This document consists of a synthesis of the different obstacles and solution founded I faced during my project:

Firstly, I never did audio analysis previously, so I had to find library adapted to extract information from audio files.

To do this I tried PyAudio <http://people.csail.mit.edu/hubert/pyaudio/> who allowed me to listen to the recording but not to make advanced processing.

The library librosa <https://librosa.org/doc/latest/index.html> was more adapted with many audio features as Fourier transformation included.

The second thing was to understand what an audio file is, what is the sample rate and how to find the frequencies by doing a Fourier transformation. This part is included inside “Work with a song” in the notebook.

The training data consists of short extract (around one second) where we know if it is possible or not to ear one specific species. My first idea is to build a model by species as there is no link between them and the presence of one species shouldn’t affect the presence of others.

For the first model I transformed the audio using mfcc ( <https://towardsdatascience.com/extract-features-of-music-75a3f9bc265d> ) and then trained a random forest classifier which gives me interesting results for different species. So I was able to train one classifier by species to detect if a short extract is a true positive or a false positive.

Using this prediction on short periods of times, I was able to generalize it to a whole record by splitting the record into several periods of 2 seconds. This gave the probability to ear one species at each period of the record. For the full record I am keeping the maximal probability.

Using that first pipeline I submitted my results in Kaggle which gives me a score of 0.61278 and a rank of 878/1143.

One of the most important problems of this first submission is that I am not taking the frequency range furnished inside the training data.

For my second model, at the moment of the training I am doing a Fourier transformation, setting all frequencies not in the range to 0 and then doing the inverse transformation to get a filtered recording. When listening an extract, this process allows a better recognition of the species song.

When submitting my results into Kaggle my scores drop to 0.38.

This low score shows the presence of a problem inside the training process. I haven’t found the solution, yet. I was able to spot some species\_id where the probability to find them during full recording was always inferior to 0.5 so I needs to focus on those species\_id.

Future steps:

-Resolve current filtering problem

-Advanced model choice by testing several possibilities (potential use of tpot) and creation of a metric who considers the score on all species.

-Advanced filtering techniques, use of audio transformation more specific and adapted than mfcc

-Use of neural networks in order to provide a better analysis of the complexity of an audio file