Ibrahim Akhtar / 21i-1679

Adter the issues with the Audio Feature Extraction were handled i tried to implement the band and minhash implementation we were taught in class. I kept getting errors due to the large dataset of the audio features. In my pursuit of a viable solution i went down a rabbit hole of statistical staistics and linear algebra for Data Science. There i learned that the minhash method is far from ideal when dealing with audio data. There i switched to the Multiple hash method usign prjections of the audio files.

The amount of projections you make increases the accuracy as well as the length of the binary vectors which are used as the hash value. the number of hashtables you make the more false postives you will get in the case of Bvector hashing. After all that was implemented, question 1 had come to an end.

For Question 2 i simply had to extract the features using Afaq's method and insert the data into the hash and compute similar vectors. The similarity coefficient in our program is cosine similarity which i read is best when dealing with audio problems.

**Afaq Alam / 21i-1700**

**Audio Pre-processing:**

Our group agreed on finding 5 MFCCs of every audio file, which would give us a 2D array. We then flattened it into a 1D array for us to be able to easily work on the MFCCs and apply them to our project's fitting.

**Project Findings:**

After traversing through every audio file, the Data Frame consisted of 1180 Rows and 6455 Columns.

**Errors encountered:**

Different column sizes of the audio files. Some of the audio files gave us

more columns after flattening them; 6465 columns. While others gave 6455

columns. To solve this problem, we sliced the greater column number down

to match 6455 columns so they all could fit into the same Data Frame

without any error and the data loss here is only a partial, hence it won't

affect our results significantly.

"*Dequantization failed.*"

This error seem to come up after traversing 10 audio folders. To traverse through the files in each folder, I used os.walk(). After searching online to what could cause this error, but sadly I was not able to come up with a solution.

Fasih Ur Rehman / 21i-1705

An equal division of all work was done amongst the group, with me drawing

the straw to create the front end and the display of *“Something Interesting.”* I continued by creating a simple flask application taking in an input and returning it to the home directory, easy enough.

We had decided that our something interesting would be a graph and description of similar audio files, and to display the graph we would use an exported image using the library *matplotlib.* This seemed the most daunting part, as displaying an image on runtime didn’t seem possible at the beginning.

I created a separate page to display the results provided, which could be accessed when the whole audio processing was done by clicking a button. By then, the image would have been exported to our chosen directory and we could select that image to display. This however provided the most issues, as we were unable to display an image on runtime. Even after a group consultation in the middle of the night, we fell short of displaying the graph we hoped to. We have however displayed a simple text portion explaining our findings.

During this whole process, I found myself spending all extra time trying to make the application look better, adding more pages and even a CSS file to handle the styling. Linking pages in flask is different to what html is, but nothing stackoverflow couldn’t help with.