Artificial intelligence project

Cinca Adrian

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 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**.

```
audio, sample_rate = librosa.load(file_name,res_type='kaiser_fast')
```

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). 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 frequency scale, which is more similar to how the human auditory system processes sound.

```
mfccs = librosa.feature.mfcc(y=audio, sr=sample_rate)
mfccsscaled = np.mean(mfccs.T, axis=0)
```

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

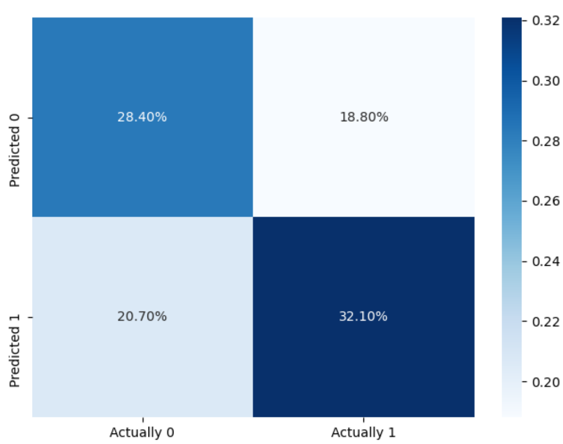
Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale, without distorting differences in the ranges of values. It is required only when features have different ranges.
I used **l2 normalization** to scale train and test data: - also known as least squares

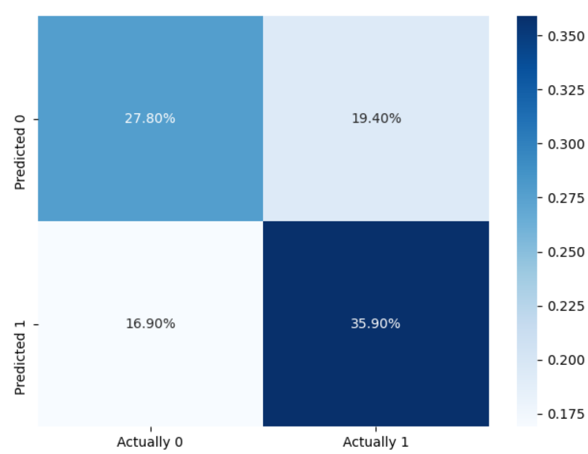
$$\boxed{S = \sum_{i=1}^{n} (y_i - f(x_i))^2}$$

Algorithms performance

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

Confusion matrices





Accuracy, precision, recall and other stats

Accuracy = 0.605

	precision	recall	f1-score	support
Without mask	0.58	0.60	0.59	472
With mask	0.63	0.61	0.62	528
accuracy			0.60	1000
macro avg	0.60	0.60	0.60	1000
weighted avg	0.61	0.60	0.61	1000

Accuracy: 0.637

	precision	recall	f1-score	support
Without mask	0.62	0.59	0.61	472
With mask	0.65	0.68	0.66	528
accuracy			0.64	1000
macro avg	0.64	0.63	0.63	1000
weighted avg	0.64	0.64	0.64	1000

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.