In the beginning, user’s voice is recorded using pyaudio module and then saved in a numpy array called data. Then, Fast Fourier Transform is applied on this signal and since Fourier coefficients are complex, the absolute function is used on them. Also because of the symmetricity of Fourier Transform, only the first half of the array is saved and the rest is dismissed.

And now here is the part where we actually process the FFT. The signal is divided into 4 parts, frequency-wise:

1. 0 - 3000 Hz
2. 3000 - 5000Hz
3. 5000 - 7000Hz
4. 7000 >

The sum of FT coefficients of each part and the total sum is calculated. Hence, we can decide how much powerful each part is in relation to the whole signal (each part’s rate is calculated).

Finally, based on these rates, we can detect whether the user said yes, no or was being mute.

1. If the total rate below 5000hz is very high (say 85%), then the word said is No.
2. If we have a fair amount of energy (say 50%) in the frequencies higher than 3000, then the word said is Yes.
3. If none of the above is true, then the user has not said anything.

It should be mentioned that these numbers and intervals are used based on trial and error.

As with the matter of accuracy, I didn’t face any misclassification during the tests, in which I used my own voice, plus one male and one female from the website : <http://onlinetonegenerator.com/voice-generator.html>

But it better be mentioned that my tests were taken in a rather quite environment. Since we are dealing with recorded signals, noises in the environment might affect the performance of the system.