CS 6170 - Final Project Report

Yash Gangrade (u1143811) and Siddhant Ranade (u1141131) April 30, 2019

Contents

List of Figures

1 Introduction

Music in general is rich in structures and patterns. If one looks at the sequence of notes that make up a song, one can usually find repeating patterns, for instance, verses will often have exactly the same tune. One can also observe that certain structures or phrases occur frequently. On a more fundamental level, the notes themselves have a certain structure with regards to how we perceive them. We hear "pitch" in the log scale in terms of the frequency, i.e. we perceive two frequencies to be musically one semitone apart if their frequencies are related by a certain multiplicative factor. This implies that the notes can be thought of as lying on a circle (see figure 1), and that we can define a distance between two notes that respects this circular structure. This suggests the existence of meaningful topological structures, and this is just looking at a single note.

Having the basic idea of Music data, we wanted to explore the topological structures and features in the data. To gain some insights from the music data, we started off the project by collecting several midi files from the available online sources. Next step is to develop a way to convert a song to the point cloud. Once we have the point clouds of all songs, we have to get the time series embedding to encode the temporal information of songs. We are computing the persistent homology for the point clouds and using the persistence diagrams as features for different machine learning tasks described in the subsequent sections.

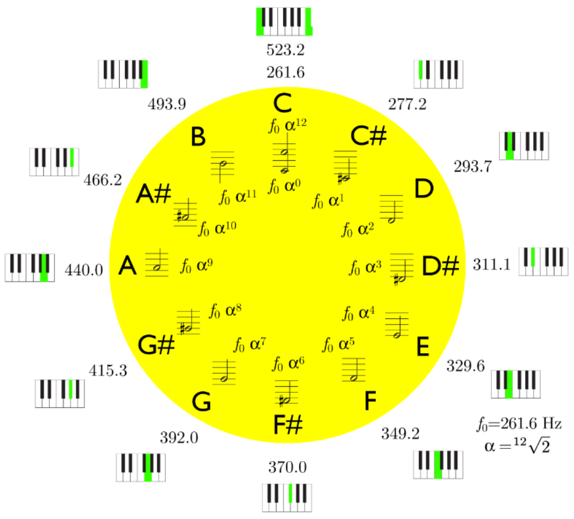


Figure 1: Circle of notes, from [7]

2 Technical Contributions

Through this project we aim to understand the music data in depth and analyze it to extract the topological structures, features, trends etc. in the data. We read a few research papers dealing with the topology in music. The current state of the art systems developed and utilize the similarity measures at the note level, beat level, chord level, and even at chord combinations level. It also validates this analysis by recovering the well-known topological structures such as the circularity of octave-reduced musical scales, the circle of fifths, and the rhythmic repetition of timelines.

Through the course of our project, we realized that our project objective should not only be extracting the topological structures but also to analyze them to gain new perspectives and find interesting patterns using those structures. We are using the resultant persistence diagrams from persistent homology as features for different machine learning tasks. We also want to answer some general queries and finding trends which earlier were not visible. The methodology behind the tasks is defined in detail in the subsequent sections.

- Develop point clouds embedding which preserves the temporal information of the data.
- Compare the work of multiple artists. Identify which artist have the most diverse set of songs and which have narrow field of work in Music.
- Comparing how different genres mix within each other. For eg. some rock songs are written with a background score being classical.
- Using topological features for Genre Classification and Artist Identification.

3 Data

We are dealing with MIDI (Musical Instrument Digital Interface) format, which records musical"events" (i.e. playing a key on the keyboard) as "messages", which describe which key was played (pressed or released) at what time and with what intensity ("velocity"). Essentially, a MIDI file is considered as a full description of the song. It can be played back with "sound font". MIDI files for several songs are available online from Bitmidi and Classical Archives . This data is perfect for our project because it will allow us to extract the notes directly, as opposed to say audio data where extracting notes is in itself a challenging task. The tracks which defines the song in the best possible way are extracted from the midi files.

4 Background and Related Work

Budney et. al. [7] apply persistent homology to musical data. Out of all the other papers, this paper is what we considered as a main reference to understand the theoretical aspects of the music data and also to understand the topology behind it.

In addition, we have (very briefly) looked at some other papers. Catanzaro [8] uses Euler's classical tonnetz to represent the musical intervals in a lattice to do modern generalizations like Riemannian geometry. On a very different note, Partch et al. [9] try to represent a tonality diamond structure where each of the two axes represent significant musical intervals. Chew E. [10] discusses various musical progressions along a spiral array in "The spiral array: an algorithm for determining key boundaries". Similar to this, the computer based music visualizers [11] demonstrate different moving patterns in real time as music progresses. One of the good use of computational topology was by Lewin [12] who researched an application of simplicial complexes and homology to the music dataset. On the same note, there have been some works by Callender, Quinn, and Tymoczko [13] who explore a topological approach to musical spaces using the voice-leading metric. Similarly, Buteau et al. [14] use topological analysis to cluster the melodies and discover the musical motives behind it. These are just few examples of how the music data as a whole has been explored.

Our work is very different from the previous works in the field. We are trying some higher level tasks using Topological Data Analysis + Machine Learning like Genre Classification, Artist Identification. We are using the methods like computing time series embedding and persistent homology and distance computations to get the results.

5 Methodology

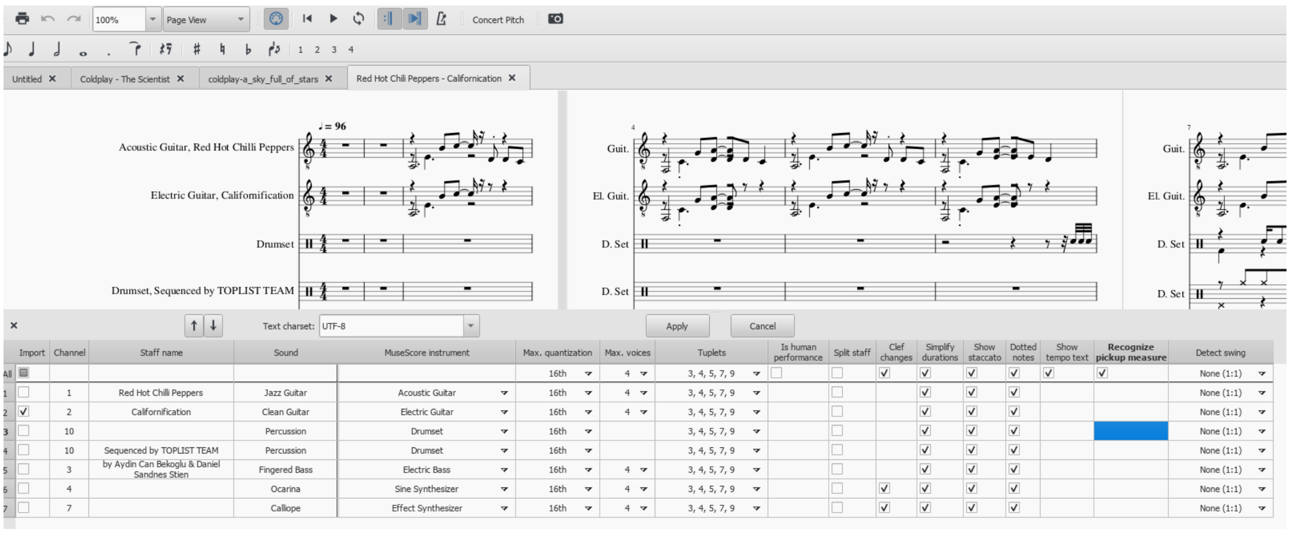
Following subsections describes the steps taken to get the results.

5.1 Data Collection and Processing

The data collection for us has been more of manual labor where we find the data and extract tracks from it manually. We collected 100 songs and processed them.

- Download the midi files for different type of songs, for e.g., pop, rock, classical etc. from any of the online source like Bitmidi and Classical Archives. We have used songs from Queen, Coldplay etc. bands along with solo artist songs such as George Ezra, Adele etc, as well as we have used classical sonatas like Beethoven, Fur Elise etc. Having the diversity in the data provides us new insights and also help us to prove the correctness of the project. This is the raw data for our project.

- Use a open-source software called MuseScore to manually select a channel from the midi files and then exporting it as another midi file. Channel selection is based on the information each channel provides, usually for topological data analysis, piano, melody, guitars are considered good channels because they have some patterns we can utilize. For some midi files, we used a single channel while for some we used all the channels, it all depends on the best representation of the song. Please find the attached screenshot of how the channel selection works in the software.


100%
Page View
Concert Pitch
Untitled
Coldplay - The Scientist
coldplay-a_sky_full_of_stars
Red Hot Chili Peppers - Californication
Acoustic Guitar, Red Hot Chilli Peppers
Electric Guitar, Californification
Drumset
Drumset, Sequenced by TOPLIST TEAM
Guit.
El. Guit.
D. Set
D. Set
Apply
Cancel

Text charset: UTF-8

	Import	Channel	Staff name	Sound	MuseScore instrument	Max. quantization	Max. voices	Tuplets	Is human performance	Split staff	Clef changes	Simplify durations	Show staccato	Dotted notes	Show tempo text	**Recognize pickup measure**	Detect swing
All	☐					16th	4	3, 4, 5, 7, 9	☐	☐	✓	✓	✓	✓	✓	✓	None (1:1)
1	☐	1	Red Hot Chilli Peppers	Jazz Guitar	Acoustic Guitar	16th	4	3, 4, 5, 7, 9		☐		✓	✓	✓			None (1:1)
2	✓	2	Californification	Clean Guitar	Electric Guitar	16th	4	3, 4, 5, 7, 9		☐		✓	✓	✓			None (1:1)
3	☐	10		Percussion	Drumset	16th		3, 4, 5, 7, 9		☐		✓	✓	✓			None (1:1)
4	☐	10	Sequenced by TOPLIST TEAM	Percussion	Drumset	16th		3, 4, 5, 7, 9		☐		✓	✓	✓			None (1:1)
5	☐	3	by Aydin Can Bekoglu & Daniel Sandnes Stien	Fingered Bass	Electric Bass	16th	4	3, 4, 5, 7, 9		☐		✓	✓	✓			None (1:1)
6	☐	4		Ocarina	Sine Synthesizer	16th	4	3, 4, 5, 7, 9		☐	✓	✓	✓	✓			None (1:1)
7	☐	7		Calliope	Effect Synthesizer	16th	4	3, 4, 5, 7, 9		☐	✓	✓	✓	✓			None (1:1)

Figure 2: Channel Selection in MuseScore

5.2 Conversion to Point Cloud

- Read the file using mido library in python. Then we have implemented some helper functions to create point cloud from the extracted data. One of the functions we have implemented makes a list of lists in python where each element is the timestamp in the midi file along with the midi message. Midi message is just the information about which note was pressed on which channel with what velocity (eg. pressure applied on piano key) for what amount of time.

- Given a MIDI file, there are several ways to extract point-clouds for persistent homology. These include a simple single note extraction (i.e. extract the set of all the notes in the song), time-series embedding (i.e. extract the set of all note sequences of length N), chord-class embedding (i.e.

extract the set of all chords in the track), chord sequence embedding (i.e. extract the set of all sequences of chords of length N). Out of these, the time series embedding is the most useful one for this project.

- We implemented a function which preserves the time series while creating the point cloud. Essentially we follow time series embedding to create this point cloud. We decide the number of dimensions we want in the point cloud, for eg. with 3 we will get *(note1, note2, note3); (note2, note3, note4)* and so on. It's a very simple way to encode the temporal information. For our project, we are using the time series embedding for 2 dimensions because as we move to 3D point cloud, the process becomes computationally expensive to run it on our local system. For the two dimensions we use the entire dataset.

5.3 Applying Persistent Homology

Once we have the point cloud, we use another helper function to compute the distances between the points in the point cloud. Here, we also use the fact that the notes are cyclic at every 12 notes difference. Then we have a distances matrix which we pass along to the ripser to compute the persistence diagrams. We use these persistence diagrams as an input and a feature for various TDA + ML tasks.

5.4 Distance Computation

We perform the aforementioned procedure for all the midi files and store all the persistence diagram in form of a list. Once we have this data, we compute all the bottleneck and wassertian distances for all pair of persistence diagrams. It is slightly computationally heavy process because of all pairwise distance computations and we have a decent amount of data at hand.

5.5 Visualization using TSNE, MDS, and UMAP

We plot all the bottleneck and wasserstein distances using the dimensionality reduction techniques namely TSNE, MDS, and Uniform Manifold Approximation and Projection (UMAP) [4][5][3]. The results from all the above techniques can be found in the subsequent section.

5.6 SVM Classification

We also performed the Artist Identification and Genre Classification using SVM over the distances. For the artist, we have about 30 artists like Queen, Coldplay etc. where some of the artists have more songs then others. The test and train set is divided by songs as well artists. For example, some artists are either in training set or testing set. This division is kept like this so as to train a robust model of SVM and get some reasonable results.

6 Results and Discussions

The results for all the methods discussed in the previous section are shown in the subsequent pages.

6.1 MDS, TSNE, and UMAP Plots

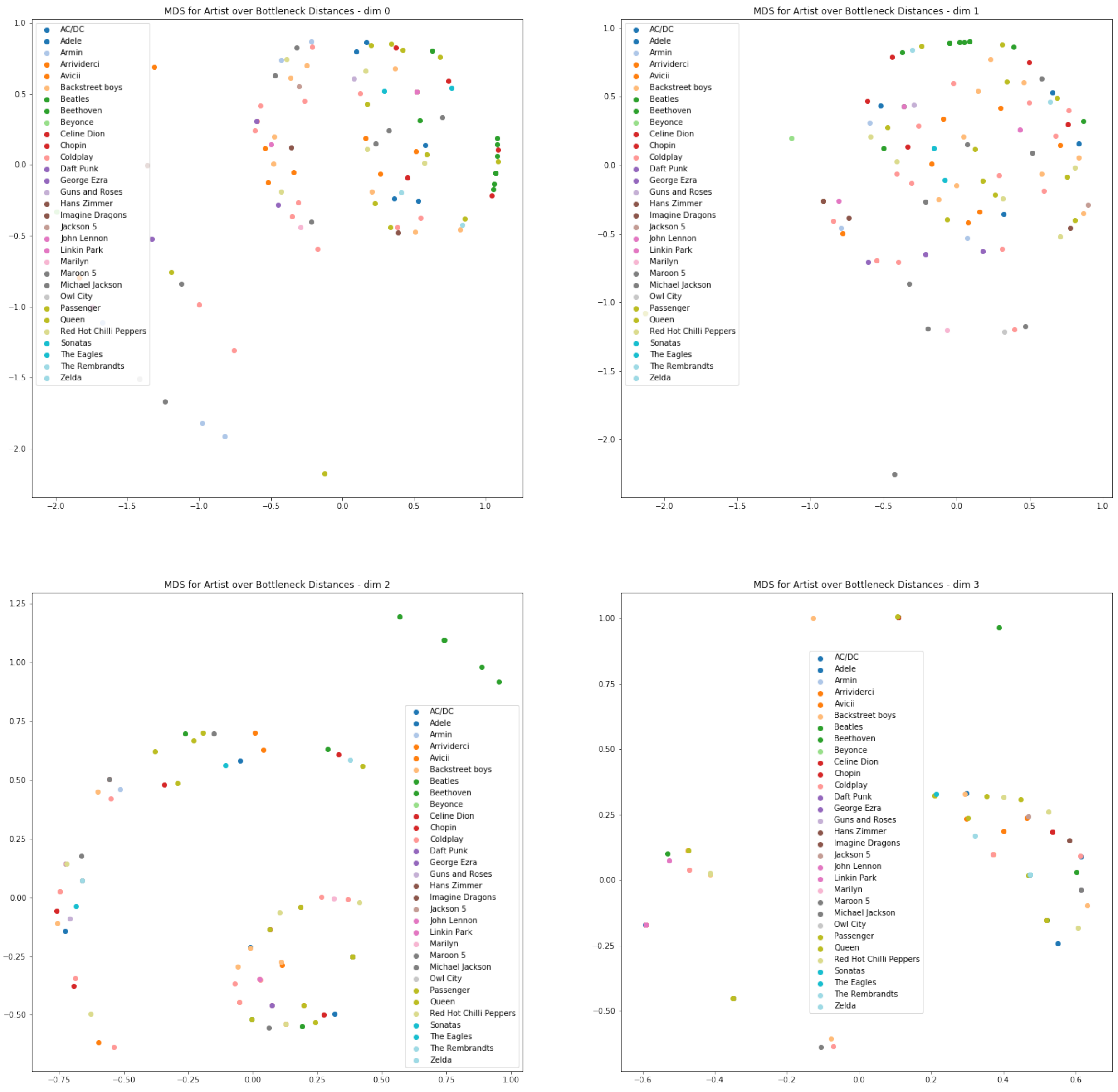


Figure 3: MDS Plots for Artists over Bottleneck Distance. As you can see that some artists have songs which are similar to their own songs or to some other singer.

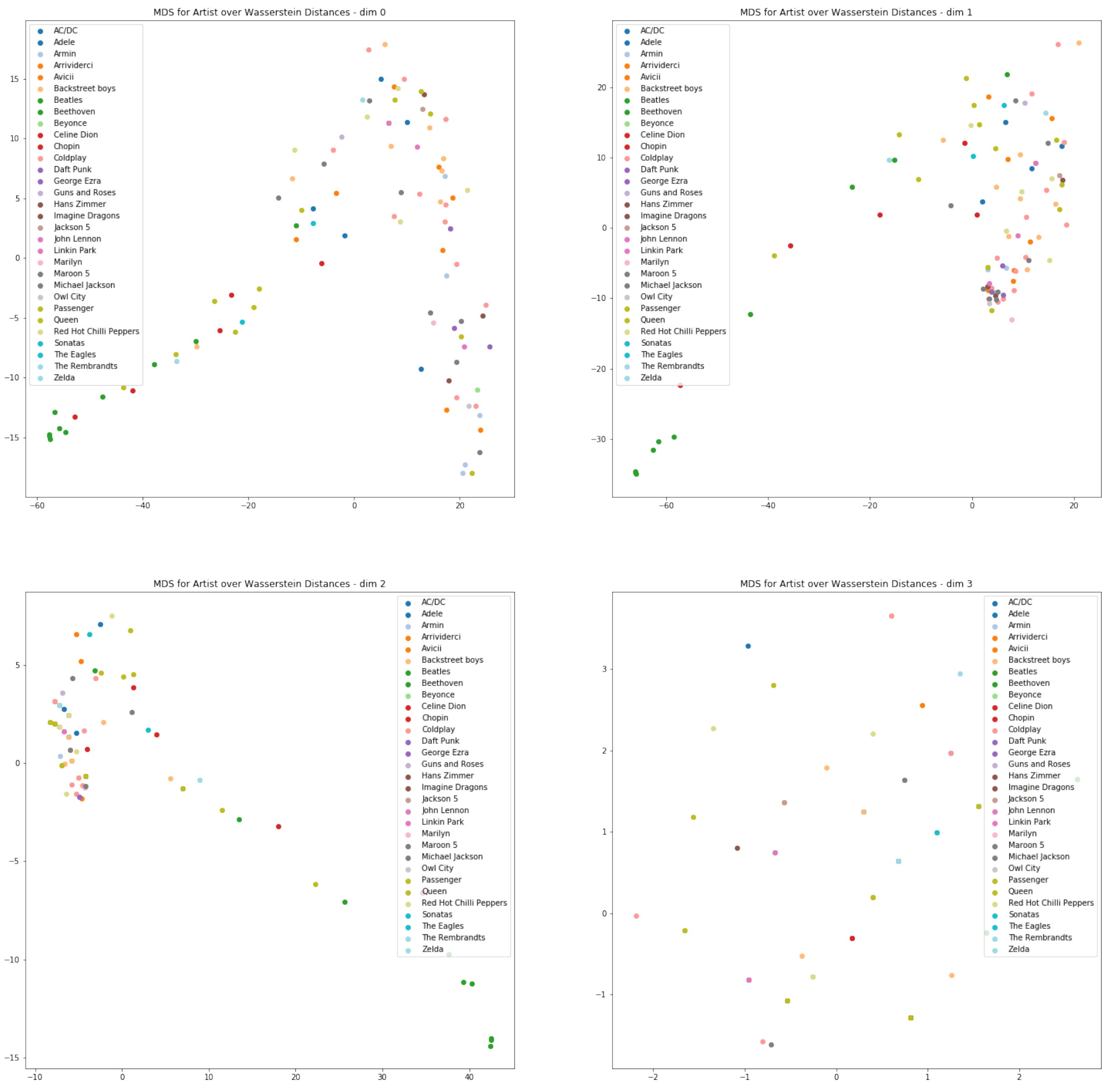


Figure 4: MDS Plots for Artists over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. As you can see that some artists have songs which are similar to their own songs or to some other singer.

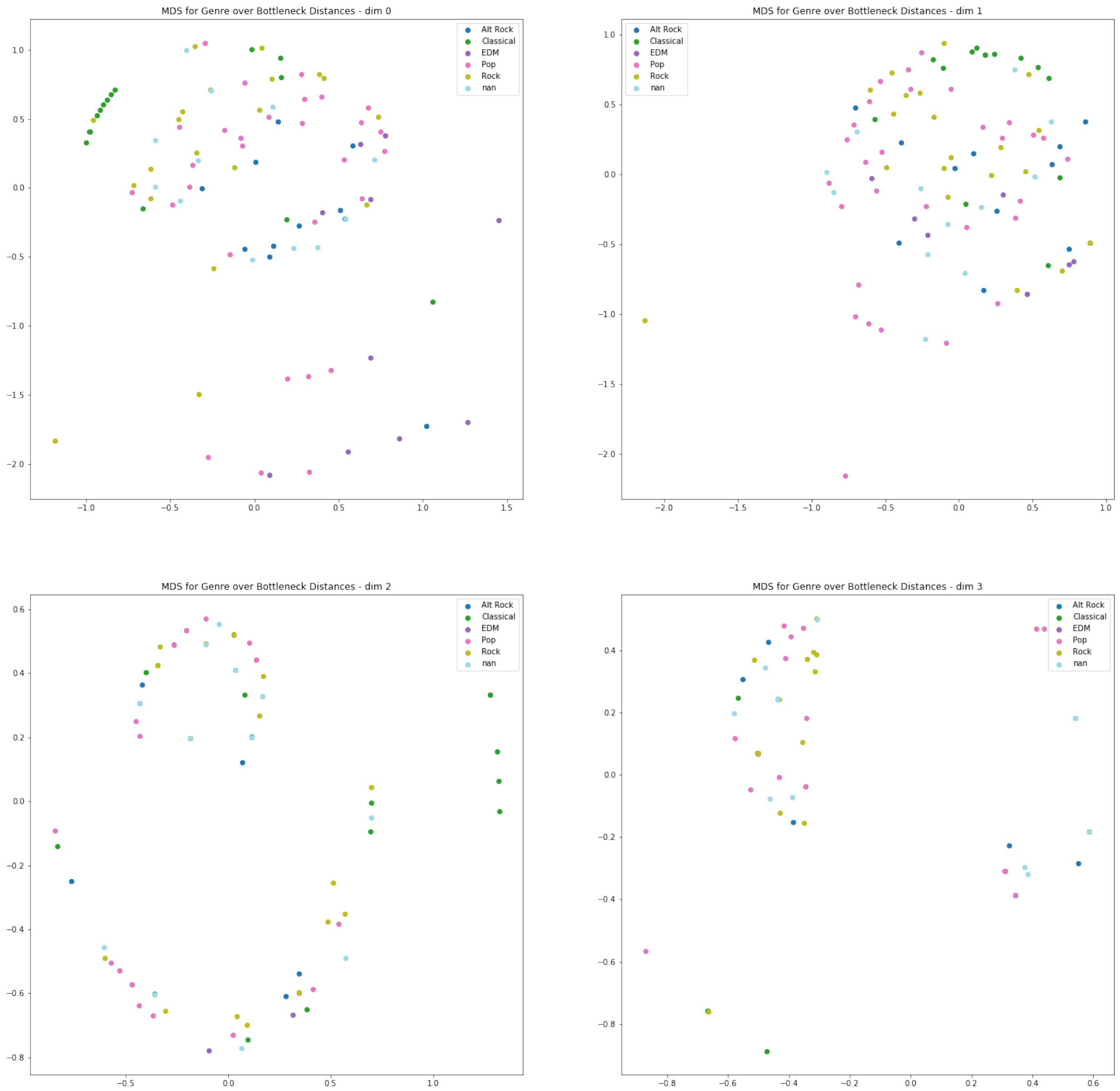


Figure 5: MDS Plots for Genre over Bottleneck Distance. Through these plots we can understand how different genres mix within each other and how different they are.

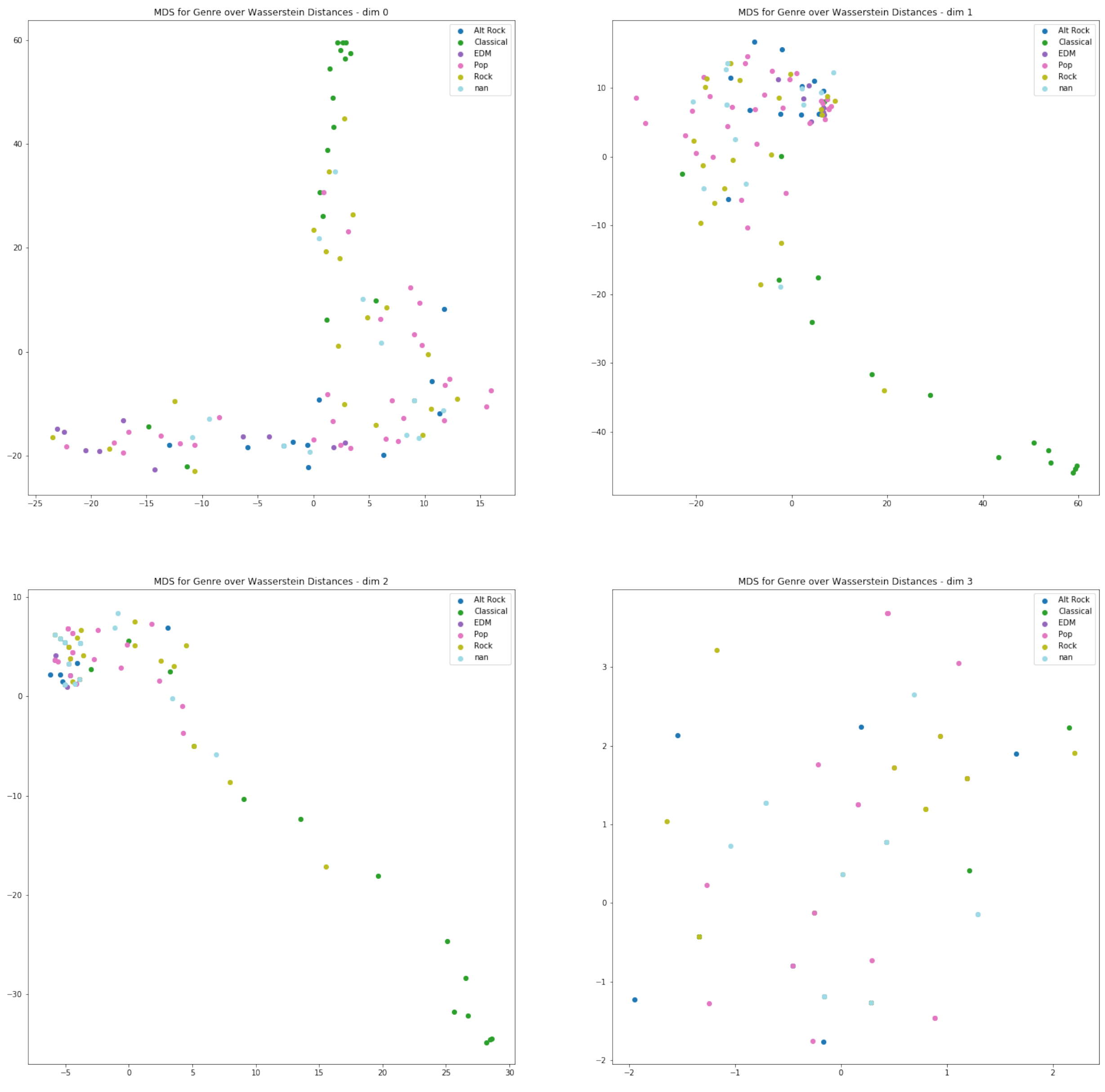


Figure 6: MDS Plots for Genre over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. Through these plots we can understand how different genres mix within each other and how different they are.

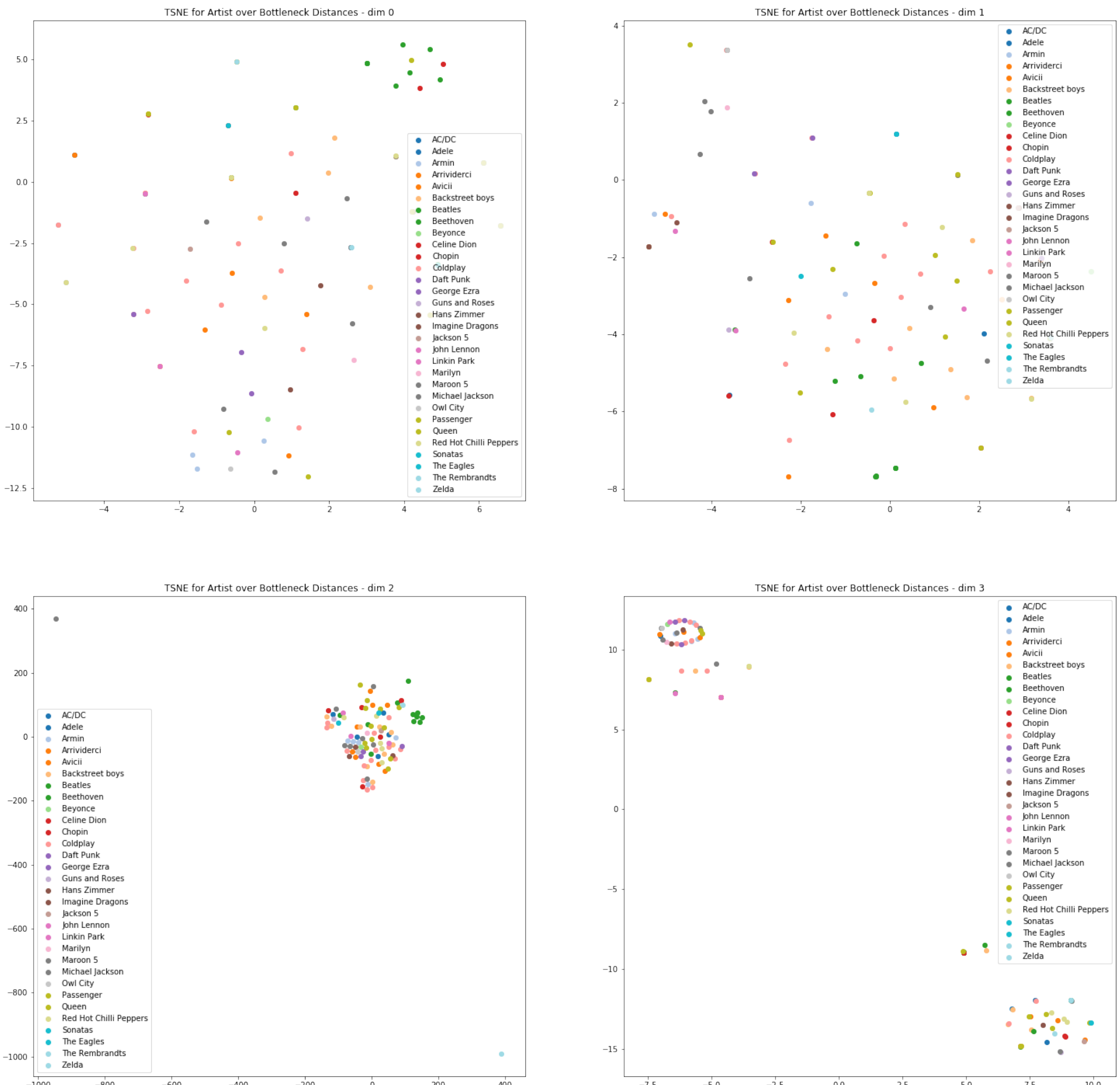


Figure 7: TSNE Plots for Artists over Bottleneck Distance. As you can see that some artists have songs which are similar to their own songs or to some other singer. Especially in Dimension 2, almost all the songs are very close while one outlier song is too far. You can see some clusters in Dimension 3.

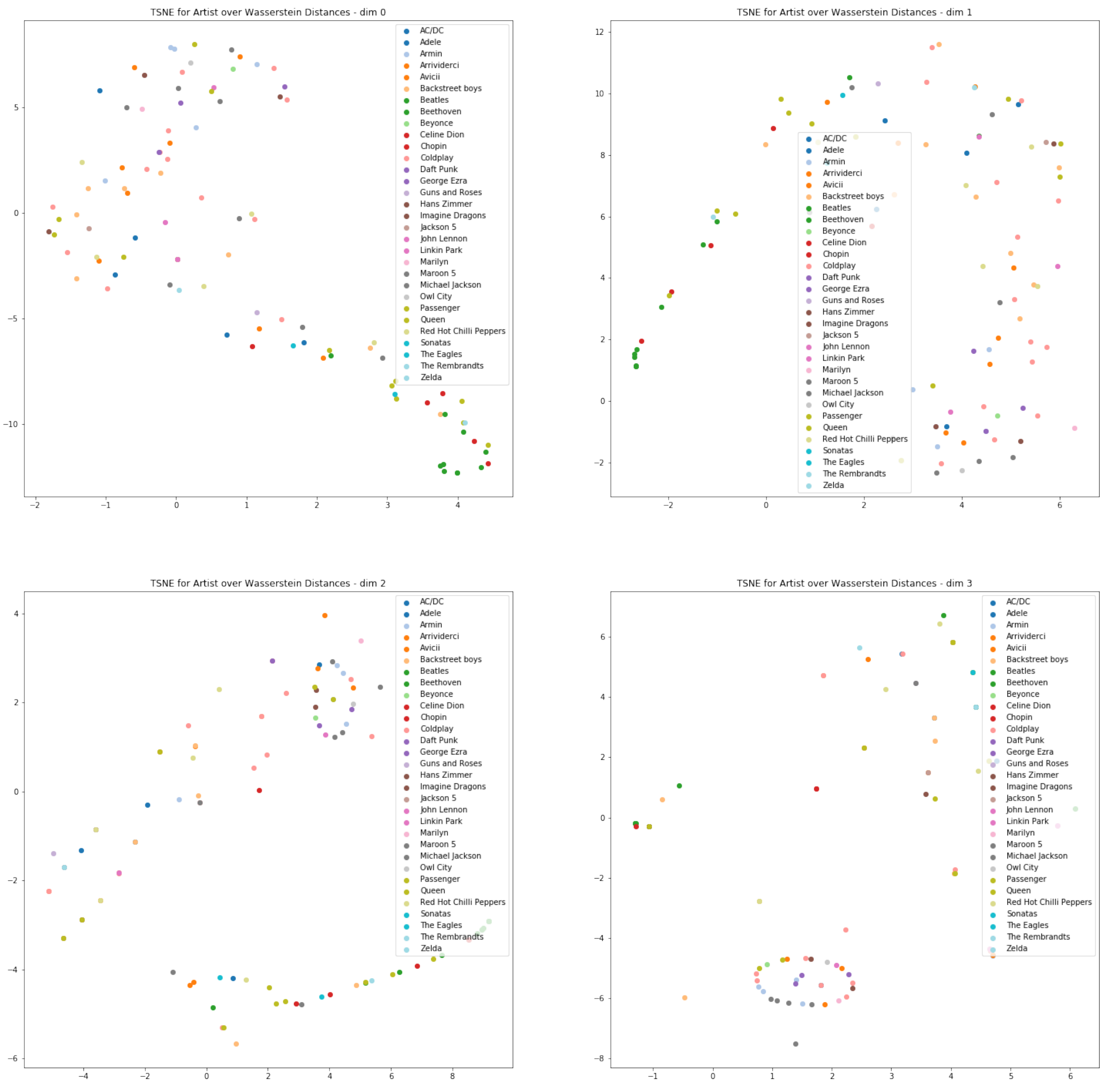


Figure 8: TSNE Plots for Artists over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. As you can see that some artists have songs which are similar to their own songs or to some other singer.

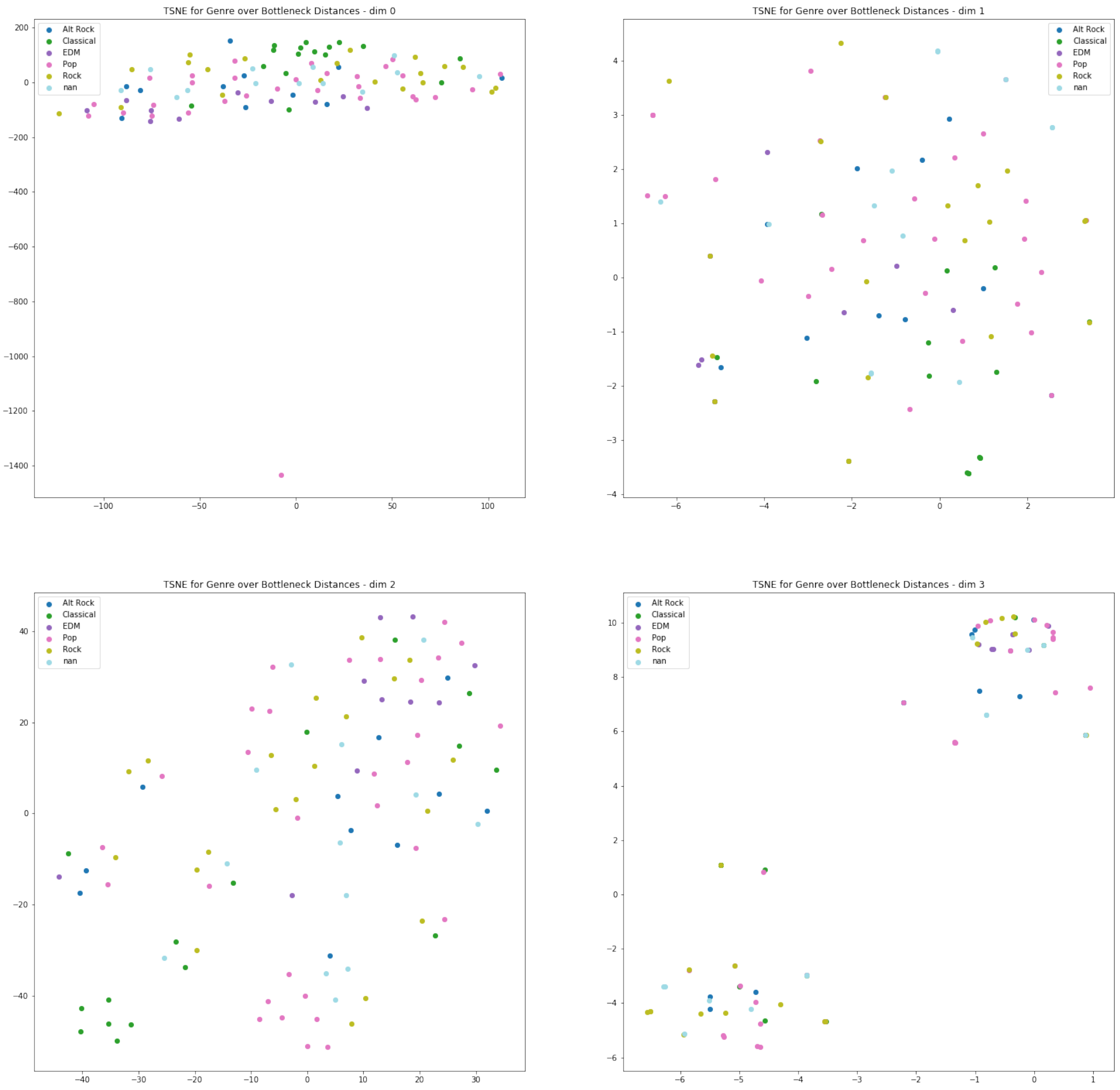


Figure 9: TSNE Plots for Genre over Bottleneck Distance. Through these plots we can understand how different genres mix within each other and how different they are.

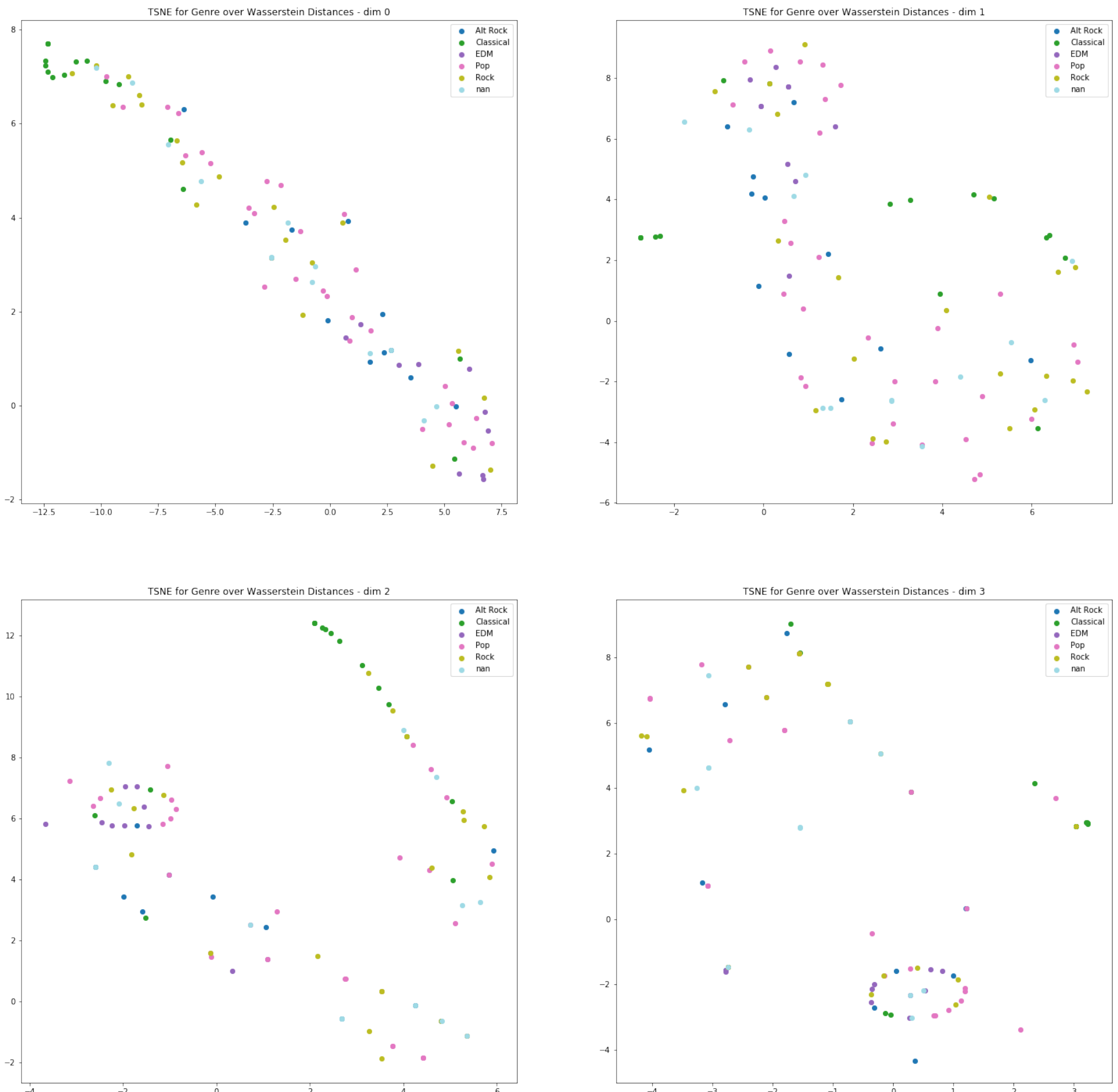


Figure 10: TSNE Plots for Genre over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. Through these plots we can understand how different genres mix within each other and how different they are. There are some clusters but not the dominant ones.

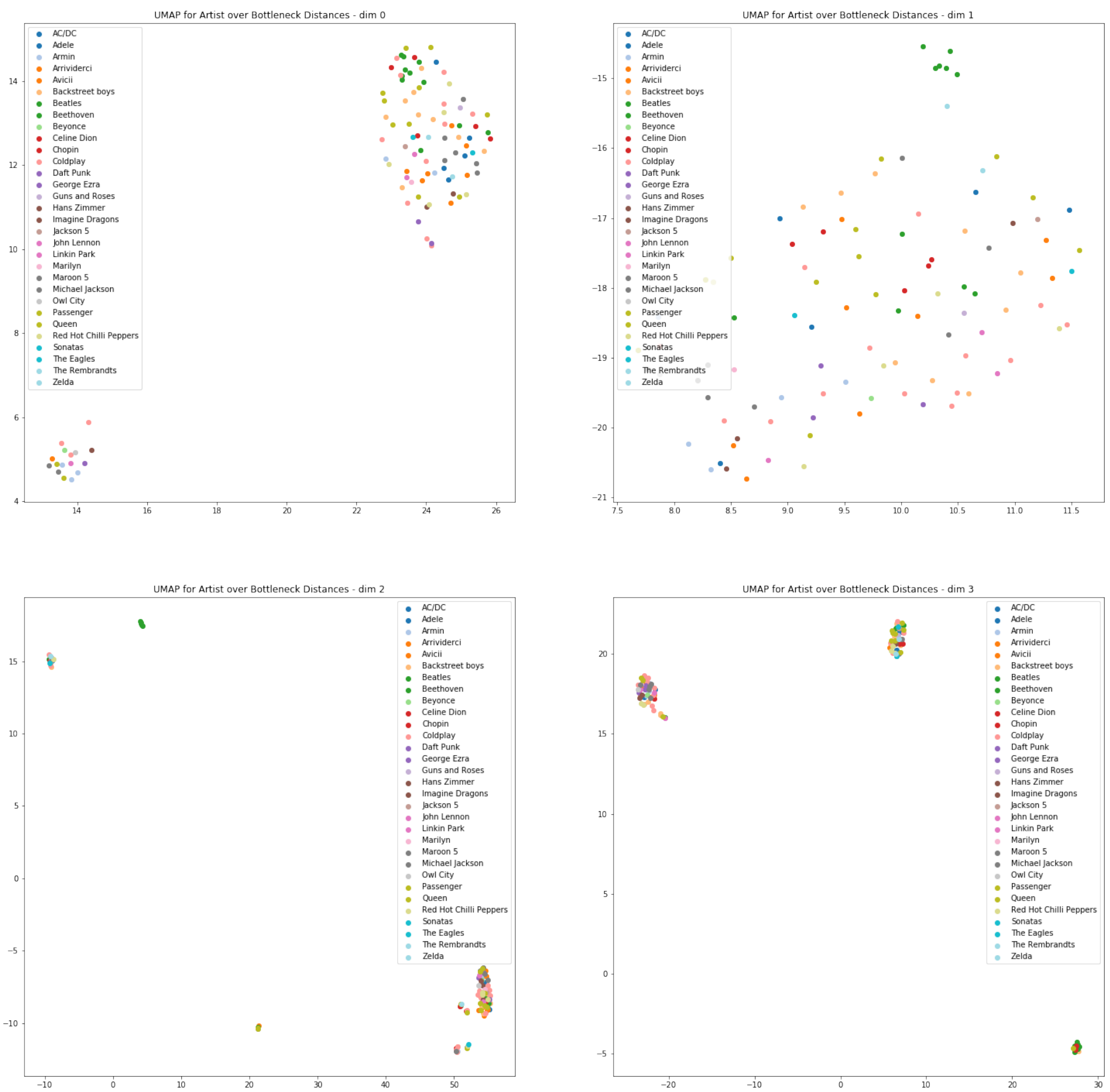


Figure 11: UMAP Plots for Artists over Bottleneck Distance. As you can see that some artists have songs which are similar to their own songs or to some other singer. Especially in Dimension 2, almost all the songs are very close while one outlier song is too far. You can see some clusters in Dimension 3.

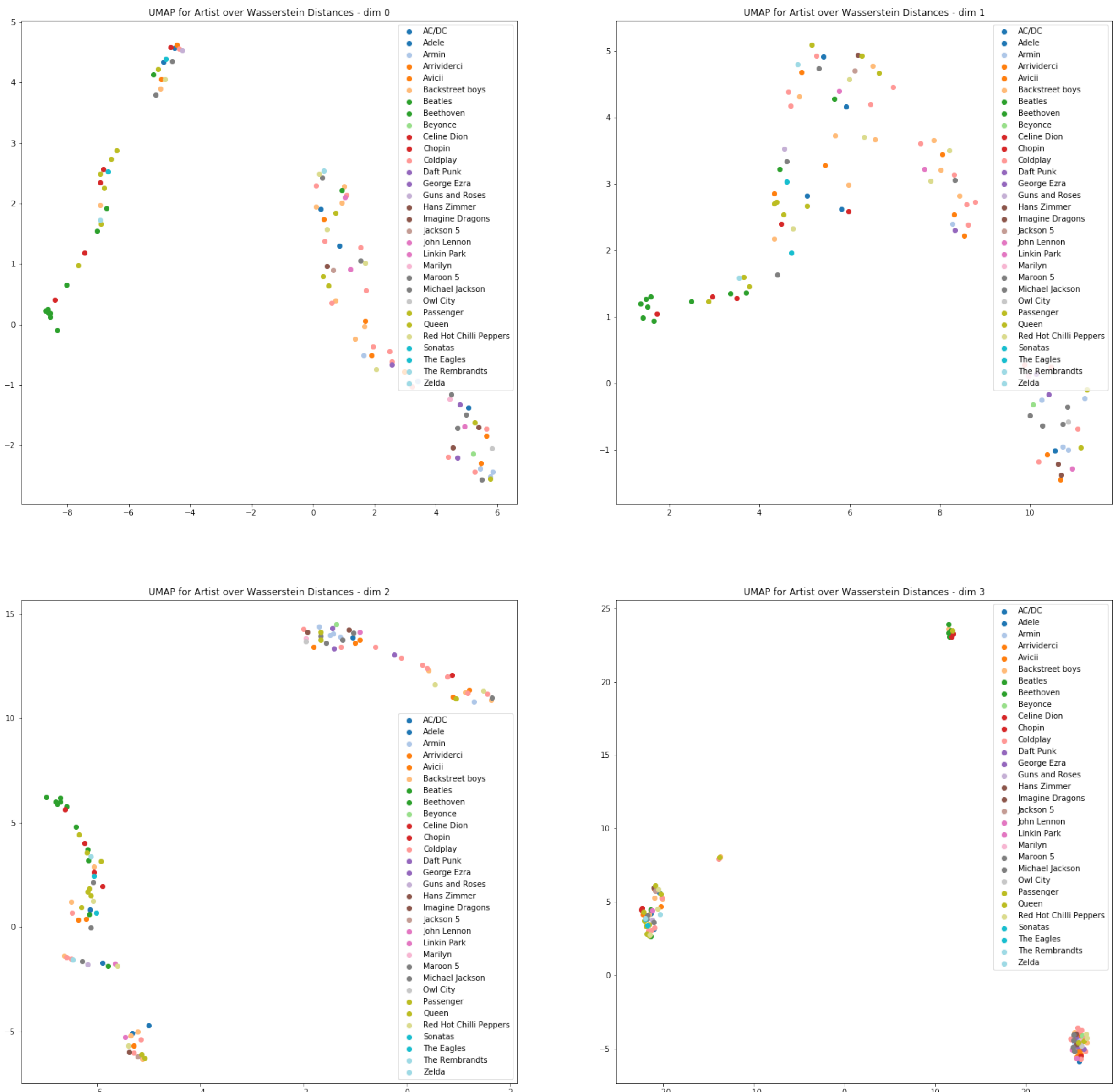


Figure 12: UMAP Plots for Artists over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. As you can see that some artists have songs which are similar to their own songs or to some other singer.

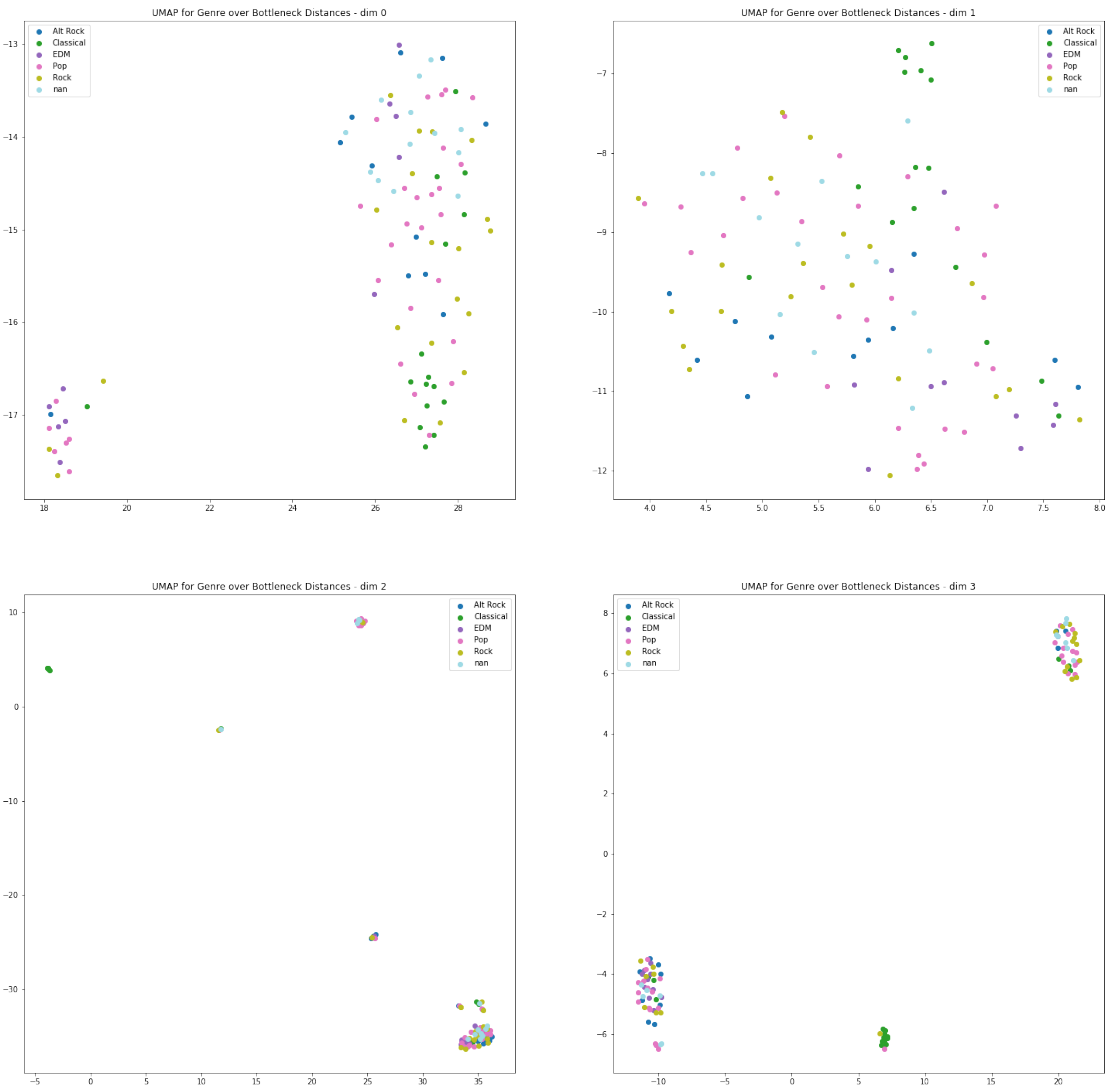


Figure 13: UMAP Plots for Genre over Bottleneck Distance. Through these plots we can understand how different genres mix within each other and how different they are.

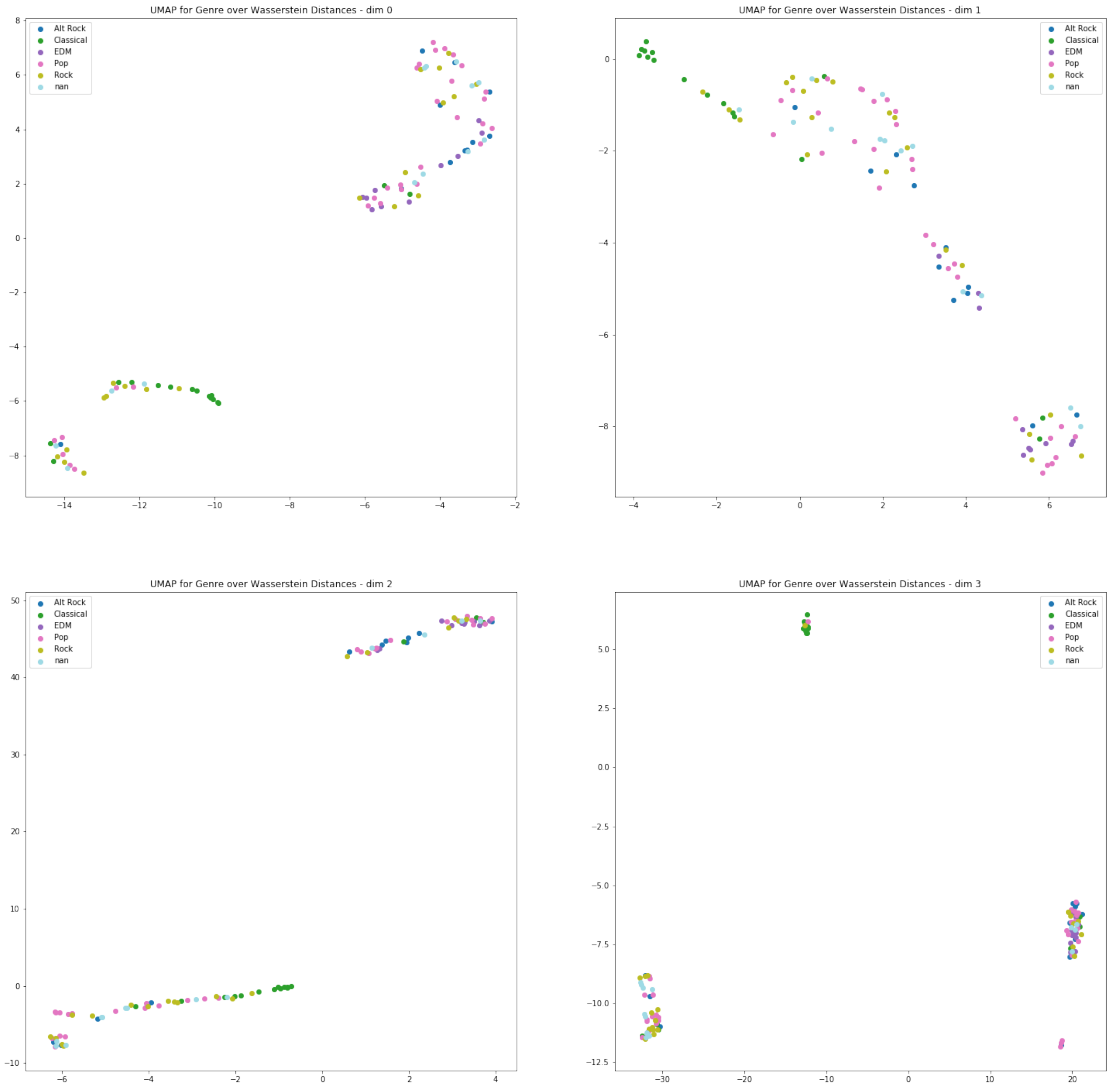


Figure 14: UMAP Plots for Genre over Wasserstein Distance. It's another distance measure here but the results are different in terms of what it looks. Through these plots we can understand how different genres mix within each other and how different they are. There are some clusters but not the dominant ones.

6.2 SVM Results

We ran the SVM for Artist Identification and Genre Classification as described in the Methodology section. The results are as follows;

6.2.1 Artist Identification

For artist identification, we ran the SVM for a testing set containing 11 artists. In this case, the random chance of chosing an artist is 9%. We got the best accuracy of **17%** which is twice as better than the random chance. The confusion matrix is shown below.

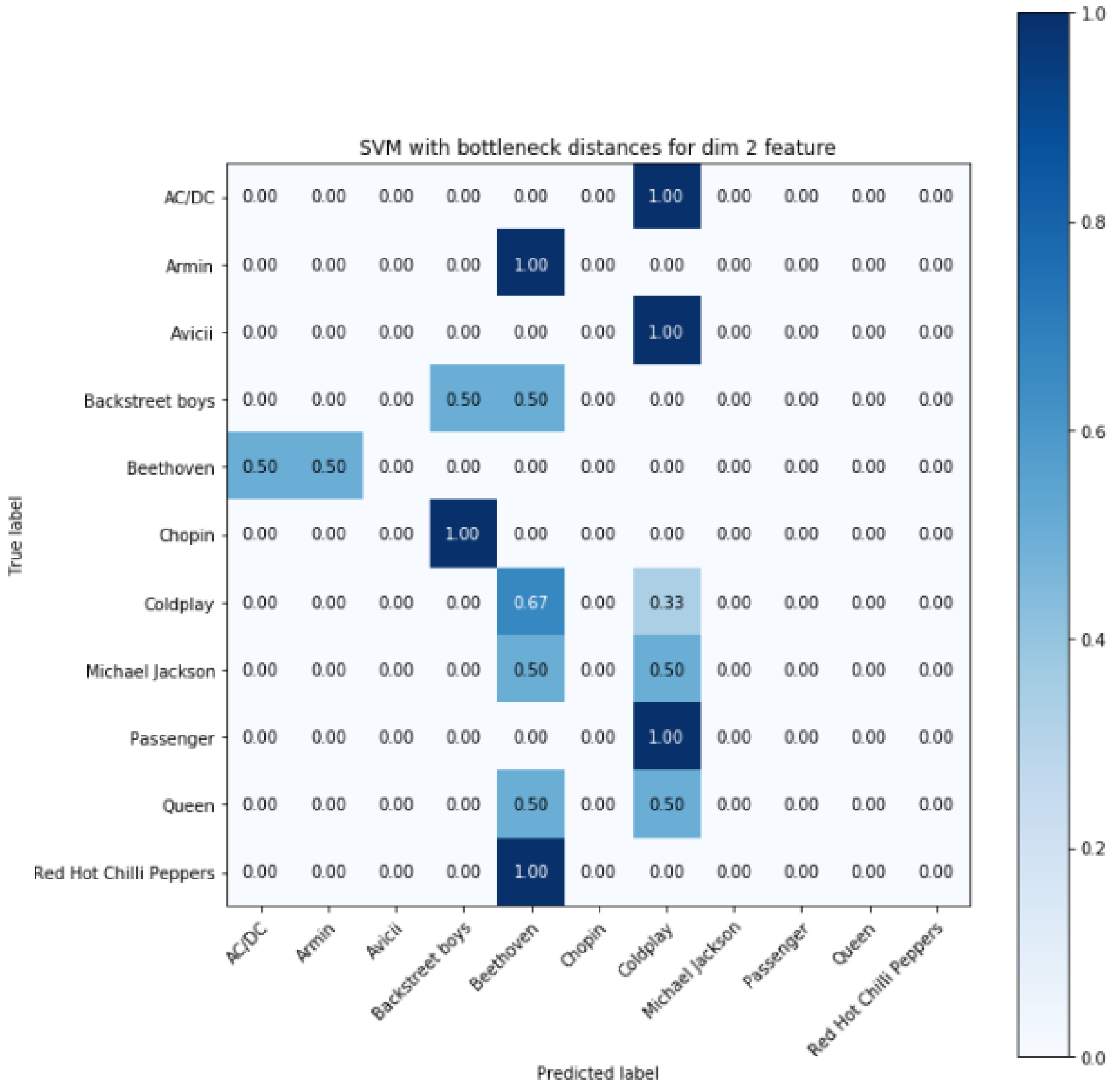
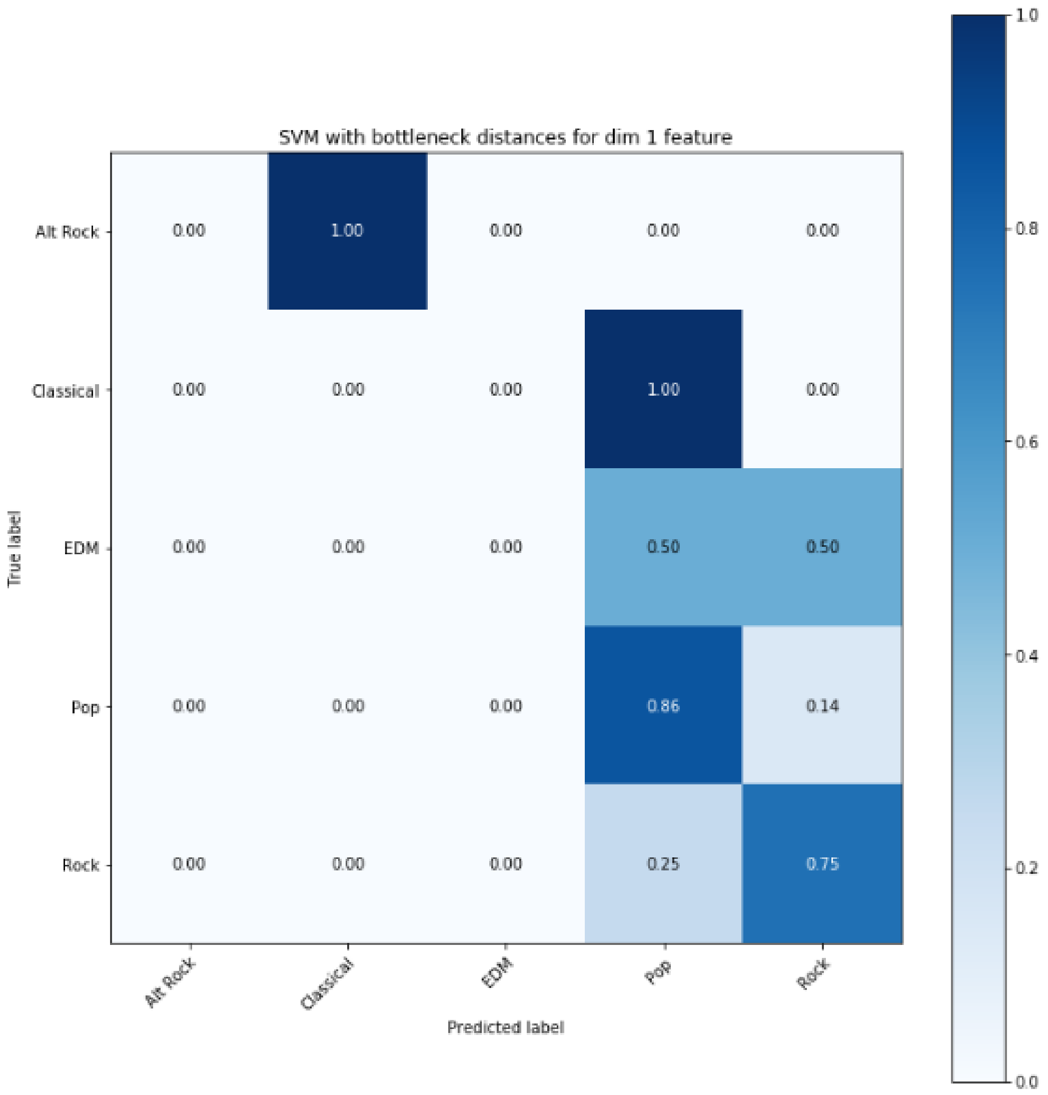


Figure 15: Confusion Matrix for Artist Identification

6.2.2 Genre Classification

For genre classification, we ran the SVM for a testing set containing 5 genres. In this case, the random chance of chosing an artist is 20%. We got the best accuracy of **52%** which is nearly 2.5 times better than the random chance. The confusion matrix is shown below.



True label \ Predicted label	Alt Rock	Classical	EDM	Pop	Rock
Alt Rock	0.00	1.00	0.00	0.00	0.00
Classical	0.00	0.00	0.00	1.00	0.00
EDM	0.00	0.00	0.00	0.50	0.50
Pop	0.00	0.00	0.00	0.86	0.14
Rock	0.00	0.00	0.00	0.25	0.75

Figure 16: Confusion Matrix for Genre Classification

7 Evaluation of the Project

We have proposed several tasks, such as artist recognition, genre classification. These have their own notions of accuracy. We have tried to provide basic understanding of the results through the accuracy of SVM Genre Classification and Artist Identification. For the TSNE, MDS, and UMAP plots, they can be analyzed manually to gain more insights into the data

8 Deliverable

We are handing in the source code which contains well commented ipython notebook. Other folders are just data and reports etc.

9 Conclusion

9.1 What is an overview of your project?

Exploring and analyzing interesting topological structures and answer several real-world questions for the music data.

9.2 What are your project objectives?

We will look at several similarity measures between notes, chords, beats etc. and extract the topological structures and features from the music data. Use these structures for higher level TDA + ML tasks.

9.3 What are the questions project answers?

We get insights into the similarity of songs by artists or songs in a genre through persistent diagrams. Next, we train a SVM for identifying an artist and classifying a genre with a much better probability than random chance.

9.4 What are the key insights based on your results?

Through this project, we found that music if seen from a topological point of view has very interesting patterns and trends which are worth studying. We tried to explore a few of those things.

9.5 What are future directions?

First thing is to automate the initial data processing and collection to get bigger datasets. We had to do everything manually for this project. Second, trying out more complex techniques like Neural Networks to the data to get better results on classification.

References

[1] BitMidi, https://bitmidi.com/, accessed March 6, 2019

[2] ClassicalArchives, https://www.classicalarchives.com/, accessed March 6, 2019

[3] UMAP, https://umap-learn.readthedocs.io/en/latest/

[4] TSNE, https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

[5] MDS, https://scikit-learn.org/stable/modules/generated/sklearn.manifold.MDS.html

[6] kunstderfuge.com, http://www.kunstderfuge.com/, accessed March 6, 2019

[7] Ryan Budney, William Sethares, *Topology of Musical Data*, arXiv:1307.1201

[8] Catanzaro M. "Generalized Tonnetze." J Math Music,2011;2:117–139

[9] Partch H.Genesis of a Music.New York: Da Capo Press, 1974

[10] Chew, E. "The spiral array: an algorithm for determining key boundaries." In Music and Artificial Intelligence - Second International Conference, ICMAI 2002, Edinburgh, Scotland, UK, Sept, 2002

[11] Music Visualization https://en.wikipedia.org/wiki/Music_visualization, accessed March 6, 2019

[12] Lewin D. "Generalized musical intervals and transformations". New Haven: Yale University Press; 1987

[13] Callender C, Quinn I, Tymoczko D. "Generalized chord spaces." Presented at the John Clough Memorial Conference, University of Chicago, 9 July 2005.

[14] Buteau C, Mazzola G. "Motivic analysis according to Rudolph Reti: formalization by a topological model." J. Math. and Music, 2008;2:117-134