Project Report

Santhoshi N Krishnan and Wei-Chong Chen
Department of Electrical and Computer Engineering
Rice University
Houston,Texas 77005
snk2@rice.edu
wc39@rice.edu

1 Introduction

Music Chart rankings are a fair indicator of the popularity of a song. For most charts such as the ones computed by Billboard and Apple Music, a lot of features factor into the ranking process other than airplay, such as social media popularity and associated album sales. Song and Genre network features have the potential to influence ranking patterns over the weeks, and play an important role in designing recommender systems for streaming applications such as Spotify.

In our project, we attempt to cover two main objectives. Firstly, we observe the change in ranking prediction accuracy on addition of network features such as song and genre centralities to the prediction features set. Secondly, we also use network visualization measures to identify and draw inferences on distances between genres and change in popularity over the years.

2 Supporting Literature

Machine learning methods have been used to predict the success of a song in various global charts over the past few years.

Datla and Vishnu [2015] used lyrical data as a predictive feature to determine if a song belongs to the top or bottom of the Billboard charts. A Support Vector Machine (SVM) classifier with a radial kernel function on the linguistic features was used, with a precision of 0.76 .

On another note,Ni and Santos-Rodriguez [2011] used only audio features such as Tempo, Time Signature and Note duration to develop a binary classifier to predict the performance of a song in a Top 40 chart. Their results showed a peak accuracy of 60% .

In a study of Social Networks, Bischoff et al. [2009] show that it is possible to accurately predict the commercial success of a song given its the relationships formed in the graph structure of the network. Using sophisticated data mining techniques, they generate a graph from Music Social Networks such as Pandora, Last.fm, and Soundcloud.

We would like to extend this study by using data from Billboard.com, which is one of the primary sources of ranking data in the industry for many years, and incorporates data from all secondary sources such as streaming services, downloads and radio airplay.

3 Data and Methods for Implementation

3.1 Datasets

The Billboard Top 100 list from 2000-2018 along with Spotify song feature data is used as out primary data set, obtained from Google Data set [Tauberg, 2018]. The list has 7573 songs that have charted throughout the 18-year period. The dataset is split into 2 parts:

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

- Training Set
- Testing Set

There were a total of 29 feature vectors that were inherently present in the Spotify dataset, as listed in the table (Spotify.com).

3.2 Features

Both the feature vectors in the training and testing dataset except for the number of ranking related features (weeks, peak_pos, last_pos, rank) were scaled to fit a normal distribution with zero mean, as that is the preferred mode of feature entry into classification/regression models such as SVM. The z-score for a random variable X with mean μ and standard deviation σ can be written as,

$$z = \frac{x-\mu}{\sigma}$$

From the dataset, not all the features are used for the analysis. Inherent musical characteristics such as tempo, instrumentality, energy and loudness are used. An umbrella feature of 'genre' which encompasses a mix of all of these features is also considered. The feature 'genre_num' used in the model is the number of genres of the certain song, for example, if one song has both pop and rap genres, then the number of genres will be exactly 2. And we also take the probability distribution of the genres into account, therefore, the more frequent the genre appears, the more score the score of the genre distribution will be. Ranking features such as position of the song last week (last_pos), highest position (peak_pos) and duration (weeks) on the chart are also used as important parameters for ranking prediction.

3.3 Algorithm and Implementation

There are two major steps involved in the project:

Machine Learning for Ranking Prediction The data was tested using two different machine learning methods: Random Forests and SVM.

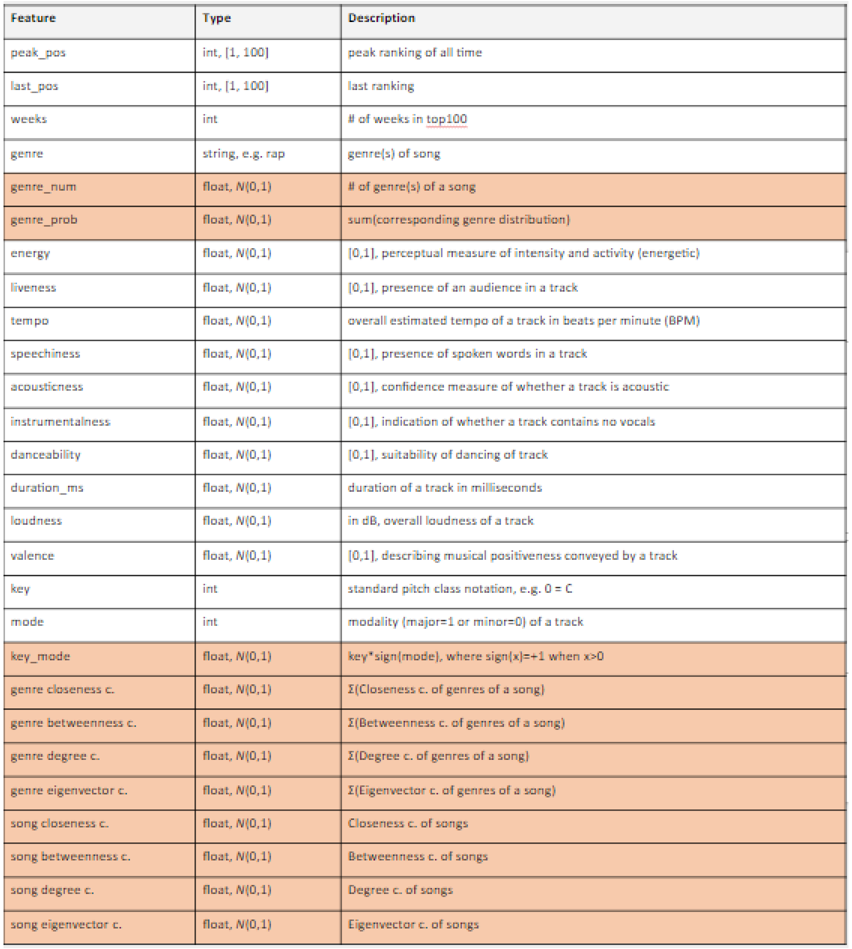
Support Vector Machines(SVM) are a form of supervised learning models that is defined by a hyper plane separating the two discriminating classes (Scikit-learn.org). Given labelled training data, the SVM algorithms models an optimum hyper plane that allows for classification of unlabelled input data. It belongs to the class of non-probabilistic binary linear classifiers. A linear SVM is used for this prediction system.

The Random Forest is another form of supervised learning algorithm and an ensemble of multiple decision trees, trained with the 'bagging' method. It works on the assumption that the merging of multiple decision tree model outputs would improve prediction accuracy and stability over a single learner (Donges [2018]). This method also allows us to measure the importance of different features in prediction output.

Distance Metrics Four classification interval metrics were considered for our problem:

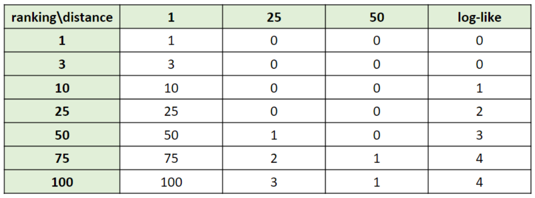
- Distance-1 metric, where, 100 classes were considered with each class corresponding to a single rank group
- Distance-25 metric, where 4 classes were considered with each class corresponding to a group of 25 ranks
- Distance-50 metric, which was essentially a binary classification problem, with each class corresponding to a group of 50
- Log-like Distance Metric with 5 classes with increasing intervals. This category was considered as the change in ranks at the upper end of the charts are more significant than the changes in the bottom end.

Network Analysis on the Dataset Network properties for raw data is obtained. We plan to analyze various centrality measures for the feature data graph, such as:



Feature	Type	Description
peak_pos	int, [1, 100]	peak ranking of all time
last_pos	int, [1, 100]	last ranking
weeks	int	# of weeks in top100
genre	string, e.g. rap	genre(s) of song
genre_num	float, $N(0,1)$	# of genre(s) of a song
genre_prob	float, $N(0,1)$	sum(corresponding genre distribution)
energy	float, $N(0,1)$	[0,1], perceptual measure of intensity and activity (energetic)
liveness	float, $N(0,1)$	[0,1], presence of an audience in a track
tempo	float, $N(0,1)$	overall estimated tempo of a track in beats per minute (BPM)
speechiness	float, $N(0,1)$	[0,1], presence of spoken words in a track
acousticness	float, $N(0,1)$	[0,1], confidence measure of whether a track is acoustic
instrumentalness	float, $N(0,1)$	[0,1], indication of whether a track contains no vocals
danceability	float, $N(0,1)$	[0,1], suitability of dancing of track
duration_ms	float, $N(0,1)$	duration of a track in milliseconds
loudness	float, $N(0,1)$	in dB, overall loudness of a track
valence	float, $N(0,1)$	[0,1], describing musical positiveness conveyed by a track
key	int	standard pitch class notation, e.g. 0 = C
mode	int	modality (major=1 or minor=0) of a track
key_mode	float, $N(0,1)$	key*sign(mode), where sign(x)=+1 when x>0
genre closeness c.	float, $N(0,1)$	Σ(Closeness c. of genres of a song)
genre betweenness c.	float, $N(0,1)$	Σ(Betweenness c. of genres of a song)
genre degree c.	float, $N(0,1)$	Σ(Degree c. of genres of a song)
genre eigenvector c.	float, $N(0,1)$	Σ(Eigenvector c. of genres of a song)
song closeness c.	float, $N(0,1)$	Closeness c. of songs
song betweenness c.	float, $N(0,1)$	Betweenness c. of songs
song degree c.	float, $N(0,1)$	Degree c. of songs
song eigenvector c.	float, $N(0,1)$	Eigenvector c. of songs

Figure 1: Features used in the Prediction Model. The features in yellow indicate those calculated and introduced into the model after post-processing original data.



ranking\distance	1	25	50	log-like
1	1	0	0	0
3	3	0	0	0
10	10	0	0	1
25	25	0	0	2
50	50	1	0	3
75	75	2	1	4
100	100	3	1	4

Figure 2: Distance Metrics for Prediction Classes

- Closeness Centrality

$$C_{Cl}(v) = \frac{1}{\sum_{u \epsilon V} d(u,v)}$$

- Betweenness Centrality

$$C_{Be}(v) = \sum_{s \neq t \neq v \epsilon V} \frac{\sigma(s,t|v)}{\sigma(s,t)}$$

- Degree Centrality

$$C_D(i) = \sum_{j|(i,j) \epsilon E} A_{genre(i,j)}$$

- Eigenvector Centrality

$$C_{Ei}(v) = \sum_{(u,v) \epsilon E} C_{Ei(u)}$$

The code is completely written in python, with network analytics done using the networkx library [Hagberg et al., 2008] and machine learning done using the scikit-learn library.

4 Results

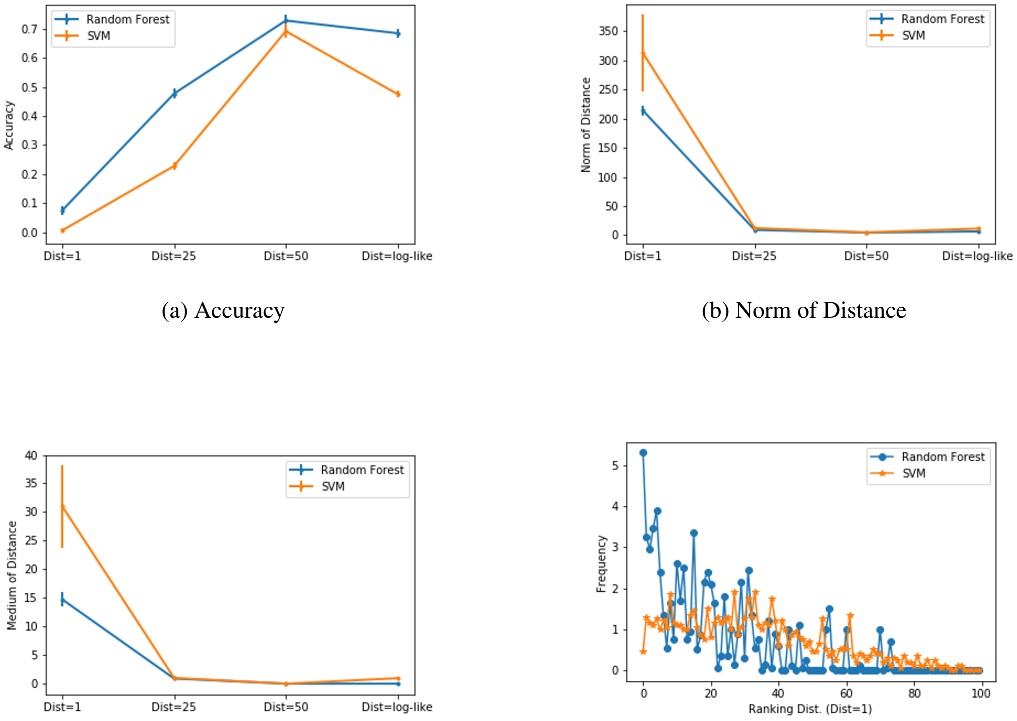
4.1 Ranking Training and Testing for Initial Data

A number of parameters were used to assess the prediction parameters of the SVM and Random Forest methods. They are,

$$Accuracy = \frac{1}{n}\sum_{k=1}^{n} I(y_{test}[k] \neq y_{predict}[k])$$

$$Norm\,of\;Error\;Distance = \| \; y_{test} - y_{predict} \; \|_2$$

Here, we use all the songs from 2000-2018 except for the last week (70 songs) as training set, and the remaining 70 songs as testing set. In Figure 3, it is observed that the Random Forest method performs better than SVM at all four distance metrics, with the exception being for the binary class case, where their performance is comparable. This can be attributed to the fact that linear SVM performs very well in cases where the classes are linearly separated, compared to the other metrics. When the norm and median of distances are compared across classes, both classifiers have comparable performance with the exception at D=1, where SVM is observed to have higher norm of error with high variance. The presence of too many classes (100) explains the variation in performance for SVM.



(a) Accuracy

(b) Norm of Distance

(c) Median of Distance

(d) Histogram of Error Distance for Ranking Dist = 1

Figure 3: Original Data Prediction Results

4.2 Ranking Training and Testing for Data with Genre Network centralities

We constructed a network with the the genre of songs in the data set as the nodes (the weight of the edge is calculated as the number of the songs sharing both genres). We observed that a giant component was present in the network structure, with most of the major genre being a component, as shown in Figure 4.

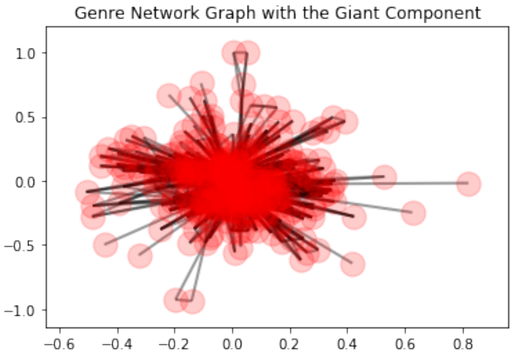
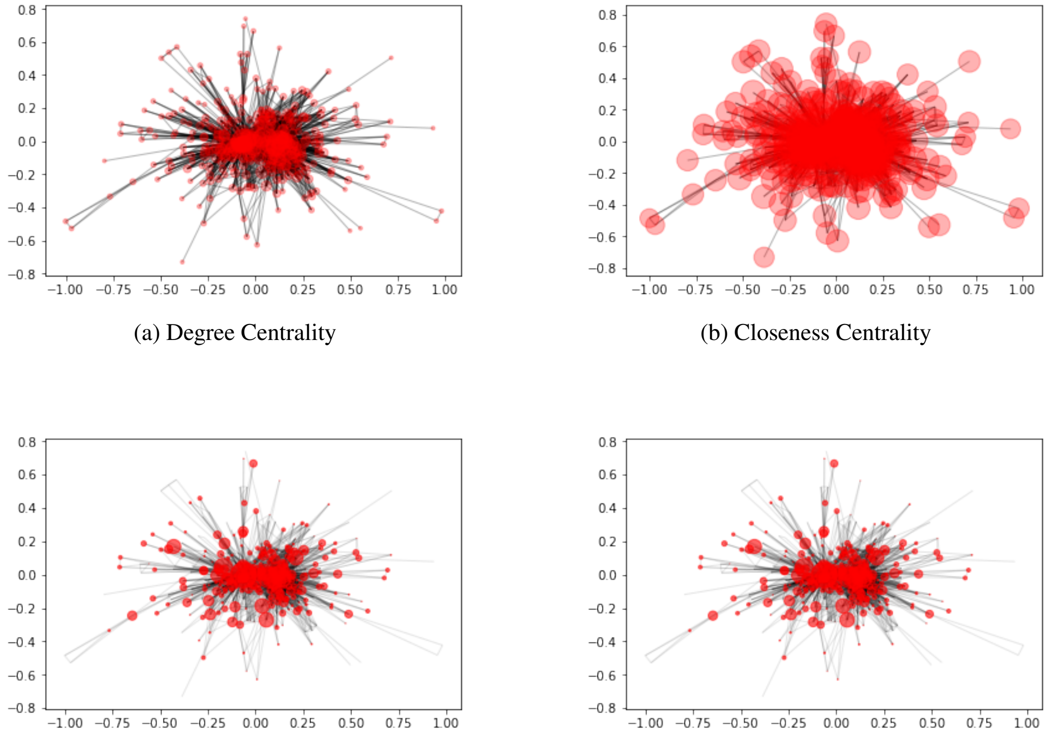


Figure 4: Genre Network-Giant Component

An adjacency matrix was constructed for the song network using the sub-genre feature as the edge data, and it is of the form,

$$A_{genre(i,j)} = \sum_{k=1}^{N_{songs}} I[(genre_i, genre_j)\, \epsilon\, Genre_{song_k}]$$

Various centrality measures were computed on this giant component, and are as indicated in Figure 5. This is based off a study done by Bryan and Wang [2011], who had derived some interesting patterns from sample-based musical network data.



(a) Degree Centrality

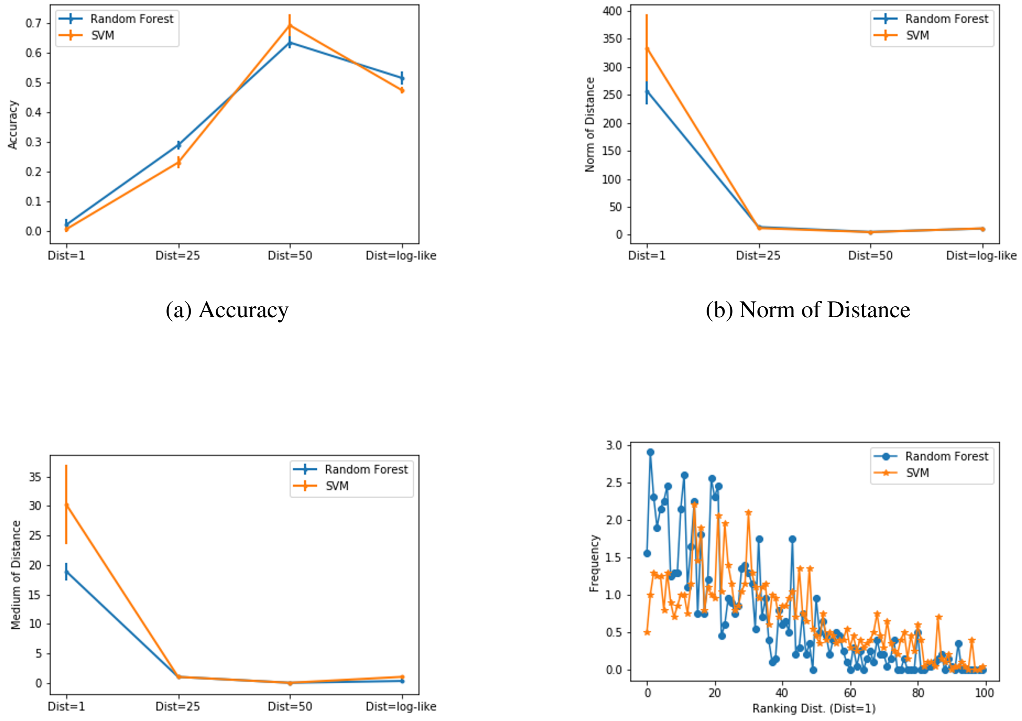
(b) Closeness Centrality

(c) Betweenness Centrality

(d) Eigenvector Centrality

Figure 5: Plots of the giant components with the scaling of the nodes reflecting the different centrality measures

When comparing accuracy metrics, it is again observed that at D=50, SVM marginally performs better than Random Forest in Figure 6. Higher variance is observed at D=1 for SVM when looking at the norm and median of variance. The presence of too many classes(100) explains the variation in performance for SVM, similar to the previous



(a) Accuracy

(b) Norm of Distance

(c) Median of Distance

(d) Histogram of Error Distance for Ranking Dist = 1

Figure 6: Prediction Results with genre centrality measures included

4.3 Graph Construction

We also take a look at the year-by-year trend of the genre network in popularity from 2000-2017. To begin with, we look at the larger genres network trends, which where a set of 6 categories. Songs where the genre was unknown was not counted while plotting the network. Since there was only one genre per song, the network edges were directed to reflect the popularity of each genre in descending order.

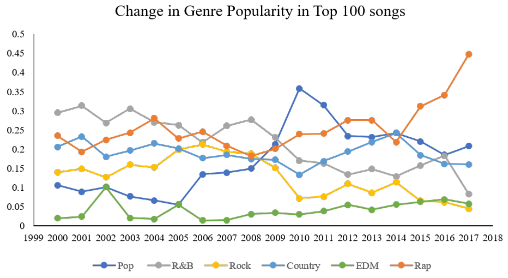
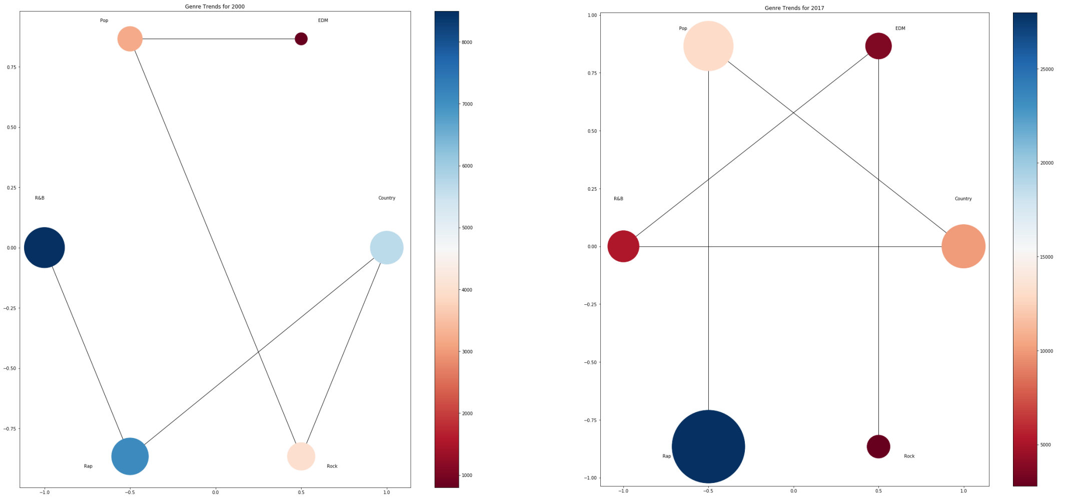


Figure 7: Change in Genre Popularity in Top 100 songs

On observation of Figure 7, we find that pop has been increasing in popularity over the years, with a large spike in 2017, and rap has shown consistent popularity, ranking either 1st or 2nd in yearly popularity rankings.



(a) Genre Network plot-2000

(b) Genre Network plot-2017

Figure 8: Genre Network sample plots

Next, an attempt was made to construct the sub genre network to study in greater detail the evolution of sub genre trends. One great challenge encountered during this process was that out of 391 sub genres present in the dataset, only an average of 80 were present each year, many of them overlapping. Thus, using this network to study trends is not an accurate method, and there would be need to create a more general sub genre class.

4.4 Ranking Training and Testing for Small Networks

Since the prediction results are not improved with genre centrality features, we have a theory: there is a possibility that genre trends and other features affecting the ranking of a song would not have stayed constant through the 18 years considered in this data set. In order to capture those differences effectively during the prediction, a ’small network’ consisting of a subset of the data was used for training and testing. The training and testing sets consists of 2500 and 100 songs, respectively, and a ’sliding window’ shift of 238 songs, in a chronological order, averaging to a total of 21 trials over the whole data set (Figure 9).

In Figure 10, it appears that ’small network’ performs better than training all of the songs that we have. However, according to the important features revealed from classifier, features ’weeks’, ’peak_pos’, and ’last_pos’ play a vital role in prediction, not genre centralities. Therefore, the theory of genre trend could not be a good assumption.

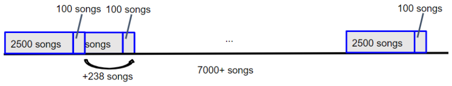
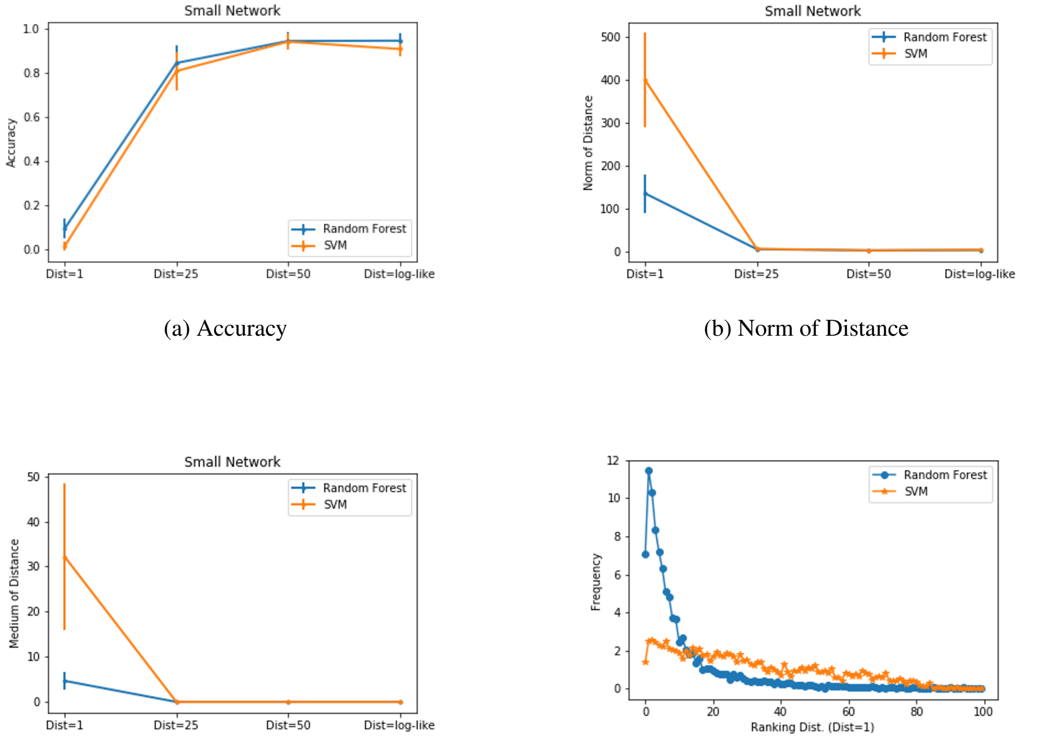


Figure 9: Small Network Generation Model for Genre Network centralities



(a) Accuracy

(b) Norm of Distance

(c) Median of Distance

(d) Histogram of Error Distance for Ranking Dist = 1

Figure 10: Small Network Prediction Results

Because genre centralities barely provide help, we then introduce song network centrality features into the small network prediction paradigm. The plot of the various song network centrality measures are shown in Figure 12. The song network adjacency matrix can be be defined as follows:

$$A_{song(i,j)} = \sum_{k=1}^{N_{genre}} I[(song_i, song_j)\, \epsilon \, song_{genre_k})]$$

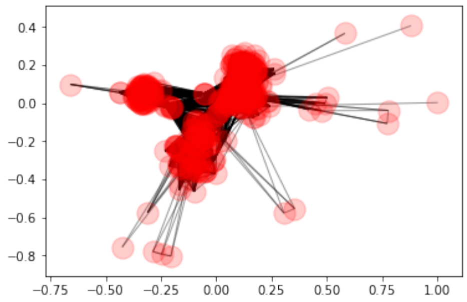
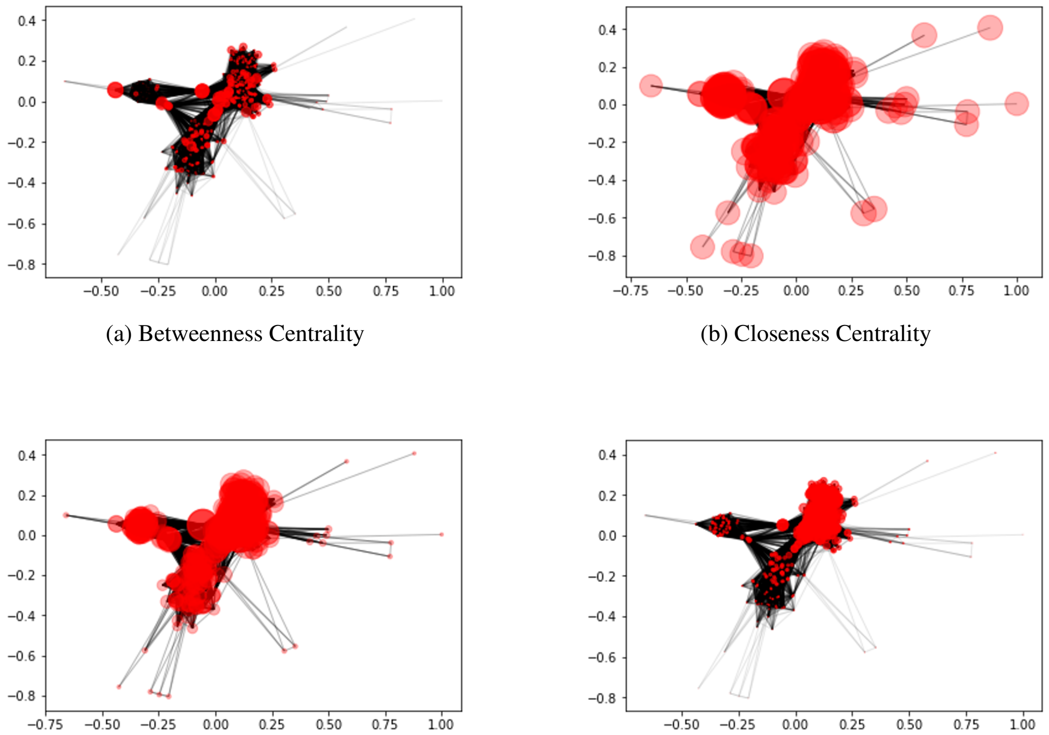


Figure 11: Song Network-Giant Component



(a) Betweenness Centrality

(b) Closeness Centrality

(c) Degree Centrality

(d) Eigenvector Centrality

Figure 12: Song Network Centrality Measures

It is noted that 7000+ songs cause out of memory (OOM) problem in calculating the centrality measures. In addition, it takes a long time to calculate centralities for 2600 songs. In order to account for available processing power without compromising on centrality measures, a much smaller subset of 500 songs was considered. As before, the data was split into training and testing of 400 and 100 songs, respectively. The sliding window in this case was of 137 songs, which added up to a total of 51 trials across the whole data set (Figure 13). The chronological arrangement was preserved in this case as well.

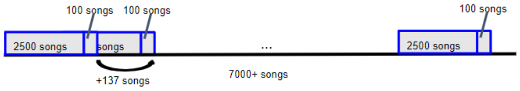


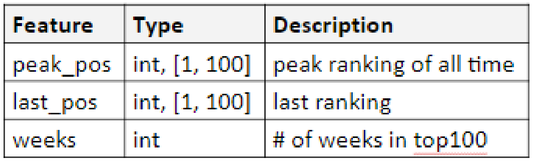
Figure 13: Small Network Generation Model for Song Network centralities

The results are observed to be similar to what is observed with the earlier case of 2500 songs, with a smaller error distance. The song centralities are shown to not substantially improve the prediction accuracy, as without their introduction they were quite optimal.

What is more, we also shuffle the chronological ordered dataset to observe changes in the prediction accuracy, but the results still remain similar.

5 Conclusion

There were three key features identified to have the highest impact in the prediction model, and they are mentioned in the table below.



Feature	Type	Description
peak_pos	int, [1, 100]	peak ranking of all time
last_pos	int, [1, 100]	last ranking
weeks	int	# of weeks in top100

Figure 14: Key Features influencing Prediction model

The attempt is to observe if the inclusion of genre and song network centrality features would improve accuracy in prediction of ranking across all four intervals. From the results, it is seen that some combination of centralities does not help in improving ranking prediction, either the linear combinations of genre centralities, or song centralities.

With the exception of the binary classification case, Random Forest has a higher prediction accuracy as compared to SVM. This can be accounted for by the presence of too many classes with a substantial outlier presence in the feature set which is known to degrade SVM performance. Linear SVM also does not perform well with features that might not be linearly separated. Another point to note is that Random Forest has been known to handle categorical features well, as is in this data set, and is also more stable in handling the feature non-linearity.

With the observation that we have, small network performs better. Also,the shuffled data works as well as chronological small network. Therefore, these results show that genre trend cannot help a lot in out designed experiments.

6 Further Work

Since we mainly take advantage of the linear combination of genre centralities, the important factor could be inherent and found in another combination. Besides, instead of using traditional classifier like Random Forest, SVM, we may use deep learning concept such as long short-term memory (LSTM) to see if the prediction becomes better.

Another factor to keep in mind is the extent of influence of factors other than the musical parameters of the song influencing the ranking. Factors such as virility of the song, tie-ins with other media such as a popular movie or TV show also influence exposure and thus ranking rather than pure airplay. It is also important to keep in mind the increasing influence of social media in the popularity of songs, as ranking bodies such as Billboard also have starting placing increasing importance on music and YouTube stream counts, and mentions on social websites such as Facebook and twitter. This was different from the early 2000's where more weight-age was on radio play and album/single purchases. Thus, considering social media factors in our implementation might improve the accuracy of prediction.

Observe sub-genre network trends would be the next step to studying the evolution of genre classification over the years. The bottleneck of having too many sub classifications could be used as an tool to track rising and falling trends within the larger genre classification, and observe similarities between sub-genres falling under seemingly different genres.

Another interesting result we would like to obtain is to use the network characteristics to determine distances between different genres of music through discovering collaborations between artists across different musical classes. Along with using the ranking as a popularity metric, we can try to infer what genre of music a person is more likely to explore next, given his current musical preferences.

References

Kerstin Bischoff, Claudiu S. Firan, Mihai Georgescu, Wolfgang Nejdl, and Raluca Paiu. Social knowledge-driven music hit prediction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5678 LNAI:43–54, 2009. ISSN 03029743. doi: 10.1007/978-3-642-03348-3_8.

Nj Bryan and Ge Wang. Musical Influence Network Analysis and Rank of Sample-Based Music. *Proceedings of the 12th International Society for Music Information Retrieval Conference*, (Ismir):329–

334, 2011. URL http://www.mirlab.org/conference{_}papers/International{_}Conference/ISMIR2011/papers/OS4-4.pdf.

Vivek Datla and Abhinav Vishnu. Predicting the top and bottom ranks of billboard songs using Machine Learning. 2015. URL http://arxiv.org/abs/1512.01283.

Niklas Donges. The Random Forest Algortihm, 2018. URL https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd.

A A Hagberg, D A Schult, and P J Swart. Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference SciPy2008*, volume 836, pages 11–15, 2008. ISBN 3333333333. URL http://www.osti.gov/energycitations/product.biblio.jsp?osti{_}id=960616.

Yizhao Ni and R Santos-Rodriguez. Hit song science once again a science. *4th International Workshop . . .*, pages 2–3Ni, Y., & Santos–Rodriguez, R. (2011). Hit song, 2011. URL http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.412.9732{&}rep=rep1{&}type=pdf.

Scikit-learn.org. Support Vector Machine. URL https://scikit-learn.org/stable/modules/svm.html.

Spotify.com. Get Audio Features for a Track. URL https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/.

Michael Tauberg. Billboard Hot-100 Songs 2000-2018 w/ Spotify Data + Lyrics, 2018. URL https://data.world/typhon/billboard-hot-100-songs-2000-2018-w-spotify-data-lyrics.