**Лабораторная работа №2**

1.path\_to\_wave **=** "/content/anna\_voice.wav"

*2.***import** numpy **as** np

**from** IPython.display **import** Audio

**from** scipy.io **import** wavfile

**from** scipy.fftpack **import** fft, dct

**import** matplotlib.pyplot **as** plt

**from** itertools **import** zip\_longest

**import** more\_itertools **as** mits

3. frame\_rate, sound\_data **=** wavfile**.**read(path\_to\_wave)

time **=** np**.**arange(0,len(sound\_data))**/**frame\_rate

sound\_data **=** sound\_data**.**T[0]

print('Sample rate:',frame\_rate,'Hz')

print(f'Total time: {len(sound\_data)**/**frame\_rate}s')

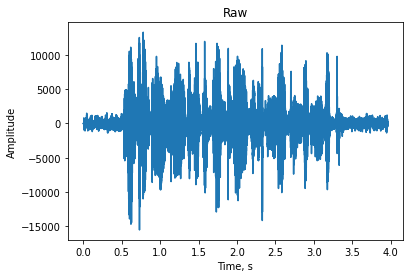
plt**.**plot(time, sound\_data)

plt**.**title("Raw")

plt**.**xlabel("Time, s")

plt**.**ylabel("Amplitude")

plt**.**show()



4. Audio(sound\_data,rate**=**frame\_rate)

5. dimension **=** 2 **\*\*** 8

spctr **=** round(frame\_rate**/**dimension)

overlap **=** 0.25

6. seqs **=** list(mits**.**windowed(sound\_data, n**=**dimension, step**=**int(overlap **\*** dimension)))

prep\_seqs **=** list()

**for** val **in** seqs[**-**1]:

**if** val **is** **not** **None**:

prep\_seqs**.**append(val)

seqs[**-**1] **=** prep\_seqs

seqs **=** [i **\*** np**.**hamming(len(i)) **for** i **in** seqs]

**for** index, element **in** enumerate(seqs):

seqs[index] **=** fft(element)

spector **=** []

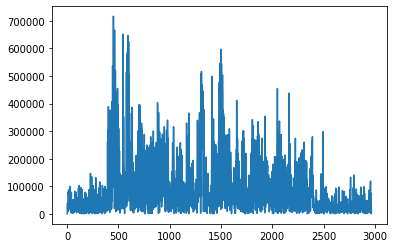
**for** element **in** seqs:

spector**.**append(sum([k **\*\*** 2 **for** k **in** element[:int(len(element) **/** 2)]]) **\*\*** 0.5)

spector **=** np**.**array([el**.**real **for** el **in** spector])

plt**.**plot(spector)

plt**.**show()



7. sr **=** int(frame\_rate **/** dimension)

limit\_k **=** 2

F0 **=** []

trace **=** [spector[k:k **+** sr] **for** k **in** range(len(spector))[::sr]]

**for** element **in** trace:

limit **=** element**.**mean() **\*** limit\_k

**if** element**.**max() **>** limit:

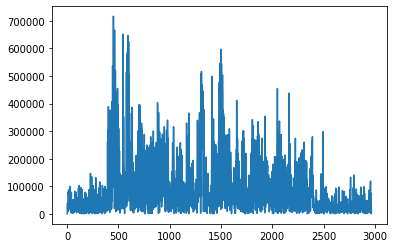
F0 **+=** list(element)

**else**:

F0 **+=** list(np**.**zeros(len(element)))

plt**.**plot(F0)

plt**.**show()



8.length\_f0 **=** len(F0)

**def** calculate\_F(index\_f):

result **=** []

index\_f **+=** 1

**for** i **in** range(length\_f0):

**if** index\_f **\*** i **+** 1 **>=** length\_f0 **or** index\_f **\*** i **-** 1 **<** 0:

result**.**append([0])

**else**:

limit **=** max(F0[index\_f **\*** i **-** 1], F0[index\_f **\*** i], F0[index\_f **\*** i **+** 1])

**if** F0[i] **>** limit\_k **\*** limit:

result**.**append([limit])

**else**:

result**.**append([0])

**return** result

F **=** []

**for** i **in** range(1, 4):

F**.**append(calculate\_F(i))

fig, ax **=** plt**.**subplots()

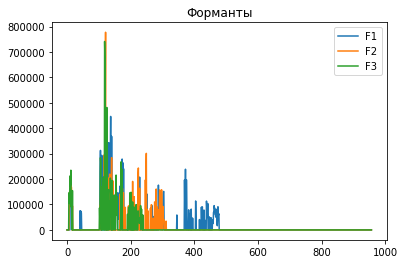
ax**.**set\_title("Форманты")

**for** idx, f **in** enumerate(F):

ax**.**plot(f, label**=**f"F{idx**+**1}")

ax**.**legend()

plt**.**show()



9. trace **=** [spector[k:k **+** sr] **for** k **in** range(len(spector))[::sr]]

amp **=** []

**for** i **in** trace[:**-**1]:

amp**.**append(i)

amp **=** np**.**array(amp)

**import** plotly.express **as** px

fig **=** px**.**imshow(amp)

fig**.**show()

