Project Report CS210

This is our project that is called the Spotify Song Recommender. Our project question was to make an effective song recommender that would use data management techniques to optimize the songs that would be recommended to them based on a multitude of preferences. Our project aimed to address the data management challenge of creating a recommendation system that would allow for individual preferences and also adapts to a diverse learning portfolio. The core problem of these recommendation systems is data sparsity. The thing about modern recommendation systems is that they currently do not rely on niche attributes and some things that we have noticed were that they base it off just the genre itself. Recommending songs based solely on the genre itself is not actually going to give a song that would be effective in broadening the user's music choices. This project related to our subjects in class because it uses content based filtering, collaborative filtering, and machine learning in order to make effective recommendations and create a model that would show it as well. In this case our model would be a pie chart. The novelty of this project lies in the fact that our recommendation system is unique. We are not like other systems that rely on "content bubbles" and that the user itself will be stuck in the same content or genre when they try to branch out to other music artists. Most spotify systems focused heavily on listening data rather than the song attributes themselves. The project that we have done takes inspiration from that system but we made it more niche hence more effective. We made a filtering system based on the actual attributes of the songs rather than the genres of songs itself. To do this, we visited kaggle.com and found a dataset of the top songs from the 2010's to the 1990's. We believe that this would be a broad enough range of music to compare our songs to. With our current music tastes it seems very broad. We used machine learning in the form of standardizing the data to make sure that all of the features weigh equally. Our dataset has 19 columns with all different attributes of the songs and we want to make sure that all attributes are counted for equally. With that, not a singular column or

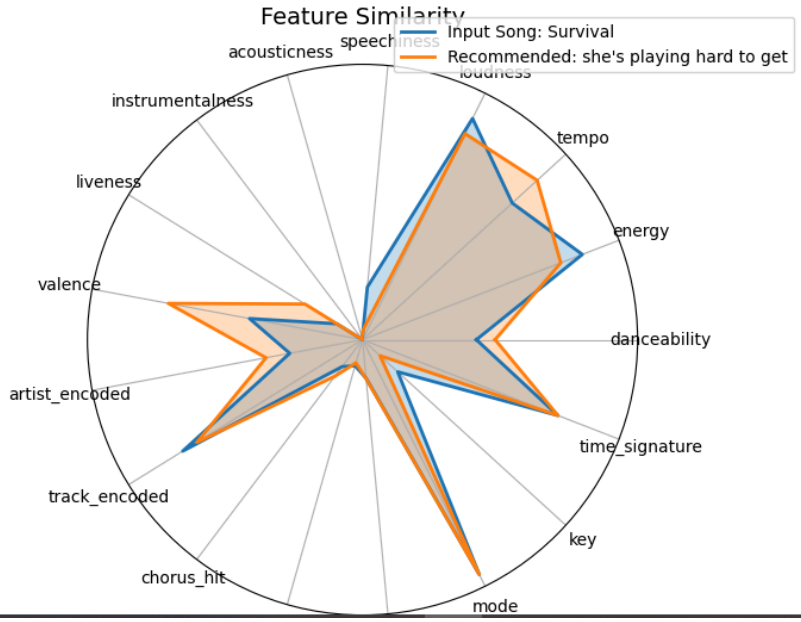
attribute of a song would overpower the rest and skew the recommendation. We also trained the model to learn embeddings. We also stored our results in an SQL database called songs_database.db. A table named SONGS was created and after encoding the data with machine learning using the label encoder and minmaxscaler then the combined datasets are inserted into the table.

For the actual project usage itself, it starts with the user running the code on jupyter lab and it prompts the user to input a song with the quotes “Enter the name of a song”. The datasets that have been provided have “hits” from the 1990’s all the way to the 2010’s. This 30 year time frame allows for a lot of variety as it now allows for the user to get recommended songs from a wide time frame. After inputting the song, our program will list the song name and artist if it is within the database. If the song is not within the database then the program will simply put “the song is not within our database”. Then it will output 5 recommended songs in which the top one with the most similar attributes will be displayed. For example if I were to input “In Da Club” by 50 Cent then the song would not be in the database so the program would terminate and print out “song not in database”. However when I input a song such as “Survival” by Eminem, then the program would list out 5 songs. In this case it is “she's playing hard to get”, “the girl I used to know”, “storm warning”, “tubthumping”, and “make love like a man” in that order. These evaluations were all based on the methods I

listed out earlier with our machine learning and database principals.

```
Enter the name of a song:  Survival
Input Song:
Track: survival
Artist: Eminem

Recommended Songs:
                       track          artist
0  she's playing hard to get         Hi-Five
1    the girl i used to know  Brother Beyond
2              storm warning    Hunter Hayes
3                tubthumping     Chumbawamba
4       make love like a man     Def Leppard
```



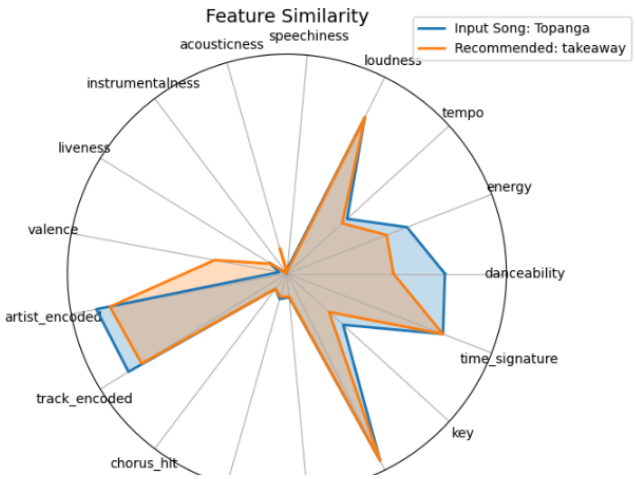
As you can see, the output is a chart that graphs the user's preferred song with all the attributes such as loudness, tempo, energy, danceability, time_signature, key, mode, chorus_hit, track_encoded, artist_encoded, valence, liveness, instrumentalness, acousticness, and speechness. This is all calculated by the cosine_similarity function from the sklearn.metrics.pairwise. It computes the cosine similarities from the earlier pca embedding of the songs. By analyzing the multitude of songs, we can see that many songs from the 2010's and 1990's are very similar despite the time gap. This would be a problem with other song recommenders as stated earlier, many song recommenders including Spotify's own one are stuck in content bubbles. They would simply just give a song within the same genre that is most likely within the same time frame as well. However with our recommender program it is very different, it takes the actual attributes of the song and leaves the content bubble with that along with the

fact that it is such a long time frame, it allows for the user to truly find a more unique song and expand their music taste. We have solved our initial problem! Here is another example just to show you:

```
Enter the name of a song:  Topanga
Input Song:
Track: topanga
Artist: Trippie Redd

Recommended Songs:
                        track  \
0                    takeaway
1     the dream is still alive
2                    valentine
3                  sweet annie
4  walls (from "she's the one")

                                        artist
0  The Chainsmokers & Illenium Featuring Lennon S...
1                                Wilson Phillips
2                                The Get Up Kids
3                                 Zac Brown Band
4                 Tom Petty And The Heartbreakers
```

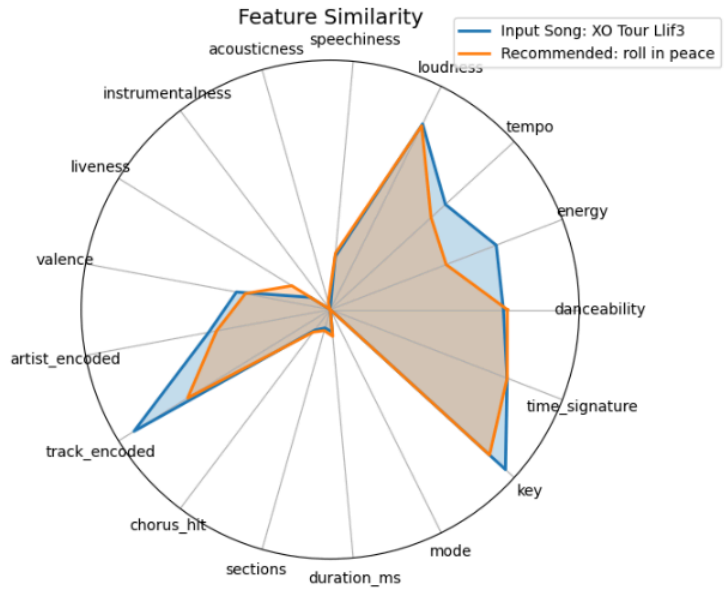


The song that we inputted as actually a favorite of mine. It is Topanga from the artist “Trippie Redd”, the thing is this is a great example because artists in the 90’s were not like Trippie Redd in terms of personality wise. In fact, many artists today are a lot different from those in the 90’s but with our song recommender we can see that there is actually a song that is very similar to Topanga from the 90s! This is just another example of breaking out of that “content bubble”. One last example just to make sure

would be

```
Enter the name of a song:  XO Tour Llif3
Input Song:
Track: xo tour llif3
Artist: Lil Uzi Vert

Recommended Songs:
             track                          artist
0     roll in peace  Kodak Black Featuring XXXTENTACION
1  pretty girl rock                      Keri Hilson
2     the the empty                         Le Tigre
3        to the max          DJ Khaled Featuring Drake
4                 x                Nicky Jam x J Balvin
```



"XO Tour Llif3" by Lil Uzi Vert. He is one of the "newer generation" rappers in which are very different from rappers from the 1990's but with this recommender, it allows for the user to find artists from different era's that are similar, hence breaking out of that content bubble and solving the problem! Our program had a unique aim to break out of that content bubble and I believe that we were successful!