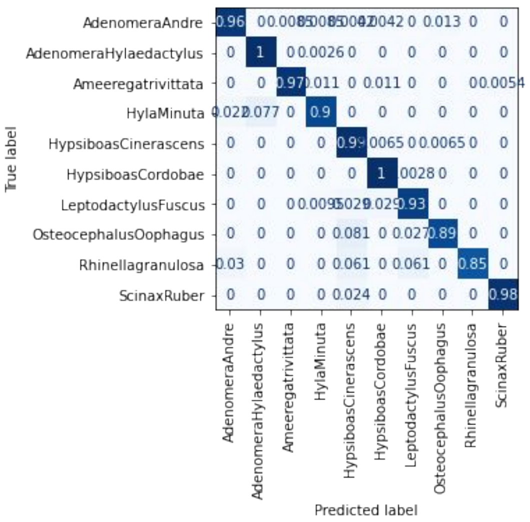
The majority of the individuals are well segregated. Most of the predictions match with the true label of the individual.




Confusion Matrix at the family level




Confusion Matrix at the genera level

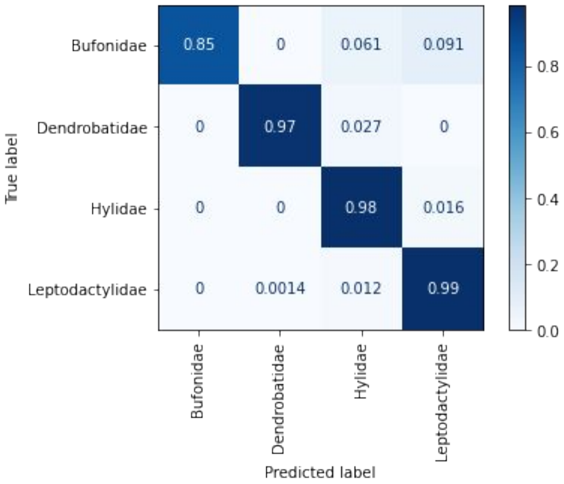



Confusion Matrix at the specie level

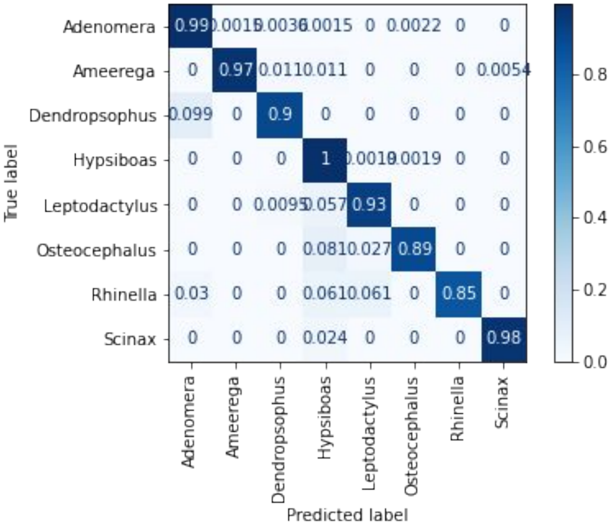
**Accuracy : 0.978**

# Hierarchical SVM

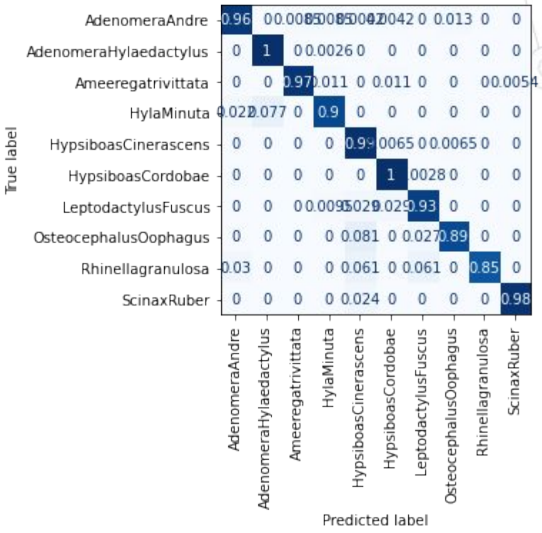
It is still a really good classifier, but the overall accuracy is only improved by 0.2%.




Confusion Matrix at the family level




Confusion Matrix at the genera level




Confusion Matrix at the specie level

**Accuracy : 0.981**

# Conclusion

# Hierarchical SVM

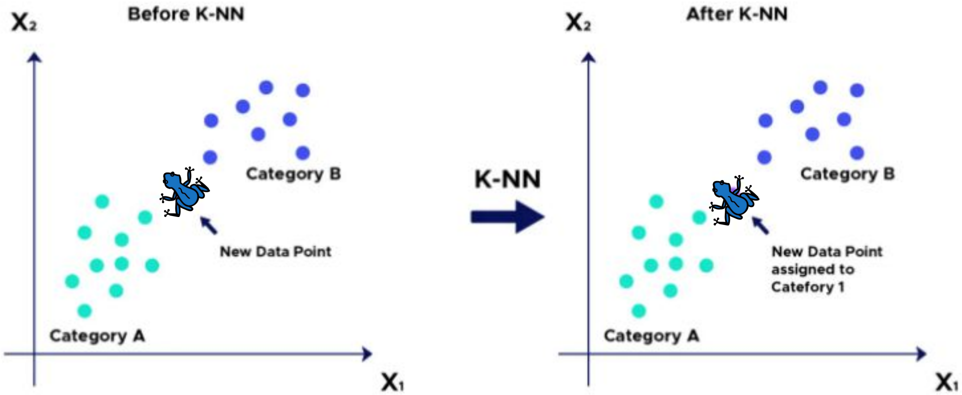
- A big part of the **information** is **lost** during **MFCC computation**, which makes classification **harder**.
- Using MFCCs, **SVM** appears to be a **better** classifier than **RF**.
- Leveraging the **topology** of **Linnaeus taxonomy** seems to **increase** the viability of a sub optimal classifier.
- There is **almost no improvement** when the classifier does well already

# Thank you, froggy friend

## Any questions?

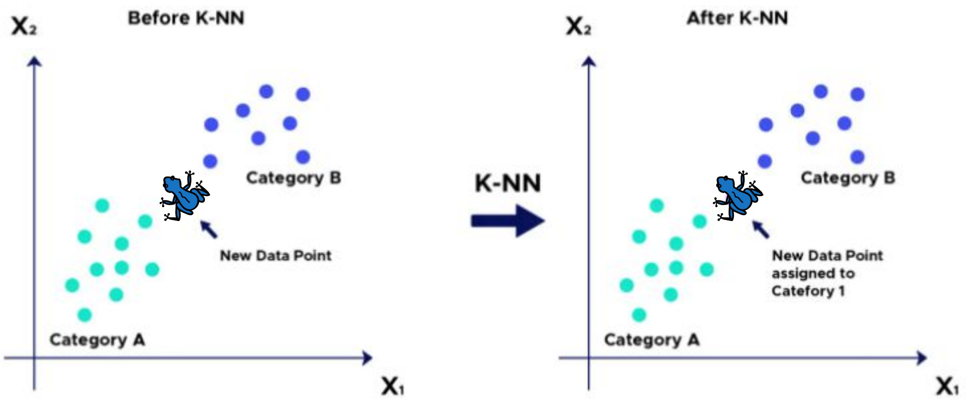
# K-Nearest-Neighbor

- Goal : Infer the class of a new individual based on the K nearest neighbor (the name is pretty descriptive)
- The training dataset is placed in a space.

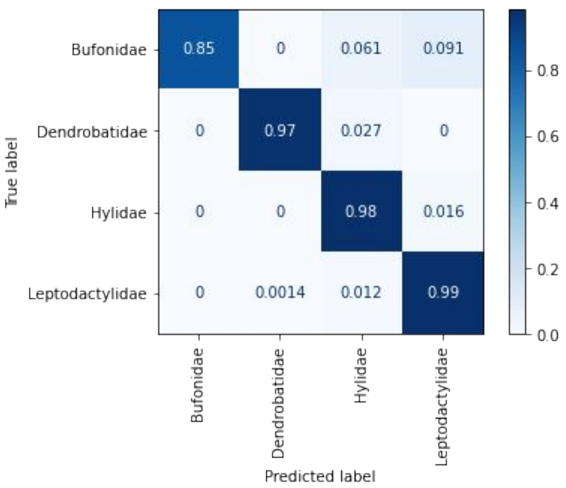


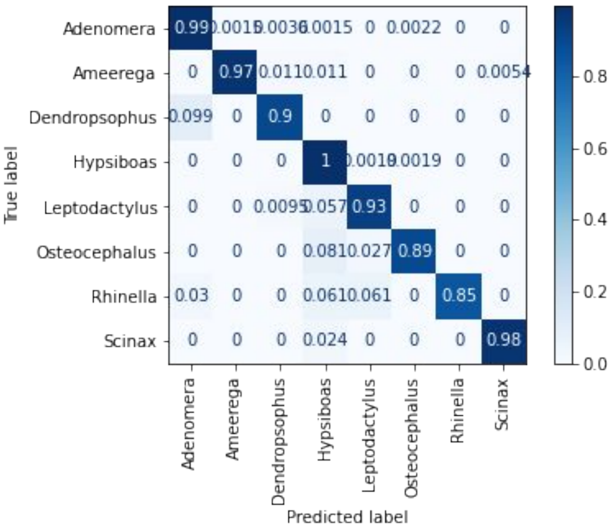
# K-Nearest-Neighbor

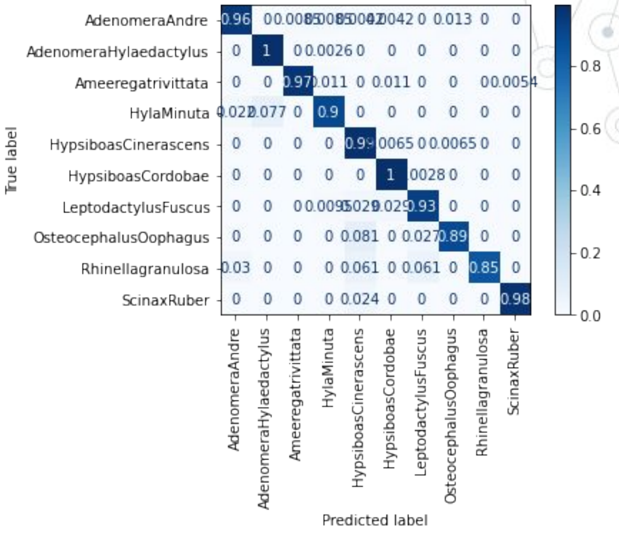
- The test individual is placed in this space and the K elements that are the nearest from it are selected.
- Because we are in a case of classification, our test individual's class will be the most frequent class among the K elements.



# KNN




Confusion Matrix at the family level

Confusion Matrix at the genera level

Confusion Matrix at the specie level