Project Report
SYS 6018: Data Mining
Prof. Matthew S. Gerber
TA: Peter Wu

Team Members
Sakshi Jawarani
Aman Shrivastava
sj8em, as3ek

Problem Statement

The aim of the project is to build an unsupervised framework to **automatically generate playlists** from a given set of songs.

Major music streaming platforms continually look to improve their products and develop features that lead to a more personalized user experience. Playlists are an integral part of their services. With respect to automated playlist generation, currently music platforms generate playlists based on similar artists or genres. This project aims to generate playlists automatically only from a given set of songs without using existing playlists as training data, using a set of diverse qualitative and quantitative features, ranging from song lyrics to features that parameterize the musical properties of songs. Users want a premium service that excels at building playlists, hence premium music streaming services would care about this problem.

Objectives And Metrics

We approach this unsupervised non parametric problem by using clustering methods in which songs are grouped based on their attribute similarity. The objective is to cluster songs such that the songs in the same cluster are as similar as possible, and the songs in different clusters are highly distinct. The average distance within the cluster should be as small as possible and the average distance between clusters to be as large as possible. The Dunn Index[1] is one such metric that assesses the goodness of a clustering, by measuring the maximal diameter of clusters and relating it to the minimal distance between clusters. If the data set contains compact and well-separated clusters, the diameter of the clusters is expected to be small and the distance between the clusters is expected to be large. Thus, Dunn index should be maximized.

Alternative Solutions

The data set that we use for this problem is an amalgamation of multiple music related datasets. Further Relevant data was collected using public APIs offered by music streaming services. Existing research in this domain is majorly focused on approaching this as a supervised classification problem. The songs in the data are tagged based on existing playlists and the model is built to classify a new song to one of those existing playlists based on historical tagging[2] or playlists are generated using seed songs[3]. The various supervised learning techniques that have been explored for this problem include linear and tree based classification methods. We believe that this problem is better dealt as an unsupervised problem since that would enable us to generate playlists that contain similar songs more organically.

Hypotheses and Approach

Looking at other approaches in this domain, we realize that using existing playlists to classify songs into one of them inadvertently introduces a certain amount of bias in the method stemming from the historically tagged data. Additionally, some features in the metadata of the songs (specifically the genre) get disproportionate importance while classifying songs into playlists as the training data available largely is clustered into playlists based on one of these features. We develop a more robust and data-driven approach that minimizes these biases by staging this problem as an unsupervised clustering problem.

We extracted features that holistically capture the musical and lyrical properties of songs and used these abstract features to cluster given songs into a user-specified number of playlists based on their intrinsic similarity. We hypothesized that this will deliver better results that are free of biases in the training data that other supervised methods use.

We used a dataset containing song lyrics and extracted features using Spotify's API to assemble a data set that contains the metadata, lyrical and musical properties of songs. These were then clustered into playlists using various clustering algorithms like k-means[4] and hierarchical[5] clustering. The validity and performance of these methods have been evaluated using internal measures for cluster purity, also since the quality of a playlist is a largely subjective matter, some amount of human assessment has been performed to evaluate these.

Execution and Results

0.1 Data Collection and Feature Extraction

The song lyrics data set from extracted from kaggle[8] and audio features documented in `Table 1` were extracted from spotify for all the songs. Additionally, it was imperative that we numerically capture the representation of the lyrics of the song for a more effective clustering performance. Word vectors trained on the entire text of all the lyrics for this purpose using the `word2vec`[7] algorithm word2vec, which is an unsupervised learning algorithm for obtaining vector representations for words. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space.

duration	tempo_confidence	start_of_fade_out
mode_confidence	key	tempo
mode	time_signature_confidence	loudness
key_confidence	time_signature	end_of_fade_in
danceability	speechiness	acousticness
energy	instrumentalness	liveness
valence	tempo	duration_ms
time_signature	analysis_sample_rate	analysis_channels

Table 1: Features extracted from the spotify API

To capture the sentiment expressed in the lyrics of the song as a numerical feature, we use a lexicon and rule-based sentiment analysis framework called `VADER`[6] (Valence Aware Dictionary and sEntiment Reasoner), that is specifically attuned to sentiments expressed in popular media. We obtain the amount of positive, negative and neutral sentiment along with a compound score indicating overall sentiment polarity.

These features together gave a holistic and a comprehensive numerical representation of each song and were then used as parameters in the clustering algorithm. Songs that were grouped in the same cluster were the ones that were the closest to one another in the high dimensional vector space defined by the parameters that we extracted. To make sure that all the features were given equal importance, all the parameters were scaled using the min-max scaling method.

0.2 Exploratory Data Analysis

To explore relationships and trends in the numerical data collected, a correlation matrix was plotted. We observed that features like danceability, energy and loudness were highly correlated with each other. The correlations observed were in sync with our intuitive understanding of the features and reinforced our hypothesis that human understanding of musical properties were comprehensively represented in the numerical features we extracted. We also plotted trends in all of these features on both the song and the artist level. Please refer to the EDA jupyter notebook for all the plots and corresponding analysis.

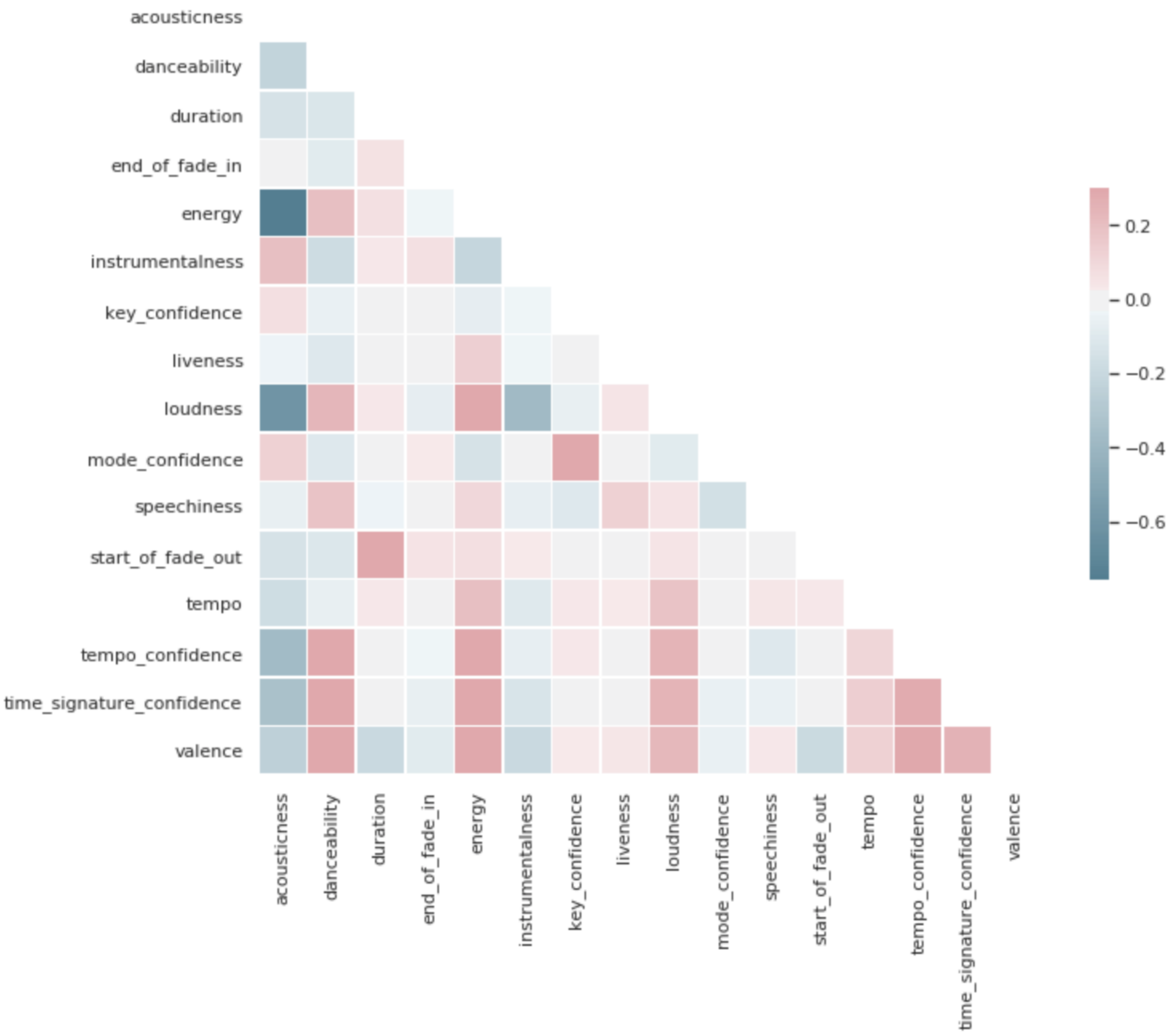

Figure 1: Correlation heatmap b/w audio features

0.3 Playlist generation

We used clustering algorithms to cluster similar songs based on all the numerical features and the sentiment of the songs to generate a user defined number of playlists from the songs selected for analysis. As hypothesised, songs similar to each other were automatically grouped together in a playlist. `Kmeans` and `Agglomerative` (a form of hierarchical clustering) algorithms were implemented to achieve this. Both the algorithms were successful in creating playlists that capture the essence of the songs.

As these qualitatively cluster songs that are close together in the high dimensional space that is represented by the features that we extracted. We see that songs that are similar to each other based on equal representation of all the features that we extracted. Since the features represent the audio features of the song (extracted from spotify), the lyrical elements (using word2vec vector representation) and the sentiment of the lyrics on 4 parameters (using the VADER algorithm) the clusters contains songs that are holistically similar to one another based on all these properties.

0.4 Evaluation

For evaluation of the clusters generated we performed Human assessment on the playlists. These playlists were compared to randomly generated playlists, evaluators were asked to pick the playlist that they think makes more sense based on song similarity. 17 songs were selected and clustered into 3 playlists.

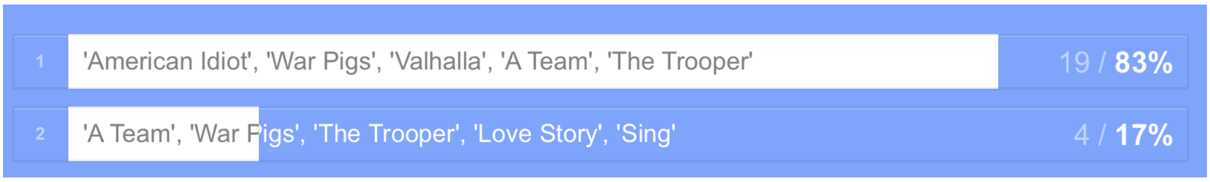

Figure 2: Human Assessment result — Playlist 1

Figure 3: Human Assessment result — Playlist 2

Of the three playlists generated the playlists generated by the algorithm fair better in comparison to randomly generated playlist in all three cases as seen in `Figure 2, 3 and 4`. We found that around 80% of the audience found that the playlist generated by the framework we developed were better than random clustering of songs.

In addition to Human assessment we used within cluster validation parameters to judge the clusters created. `Dunn index` and `Silhouette` score were calculated for all approaches to gauge

Figure 4: Human Assessment result — Playlist 3

the relative goodness of the clustering algorithm for all the approaches and the trend was captured with the number of playlists to be generated.

The number clusters can be defined based on analyzing the plot for Dunn Index(`Figure 5`)/Silhouette Score(`Figure 6`) vs The number of clusters as shown in figures . We select the cluster size for which the metrics are maximized. The user can also input the desired number of playlists to be generated.

For more personalised generation of playlists, the user can define the attributes of the song that should be given more weightage. These parameters can now be used to generate clusters using weighted Kmeans algorithm.

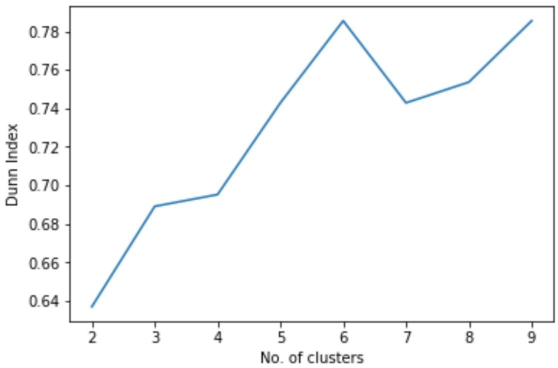

Figure 5: Trend of Dunn Index

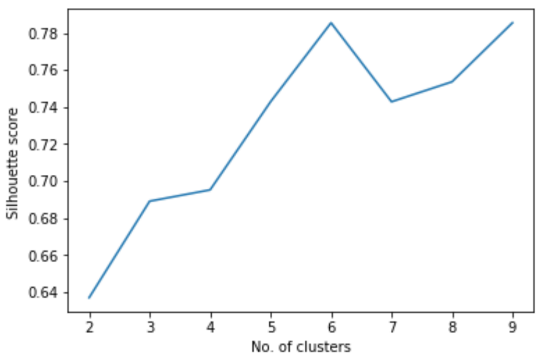

Figure 6: Trend of Silhouette Scores

Conclusion

By treating playlist generation as an unsupervised clustering problem, our approach successfully generates playlists based on intrinsic song similarity. The approach captures the essence and theme of the musical tone as well as the lyrical semantics into numerical features, and uses them to cluster songs together in this abstract high dimensional space defined by these properties. Thus we reduce the time and effort required by a user to generate playlists themselves.

References

[1] Dunn, Joseph C. "Well-separated clusters and optimal fuzzy partitions." Journal of cybernetics 4.1 (1974): 95-104

[2] Gong, Xingting, and Xu Chen. "Automatic Playlist Generation." (1989).

[3] Platt, John C. "Auto playlist generation with multiple seed songs." U.S. Patent No. 6,987,221. 17 Jan. 2006.

[4] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.

[5] Johnson, Stephen C. "Hierarchical clustering schemes." Psychometrika 32.3 (1967): 241-254.

[6] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014.

[7] Mikolov, Tomas, et al. "Distributed representations of words and phrases and their compositionality." Advances in neural information processing systems. 2013.

[8] `https://www.kaggle.com/mousehead/songlyrics`