**Surgical mask detection**

*Discriminate between utterance with and without a surgical mask*

Along the way, I have used a few classifiers such as Gaussian Naive Bayes, SVC, Random Forest. The one that had the best score was SVC, solution that I have kept for the final evaluation.

**Loading data and features extraction**

I used [librosa](https://librosa.github.io/librosa/) for analyzing and extracting features of an audio signal. Load function loads an audio file and decodes it in an array of **time series** and **sample rate**.



A spectrogram is a visual representation of the frequencies of the sound over time. It’s a visualisation of changes in frequencies or other music signals and how they vary during a very short period of time. I used  a similar technique known as [Mel-Frequency Cepstral Coefficients (MFCC)](https://en.wikipedia.org/wiki/Mel-frequency_cepstrum). The main difference is that a spectrogram uses a linear spaced frequency scale (so each frequency bin is spaced an equal number of Hertz apart), whereas an MFCC uses a quasi-logarithmic spaced frequencyscale, which is more similar to how the human auditory system processes sound.



**Description of the Machine learning approach**

The most successful model that I used was SVC. SVC is responsible for finding a boundary between classes (with or without mask in our example).

The maximum score obtained was for SVC(C=1, kernel='rbf', gamma='scale') with **l2** normalization:

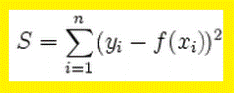
* **C**: Regularization parameter. The strength of the regularization is inversely proportional to C. It controls the trade-off between smooth decision boundary and classifying the training points correctly.
* **Kernel:** Specifies the kernel type to be used in the algorithm. ‘rbf’ uses a non-linear hyper-plane
* **Gamma**: Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’.

if gamma='scale' (default) is passed then it uses 1 / (n\_features \* X.var()) as value of gamma. The higher the gamma value it tries to exactly fit the training data set.

**Normalization**

Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. It is required only when features have different ranges.

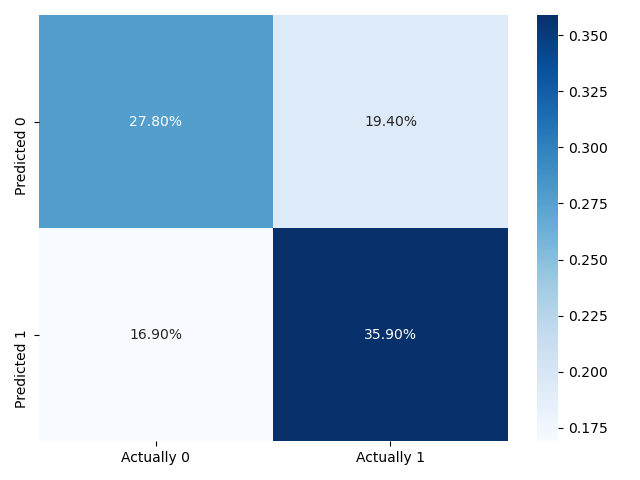
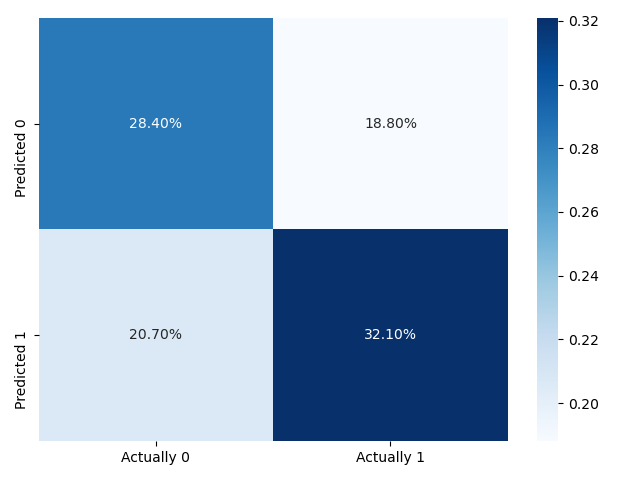
I used **l2 normalization** to scale train and test data: - also known as least squares



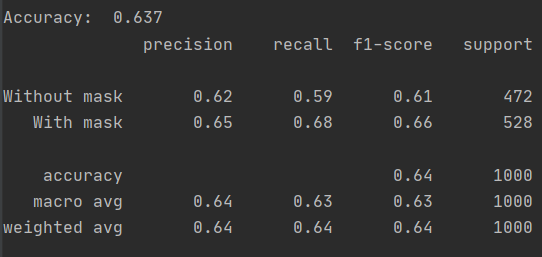
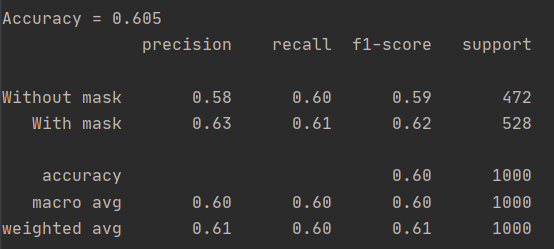
**Algorithms performance**

* GaussianNB (*left*) vs. SVC(C=1, kernel='rbf', gamma='scale') (*right*):

*Confusion matrices*



*Accuracy, precision, recall and other stats*



**Conclusions**

This project clearly teaches me the basics of machine learning, how to debug and how to properly use various libraries. Even if I felt lost sometimes and I was struggling in finding solutions, it was an interesting experience overall. The topic approached makes this competition a good story to tell in front of your non-geek friends.