Module 1 – load\_dataset\_module

**Will need to create the artist ID as it is not present here, and assign it to each non-repeating artist**

First module needs to retrieve only the properties of the songs.

“accoustiness”, “artists”, “danceability”, “energy”, “id”, “liveness”, “loudness”, “name”, “popularity”,”speechness”, “tempo”, and “valence”

Two dictionaries – Dictionaries are unordered, so must be a **nested dictionary** per id

1. Artist and their respective music with the features (artist\_music)
   1. Artist, music name, corresponding features
2. Music features (defined above) (music\_features)
   1. id, respective features

<https://stackoverflow.com/questions/56713360/create-nested-dictionary-from-text-file-using-python-3>

<https://stackoverflow.com/questions/37006534/how-to-get-dictionary-in-python-list-matching-a-given-criteria/37006797>

Module 2 – similarity\_module

5 functions

1. Euclidean similarity
2. Cosine similarity
3. Pearson similarity
4. Jaccard similarity
5. Manhattan similarity

It is important to find all the **mathematical formula** for these and create them for each individual function. Each function needs to include the mathematical method within and accept the necessary amount of arguments (3), suitable exception handling needs to be present here as to not crash the module.

All will accept 3 arguments, these are:

* Artist music or music features (Dict 1 or Dict 2)
* Music id to compare
* Artist id to compare

It needs to be obvious to the program the length of each id, so it knows when you have inputted unmatching ids. (depends on the implementation, look at it later).

**Compute similarity between music tracks / between artists using all features chosen before (don’t need to use all if all are not necessary).**

Considerations for the similarity score from the dataset: (suitable similar metrics)

* Acousticness – defined by genre, might not be worthwhile over speechiness
* Energy - Revolves around the genre
* Liveness – possibly not useful
* Tempo - different genre have different tempo
* Valence - Describes the musical positiveness conveyed by a track (sad music has low valency)
* Speechiness - the user might prefer music with no vocals
* Popularity – its possible to use this, but not likely, keep it under consideration

A main() function is used to call the methods and run the similarity measures and output a result to the user. Both modules need to be created and imported beforehand.

**Mini report (5 pages)**

You will submit a mini report. The report should provide analysis of the problem being solved, justification for your design decisions and pseudocodes for the similarity metrics and other functions you have developed. It should explain the relationships between the modules. A good report should provide evidence of critical analysis of the implemented system. Even if your application does not work correctly, you should still submit the mini report explaining what you have done, what works and what has not worked.