**Human, Mathematical, and Machine Learning Understandings of “Influentiality” in Music**

As music evolves over time, artists find new, interesting ways to express themselves, their era, and their ideas. The presence of commonality among era and the presence of genre point to the idea that artists’ productions are affected by their environment and that these environments are largely more similar within an era than across eras, independent of the individual experiences that artists have. For artists, a part of this environment which they seem to think affects them the most is the work of other artists. This is the notion of “influence” which is studied here. Influence is difficult to pin down into precise terminology for various reasons. Firstly, it takes time to notice influence at all; when an artist releases influential music, no one truly knows its influence until years later when other artists’ music appears to have been influenced by the work. New York rapper A$AP Rocky materializes this idea into his own understanding, saying, “I feel like everything I do in the hip-hop world has an influence. People don't really notice what I did until somebody else does it.” Secondly, influence is difficult to define and operationalize. Is one song influenced by another if and only if there is some measurable similarity between the two songs? Or is it true that one song could be influenced by another simply because an artists’ state was affected by listening to a song? People tend to label artists as “influential” because of some innate feeling that an artist was “ahead of their time” or had great “impact” on the next generation of artists. However, these feelings are not ones that we can easily evidence or quantify. This gap between human understanding of what it means for songs and artists to be influential versus the true, evidenced influence that we can measure between songs leads to an interesting question to examine: How does human understanding of musical influence map onto mathematical and computational models of influence and what can we learn about the human definition of influence from computational and mathematical models?

One computational and psychological study on social influence describes influence as, “individuals’ tendency to conform to the beliefs and attitudes of others,” (Zaki et al. 2011). This applies to an intuitive definition of songs and artists quite well, simply by replacing beliefs and attitudes to musical content; we might expect that a song which has been heavily influenced by other songs will conform to the themes and techniques of that song very strongly, and songs which are highly influential will in turn have many other songs which conform to the structure of its musical content. There is a gap in the literature in identifying features or psychological phenomena which make certain songs feel influential, which makes our method in using computational systems to understand these phenomena even more interesting. To identify ways to use information science to uncover the meaning of influential songs and artists, we turn to the literature for musical influence and more broadly, computation of influence networks.

In general, influence is computed after measuring similarity. There are various ways for computing influence, but at its core, in contexts similar to our own, to understand the way that works or artists influence each other, we have to understand the way that they are similar. In some other contexts such as social network relationships, we can directly view the way that the presence of one event incites the creation of another event, and measure influence in a more direct way (Zaki et al 2011). However, generally, for a topic such as music where it’s unclear how one piece or artist directly affects the creation of a new piece, especially across a large dataset, similarity measurements are the basis for measuring influence (Collins 2010). One differing and valid approach for understanding influence in music was done by Bryan and Wang, where they used sampling in music as an indication of influence (2011). For a broader corpus of music which does not constrain the works to sampled pieces, however, similarity is a valid and repeated basis from the literature (Shalit et. al 2013).

Given that we are most trained in using the Spotify API, it’s useful to turn to a musical similarity and influence study which stems from features in the Spotify API. The work done by Cui, Jia, and Gu is extremely applicable for measuring influence and similarity in music, and was largely used as a neighboring framework for the work done in this paper (2021). Cui, Jia, and Gu first constructed a music similarity evaluation model for assessing similarity. In general, they use a combination of standardization, principal component analysis on Spotify features, and Euclidean distance for their similarity measurement. Although it is difficult to measure the validity of similarity for something such as music, their work is mathematically backed (Cui et. al 2021). Furthermore, they went on to calculate “influence values.” Their influence values were based on the similarity they calculated and was time-dependent in order to determine the directionality of the influence. They went on to use the DEMATEL method to gather a comprehensive influence matrix of artist to artist influence based on these “influence values,” (Cui et. al 2021). The DEMATEL method uses graph theory to systematically analyze the comprehensive analysis between nodes. To illustrate, if Node A affects Node B, and Node B affects Node C, the DEMATEL method gathers that, to some degree, Node A did not only influence Node B, but also Node C (Zou et. al 2022). The work done by Cui, Jia, and Gu gives a feasible framework for a mathematical model to understand musical influence. Using the Cui framework can help us understand the degree to which mathematical understandings of influence overlap with a human understanding of influence. In particular, if we find songs and artists who will be considered highly influential by both this framework and by humans, then we may assert some overlap in influence ideation from both a human and mathematical perspective.

More than just identifying the existence of similarity between human and mathematical definitions of influence, we also want to examine what the human definition of musical influence is, and whether we can create computational models which mimic this human definition. By creating a computational model to mimic musical influence, we will be able to answer the latter part of this question, and we will be able to extract features which could contribute to our understanding of the human idea of influence.

In the following section, **Method**, we justify and explain two separate processes that aim to uncover the overlap between human and mathematical ideas of influence, and separately, how well computational models are able to mimic human ideas of influence and what these models can teach us about influence. We firstly predict that there will be some overlap between human and mathematical ideas of influence, namely that songs and artists which are statistically significantly influential based on the mathematical model will also fall in the human-made category of influential songs and artists. Secondly, we predict that a computational model will be able to discern the difference between songs which are deemed influential and those which are not, and that this model will use features that are related to similarity from the first process to make these decisions. By carrying out an effective methodology and analyzing these results, we will be able to compare and contrast human and mathematical understandings of influence.

**Method**

1. *Corpus Construction*

To make statistical claims about music, we need to create a collection of songs which will help us carry out our processes. Based on the research and the general processes that we will carry out, the corpus must represent a collection of artists and songs for each of these artists. Some of these artists will be deemed *influential* and some will be *random*. By comparing these two groups, we will be able to measure the accuracy of a mathematical model with respect to the human definition of influence. It follows that we create a list of artists and randomly select the same number of songs for each of these artists. The population from which to select these artists from is Popular Artists from the 2010s. The specific era for this population was simply selected because of our interest in artists from the 2010s, but it was needed to select artists from a common era to not need to include the complicated time-dependent measure in the influence evaluation. By selecting a specific era, the hope is that artists in the era would likely influence each other, rather than unidirectionally, as this was a challenge presented in similar studies (Shalit et. al 2013). Popular artists was a term broadly selected so that there would be sufficient and standardizable data. Creating the groups of *influential* and *random* was done based on these constraints. For *influential*, there are two subgroups: 1) CNN and 2) Author. The CNN subgroup was created based on an article by Leah Asmelash and Scottie Andrew of CNN, who wrote about 10 artists who transformed music in the 2010s (2019). The author subgroup was created on our opinion of who were 10 influential artists of the 2010s. While this is not a comprehensive definition of human influence, it is still certainly a human-defined group of influential artists, and notably, although the subgroups were created independently, 4 of the 10 artists in each subgroup was present in the other, which lends some, albeit weak, credence to the reliability of the *influential* label. For the *random* group, artists were selected from the Billboard Top 200 of the 2010s at random, so that they would still fit the population of Popular Artists from the 2010s. In the corpus, all of the available numeric Spotify features are listed. View Appendix A for the code which led to the corpus.

1. Part A, Comprehensive Impact Matrix for Influence

In this section, we use the framework outlined by Cui, Jia, and Gu, and the DEMATEL method to create a matrix of influence from song to song, and then extend this matrix to determine whether the statistically significantly influential songs and artists belong to the *Influential* group at a statistically significant rate.

After standardizing the data and computing a principal component analysis, we use the principal component vectors which can account for more than 80% of the data. At this point, we note the most weighted features among the principal components for comparison in Part B. We then compute the distance between songs, using the principal components as a basis for operationalizing the songs. Then a distance matrix among songs is constructed using Euler distance. Finally, the DEMATEL matrix is constructed, using the equation , where T is the DEMATEL matrix, N is the normalized distance matrix, and I is the identity matrix (Cui et. al 2021). For each song and artist, we sum the corresponding row(s) of T, to find its total influence. We note the difference among groups in influence. Then, as we are interested in the most influential artists and songs when we discuss influence, we find the statistically significant influential artists and songs and compare these to the original groupings to identify any overlap in mathematical influence and human-defined influence.

1. Part B, Classifying Based on Human Influence Tags

In this section we use the groupings of *Influence* to train some classifiers to examine whether it is possible to model the human idea of musical influence, and to use the result of these classifiers to examine what this idea consists of.

To train the classifiers, we separate the data into an 80-20 split of training and testing data. We then train a model for various classifiers and construct a confusion matrix to assess the accuracy of these models. We pick the best model and measure the importance of each of the features.

**Results**

*Part I, Comprehensive Impact Matrix for Influence*

|  |  |
| --- | --- |
| Feature | Total Weight |
| Danceability | 1.653851 |
| Energy | 1.162440 |
| Key | 2.037981 |
| Loudness | 1.036513 |
| Mode | 1.871516 |
| Speechiness | 1.608342 |
| Acousticness | 1.245609 |
| Instrumentalness | 1.920502 |
| Liveness | 2.020182 |
| Valence | 1.458392 |
| Tempo | 2.287712 |
| Time Signature | 2.317500 |
| Duration | 2.226715 |

*Table I. Total (absolute value) Principal Component Feature Weights for Corpus*

We note that the most weighted features are Duration, Tempo, and Time Signature.

A picture containing text, diagram, screenshot, display

Description automatically generated*Figure I.* Comparison of the groups in influence values.

There appears to be no notable difference among the three groups.

|  |  |  |
| --- | --- | --- |
|  | Proportion of significantly influential datapoints agree with *influential* classification | P-value |
| Songs | 0.515 | 0.003826\*\* |
| Artists | 0.636 | 0.1641 |

*Table 2.* Agreement between mathematical influence model and human influence tags. The first column gives the proportion of datapoints which were statistically significantly influential that agreed with the human classification. The P-value describes the likelihood that the difference between this proportion and random classification is due to chance.

We note that for both songs and artists, the proportion of significantly influential datapoints which were classified “correctly” according to human classification are both greater than the proportion of 0.4444 which represents the random chance of influence in this corpus. For songs, but not artists, this difference is significant.

A picture containing text, screenshot, map, diagram

Description automatically generated*Part B, Classifying Based on Human Influence Tags*

*Figure II.* Visualization of Classifying Influential vs. Random based on two principal components.

We note that it appears difficult to classify the data using linear models.

A picture containing text, screenshot, diagram, number

Description automatically generated

*Figure III.* Accuracies of the trained models.

Random forest appears to have done the best and is the only model with a Kappa value of 0.43. It had an accuracy rate of 0.713, a positive predictive value of 0.7024 and a negative predictive value of 0.7197 for the positive class of *influential*.

A picture containing text, screenshot, line, parallel

Description automatically generated

*Figure IV.* Importance of features in the random forest model.

We note that the most important features are speechiness, duration, and loudness. Besides duration, this is different than the feature importance found in Part I.

**Discussion**

In this study we attempt to answer the question: How does human understanding of musical influence map onto mathematical and computational models of influence, and what can we learn about the human definition of influence from computational and mathematical models?

To understand how human understanding of influence maps onto mathematical models of influence we used the information and DEMATEL matrix framework to make comparisons to human influence. To understand what we can learn from the human definition of influence from computational and mathematical models, we built classifiers which aimed to mimic human classification and examined the important features of these classifiers.

First, the mathematical model to compute influence did not appear to have any robust differences among the human-made groups for all datapoints, as seen in Figure I. It may be that there are not significant similarities between human understanding of influence and this mathematical model across all datapoints. However, it’s also possible this null result was caused by one of the limitations of this study, especially considering the significant results which were obtained in other parts of the study. These limitations were present throughout the study presented here and understanding these limitations may help further studies to improve.

The categorization of artists and songs for human-defined influence was simply created from one CNN source and our own opinion of influential artists. It was difficult to find a definitive list of influential artists that was averaged across many people, and in part, that was the motivation for the work here. In addition, another important limitation is that we used only Spotify features to construct our models. Spotify features are already an amalgamation of measurements, and it is hard to pinpoint exactly what these features mean or how to effectively use them in a model. Finally, one important limitation for our mathematical model in Part I was that we did not use a time-dependent model. Although in doing so we avoided some of the more difficult challenges associated with measuring influence, measures of influence were exclusively bidirectional, meaning that an artist who was found to be extremely influential might have in fact been extremely influenced by other artists. Certainly, other limitations in this study are present.

From the null results achieved across all songs, we examined the songs and artists which were considered significantly influential. This is a valid and applicable focus because on the whole, when we discuss influence, especially from the human perspective, we are more concerned with works and artists that are highly influential rather than works which are moderately influential. We found that songs which were found significantly influential were also considered influential by the human classification at a very significant rate (Table 1). Conversely, we found that significantly influential artists were not found to be influential by the human classification at a significant rate. However, with only 36 total artists in the corpus, it would be quite difficult to find a significant result. Even with a difference of proportions of 0.4444 to 0.6364 in finding influence, no significance was found due to the low sample size. In the future, corpora should be constructed with artist sampling in mind. Overall, the model of finding significantly influential songs appears to have significant overlap in human understanding of influence, and we suspect that with a larger artist sample size some significant overlap in mathematical and human understanding would also exist. From this result, we find that there is some basis of similarity in musical influence, but this basis, at least for the mathematical model that we selected, should be constrained to those songs which are most influential, or even outliers in influence.

To further understand what mathematical and computational models inform us about human understanding of influence, we built classifiers to distinguish between the human-made classifications of influence. Among the classifiers, the random forest classifier worked the best, with a Kappa value of 0.43, an accuracy rate of 0.713, a positive predictive value of 0.7024 and a negative predictive value of 0.7197 for the positive class of *influential*. This implies that the classifier was about equally as good at predicting influential songs as influential as it was at predicting non-influential songs as non-influential. We had predicted that it would be better at predicting non-influential songs as non-influential due to the larger sample size, and we can interpret this result as encouraging as it supports that the model was able to identify influential songs as influential quite well, which we are more concerned about. The Kappa value of 0.43 can be interpreted as a moderate agreement, which supports our hypothesis that the classifier was able to mimic the human classification (Sun 2011). Based on the success of the classifier, we claim that the computational model shows promise in uncovering what a human definition of influence means. Stripping back layers of the classifier, however, is rather difficult. Our attempt to do so involved extracting the most important features from the classifier. We found that Speechiness, Duration, and Loudness. Interestingly, this was different than what we found in Part A, where we found that the most important features were Duration, Tempo, and Time Signature. We hypothesize that in Part A, the influence measurement was based largely on similarity, and the time-based measurements probably gave a good estimate of similarity between songs. In Part B the features of Speechiness and Loudness are difficult to unpack. It may be the case that these attributes are important to the human definition of influence. However, it is perhaps more accurate to state that the Spotify features do not map directly onto human feeling or logic, and instead they simply estimate the definition of human influence in combination with each other. In future studies, it may be helpful to use more basic features which can map more clearly onto human thought.

In conclusion, in answering our research question, we found that despite the limitations of our work, mathematical models shared interesting overlap with human definition of influence. Mathematical models agreed with human definition to some extent, suggesting that we might be able to unpack mathematical models in the future to extract what song and artist features lead to an influential appearance. Similarity in song appeared to be a valid basis for measuring influence. This type of approach in comparing human ideation of music with mathematical modeling of music may be helpful in uncovering how humans perceive music.

References

Asmelash, L., & Andrew, S. (2019, December 31). *The 10 artists who transformed music this decade*. CNN. https://www.cnn.com/2019/12/31/entertainment/2010s-music-artists-define-decade-trnd/index.html

Bryan, N., & Wang, G. (2011). MUSICAL INFLUENCE NETWORK ANALYSIS AND RANK OF SAMPLE-BASED MUSIC. *12th International Society for Music Information Retrieval Conference*.

Collins, N. Computational analysis of musical influence: A musicological case study using mir tools. In Proceedings of the International Symposium on Music Information Retrieval (ISMIR), 2010.

Cui, C., Jia, Z., & Gu, K. (2021). Research on the information entropy using processing square matrix method based on similarities evaluation model. *Journal of Physics: Conference Series*, *1952*(4), 042059. https://doi.org/10.1088/1742-6596/1952/4/042059

Shalit, U., Weinshall, D., & Chechik, G. (2013). Modeling Musical Influence with Topic Models. *MLR Press*.

Weiss, J. (2018, June 25). *Q&A: A$AP Rocky on sudden fame, relating to Kurt Cobain and Rihanna*. Rolling Stone. https://www.rollingstone.com/music/music-news/qa-aap-rocky-on-sudden-fame-relating-to-kurt-cobain-and-rihanna-202097/?sub\_action=logged\_in

Zaki, J., Schirmer, J., & Mitchell, J. P. (2011). Social influence modulates the neural computation of value. *Psychological Science*, *22*(7), 894–900. https://doi.org/10.1177/0956797611411057

Zou, T., Jin, Y., Chuang, Y.-C., Chien, C.-W., & Tung, T.-H. (2022). The DEMATEL method explores the interdependent relationship structure and weights for diagnosis-related groups system. *Frontiers in Public Health*, *10*. https://doi.org/10.3389/fpubh.2022.872434

**Appendix A**

library(spotifyr) library(tidyverse)

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 -- ## v dplyr 1.1.1 v readr 2.1.4

## v forcats 1.0.0 v stringr 1.5.0

## v ggplot2 3.4.2 v tibble 3.2.1

## v lubridate 1.9.2 v tidyr 1.3.0

## v purrr 1.0.1

## Conflicts tidyverse\_conflicts()

## x dplyr::filter() masks stats::filter() ## x dplyr::lag() masks stats::lag()

## i Use the conflicted package [(<http://conflicted.r-lib.org/>)](http://conflicted.r-lib.org/) to force all conflicts to become error

library(dplyr) remotes::install\_github('jaburgoyne/compmus')

## Skipping install of 'compmus' from a github remote, the SHA1 (979921b7) has not changed since last ## Use `force = TRUE` to force installation

library(compmus) library(corrplot)

## corrplot 0.92 loaded

library(gridExtra)

##

## Attaching package: 'gridExtra' ##

## The following object is masked from 'package:dplyr': ##

## combine

Sys.setenv(SPOTIFY\_CLIENT\_ID = 'a6c987bbbdfc420a902b3825682657df') Sys.setenv(SPOTIFY\_CLIENT\_SECRET = '7b180fbab7c74d2d94802c8deeb5fb24') access\_token <- get\_spotify\_access\_token()

# R Markdown

In this assignment, I aim to provide the coprus that I will be using for my final project, and some descriptive statistics about this corpus. In my project, I aim to identify what factors lead to “influentiality.” In doing so, I will have to define what influentiality means, and find factors which may suggest influentiality. From my literature review, I plan on attempting to create an influence association matrix, where the weights of the matrix are based on “Influence Values”. These “Influence Values” between artists are calculated from the factors of 1. the influence between their genres, 2. the influence of their active years, 3. the year gap between two artists

In order to obtain evaluation indicators for these factors, or any other factors that may be more simple for this application, we will need to create some music simularity evaluation model, which from the liter- ature, appears to use principal component anaylsis based on features, and there has been success in using the 11 Spotify features of acousticness, instrumentalness, liveness, speechiness, duration\_ms, popularity, danceability, energy, valence, tempo, and loudness.

All of that to say that I really want to compare artists. Thus my corpus should be a stratified sample of artists music. To gather a sample to make reasonable statistical conclusions, I will pick artists from whom I want to study, and then gather a random sample of 30 of their songs.

To make claims about influentiality, I want to have 2 main groups of artists. Those considered influential, and a random group of artists. Within the influential group I will pick 10 artists whom I consider influential and 10 artists who are considered influential from the literature. For the random group of artists, I would still like them to be popular artists so that they have the potential for influentiality on each other, and so that the interactions and similarities among random-“influential” artists are as statistically likely to be as existent as “influential”-“influential” similarities or random-random similarities.

Importantly, these artists should all be from around the same era so that they have the potential to similarly impact each others’ music. Because of my knowledge of music, I will restrict artists to being present in the 2010’s.

Randomly Selecting Artists from the 2010s. From this Billboard source (billboard.com/charts/decade- end/billboard-200) of the top 200 songs of the 2010s, I used a random number generator to pick a song and a corresponding artist.

print(sample(1:200, 20))

## [1] 165 128 51 2 22 138 56 7 123 65 116 118 175 159 156 85 119 4 23

## [20] 9

The numbers for the first run (these numbers will change on compile) were:

Show in New Window [1] 101 116 107 47 67 143 59 141 7 172 120 157 115 14 73 189 117 5 46 63

Respectively, these map to these artists.

Rihanna, Eric Church, Zac Brown Band, Sam Hunt, Meghan Trainor, Robin Thicke, Macklemore, Brantley Gilbert, Imagine Dragons, DaBaby, Hozier, The Band Perry, Ke$ha, Sam Smith, Carrie Underwood, Sia, Lana Del Ray, Post Malone, XXXTENTACION, fun.

From my own opinion, here are 10 artists whom I consider influential to the 2010s.

Bruno Mars, Drake, Taylor Swift, Ed Sheeran, Lil Wayne, Kanye West, One Direction, Lady Gaga, Pharell Williams, Daft Punk

Here’s CNN’s opinion on influential artists:

Beyonce, Kendrick Lamar, Frank Ocean, Lady Gaga, Drake, Metro Boomin, Taylor Swift, Solange, BTS, Kanye West

Some of these are repeats, in my own list of influential artists, but I don’t think this should matter too much when constructing the overall corpus. We may leave a note to address this later.

Now let’s construct a random sample of 30 songs of each of these artists’ music to create our corpus.

# Including Plots

get\_track\_sample <-**function**(artists){ df <- data.frame()

**for** (artist **in** artists) {

tracks <- get\_artist\_audio\_features(artist) sampled\_tracks <- sample\_n(tracks, size = 30) sampled\_tracks$artist <- artist

df <- bind\_rows(df, sampled\_tracks)

}

return (df)

}

random\_artists\_sample <- get\_track\_sample(c("Rihanna", "Eric Church", "Zac Brown Band", "Sam Hunt", "Me my\_artists\_sample <- get\_track\_sample(c("Bruno Mars", "Ed Sheeran", "Lil Wayne", "One Direction", "Phar cnn\_unique\_artists\_sample<-get\_track\_sample(c("Beyonce", "Kendrick Lamar", "Frank Ocean", "Metro Boomin shared\_artists\_sample<-get\_track\_sample(c("Drake", "Taylor Swift", "Kanye West", "Lady Gaga"))

random\_artists\_sample$label <- "random" my\_artists\_sample$label <- "thomas" cnn\_unique\_artists\_sample$label <- "cnn" shared\_artists\_sample$label <- "shared"

get\_track\_sample <-**function**(artists){ df <- data.frame()

**for** (artist **in** artists) {

tracks <- get\_artist\_audio\_features(artist) sampled\_tracks <- sample\_n(tracks, size = 30) sampled\_tracks$artist <- artist

df <- bind\_rows(df, sampled\_tracks)

}

return (df)

}

Now we can view the whole corpus.

corpus <- rbind(random\_artists\_sample, my\_artists\_sample, cnn\_unique\_artists\_sample, shared\_artists\_sam save(corpus, file = "corpus.RData")

print(min(corpus$album\_release\_year))

## [1] 1997

print(max(corpus$album\_release\_year))

## [1] 2023

We can see that our corpus has 1080 songs, meaning that there are 36 artists present. 20 artists are random from the Billboard chart, 6 are of my own influentiality intution, 6 are of CNN’s influential ranking, and 4 artists CNN and I agreed on. We have a range of 1997-2023 for music, which makes sense since we are looking at artists present in the 2010s.

By comparing how well we are able to mathematically classify influentiality to our human opinions of influentiality we will be able to assess what we mean and what we should mean when we say an artist is influential. Other descriptive statistics will be important when we identify what features are important for gauging similarity via Principal Component.

**Appendix B**

---

title: "Final Project" author: "Thomas Moh" date: "5/15/2023" output:

pdf\_document: default word\_document: default

---

```{r setup, include=FALSE} knitr::opts\_chunk$set(echo = TRUE) library(devtools) install\_github("vqv/ggbiplot") library(ggbiplot) library(spotifyr) library(tidyverse)

library(dplyr) remotes::install\_github('jaburgoyne/compmus') library(compmus)

library(corrplot) library(gridExtra)

Sys.setenv(SPOTIFY\_CLIENT\_ID = 'a6c987bbbdfc420a902b3825682657df') Sys.setenv(SPOTIFY\_CLIENT\_SECRET = '7b180fbab7c74d2d94802c8deeb5fb24') access\_token <- get\_spotify\_access\_token()

```

## R Markdown

We're eventually going to measure the similarity between each song, measure how influential each song was, and then test if the most influential songs belong to influential artists, test if we can classify a song as influential, and test if artists are significantly more influential than each other, and see if we can classify an artist as influential or not.

Loading in corpus

```{r cars} load("corpus.RData")

```

```{r creating PCA for similarity} columns\_wanted <- c(9:19, 22, 26) label <- corpus[,41]

pca\_data <- corpus[, columns\_wanted]

standardized\_pca\_data <- pca\_data %>% mutate\_all(~(scale(.) %>% as.vector)) pieces.pca <- prcomp(standardized\_pca\_data, center = TRUE, scale. = TRUE) plot(pieces.pca, type = "l", main="Principal Components Analysis") print(sum(pieces.pca$sdev[1:8]^2)/sum((pieces.pca$sdev)^2))

print("81% of the variance is accounted for by the first 8 PC. We will proceed with these PC's")

relevant\_pca\_matrix <- pieces.pca$rotation[,1:8] total\_pc\_weights <- rowSums(abs(relevant\_pca\_matrix)) print(total\_pc\_weights)

```

```{r constructing song similarity matrix}

song\_similarity <- matrix(nrow = 1080, ncol = 1080)

distance\_matrix <- dist(as.matrix(pca\_data) %\*% (relevant\_pca\_matrix)) distance\_matrix <- as.matrix(distance\_matrix)

labeled\_distance\_matrix <- cbind(distance\_matrix, corpus$artist) labeled\_distance\_matrix <- rbind(distance\_matrix, corpus$artist)

```

## Including Plots

You can also embed plots, for example:

```{r creating DEMATEL matrix, echo=FALSE} normalized\_distance <- scale(distance\_matrix) I <- diag(nrow(normalized\_distance))

dematel\_matrix <- -1\*(normalized\_distance \* t(I-normalized\_distance))

```

```{r creating influentiality vectors} song\_influences <- rowSums(dematel\_matrix)

artist\_influences <- tapply(song\_influences, rep(1:(length(song\_influences)/ 30), each=30), sum)

song\_influences <- cbind(corpus$track\_name, song\_influences) song\_influences <- cbind(corpus$artist\_name, song\_influences) song\_influences <- cbind(corpus$label, song\_influences) song\_influences <- as.data.frame(song\_influences)

artist\_influences <- cbind(unique(corpus$artist\_name), artist\_influences) artist\_influences <- as.data.frame(artist\_influences)

merged <- corpus[match(artist\_influences$V1, corpus$artist\_name),]

merged$artist\_influences <- artist\_influences$artist\_influences artist\_influences <- merged[, c(1,41,42)]

artist\_influences <- as.data.frame(artist\_influences)

```

```{r graphing}

# Create a matrix with labels and song\_influences columns # Subset the matrix for each label

cnn\_data <- as.numeric(song\_influences[song\_influences$V1 == "cnn", "song\_influences"])

thomas\_data <- as.numeric(song\_influences[song\_influences$V1 %in% c("thomas", "both"), "song\_influences"])

random\_data <- as.numeric(song\_influences[song\_influences$V1 %in% c("random", "both"), "song\_influences"])

# Create a list of data for each label

data\_list <- list(cnn = cnn\_data, author = thomas\_data, random = random\_data)

# Create the box and whisker plot

boxplot(data\_list, main = "Box and Whisker Plot", xlab = "Labels", ylab = "Song Influences")

```

```{r statistical t-test for songs}

sample\_sd <- (sd(as.numeric(song\_influences$song\_influences))) sample\_mean <- (mean(as.numeric(song\_influences$song\_influences))) song\_influences$song\_influences <- as.numeric((song\_influences$song\_influences))

alpha <- 0.05

sample\_size <- 1080

# Calculate critical value for z-test z\_critical <- qnorm(1 - alpha)

# Calculate critical value for t-test

t\_critical <- qt(1 - alpha, df = sample\_size - 1)

# Calculate the standard error

standard\_error <- sample\_sd / sqrt(sample\_size) value\_t\_test <- sample\_mean + t\_critical \* standard\_error

print(value\_t\_test)

significant\_song\_influences <- subset(song\_influences, as.numeric(song\_influences) > value\_t\_test) significant\_artist\_counts <- table(significant\_song\_influences$V1) predicted\_influential\_proportion <- significant\_artist\_counts[1]

+significant\_artist\_counts[3]+significant\_artist\_counts[4] predicted\_influential\_proportion <- predicted\_influential\_proportion/ (predicted\_influential\_proportion + significant\_artist\_counts[2]) print(predicted\_influential\_proportion)

original\_proportion <- 480/1080

n\_original <- 1080 # Total number of songs in the original dataset prop\_original <- 480 / n\_original # Proportion of influential songs in the original dataset

n\_predicted <- sum(significant\_artist\_counts) # Total number of predictions made by the algorithm

prop\_predicted <- predicted\_influential\_proportion # Proportion of influential songs predicted by the algorithm

# Perform the one-sample proportion test

result <- prop.test(x = n\_predicted \* prop\_predicted, n = n\_predicted, p = prop\_original, alternative = "greater")

# Print the test result print(result)

```

From this result we find that the influential information matrix was able to identify statistically significant influential songs at a statistically significant rate (p = 0.003826).

That is, songs which were significantly "influential" from the Dematel Matrix were also considered influential by either CNN or me at a significant rate.

```{r statistical t-test for artists}

sample\_sd <- (sd(as.numeric(artist\_influences$artist\_influences))) sample\_mean <- (mean(as.numeric(artist\_influences$artist\_influences))) artist\_influences$artist\_influences <- as.numeric(artist\_influences$artist\_influences)

alpha <- 0.05

sample\_size <- 36

# Calculate critical value for z-test z\_critical <- qnorm(1 - alpha)

# Calculate critical value for t-test

t\_critical <- qt(1 - alpha, df = sample\_size - 1) # Calculate the standard error

standard\_error <- sample\_sd / sqrt(sample\_size) artist\_value\_t\_test <- sample\_mean + t\_critical \* standard\_error

print(value\_t\_test) significant\_artist\_influences <-

artist\_influences[artist\_influences$artist\_influences > artist\_value\_t\_test, ] significant\_artist\_counts <- table(significant\_artist\_influences$label) print(significant\_artist\_counts)

predicted\_influential\_proportion <- significant\_artist\_counts[1]

+significant\_artist\_counts[3]+significant\_artist\_counts[4] predicted\_influential\_proportion <- predicted\_influential\_proportion/ (predicted\_influential\_proportion + significant\_artist\_counts[2]) print(predicted\_influential\_proportion)

original\_proportion <- 16/36

n\_original <- 36 # Total number of songs in the original dataset prop\_original <- 16 / n\_original # Proportion of influential songs in the original dataset

n\_predicted <- sum(significant\_artist\_counts) # Total number of predictions made by the algorithm

prop\_predicted <- predicted\_influential\_proportion # Proportion of influential songs predicted by the algorithm

# Perform the one-sample proportion test

result <- prop.test(x = n\_predicted \* prop\_predicted, n = n\_predicted, p = prop\_original, alternative = "greater")

# Print the test result print(result)

```

Here there is no significance for the most significantly influential artists being classified as influential "correctly" but we have a very small sample size. The proportion of artists was 0.6363 as opposed to 0.4444 but perhaps due to the small sample size we are not able to infer significance.

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

```{r Plotting separation} label\_combined <- label

label\_combined[label\_combined %in% c("thomas", "cnn", "shared")] <- "influential"

g <- ggbiplot(pieces.pca, obs.scale = 1, var.scale = 1, groups = label\_combined, ellipse = TRUE, circle = TRUE)

print(g)

g <- g + scale\_color\_discrete(name = '')

g <- g + theme(legend.direction = 'horizontal',

legend.position = 'top') + theme\_bw()

```

```{r echo=FALSE}

classify\_data <- standardized\_pca\_data classify\_data$label <- label\_combined library(caret)

model\_evaluation <- function(method){

Train <- createDataPartition(classify\_data$label, p=0.8, list=FALSE) training <- classify\_data[ Train, ]

testing <- classify\_data[ -Train, ] mod\_fit <- train(label ~ .,

data=training, method=method) pred <- predict(mod\_fit, newdata=testing)

accuracy <- table(pred, testing[,"label"]) sum(diag(accuracy))/sum(accuracy) testing$label <- as.factor(testing$label) confusionMatrix(data=pred, testing$label)

}

set.seed(1234)

control <- trainControl(method="repeatedcv", number=10, repeats=3)

# train logistic regression

modelglm <- train(label ~ ., data=classify\_data, method="glm", trControl=control)

# train knn

modelknn <- train(label ~ ., data=classify\_data, method="kknn", trControl=control)

# train nnet

modelnnet <- train(label ~ ., data=classify\_data, method="nnet", trControl=control)

# train the LVQ model

modelLvq <- train(label ~ ., data=classify\_data, method="lvq", trControl=control)

# train the GBM model

modelGbm <- train(label ~ ., data=classify\_data, method="gbm", trControl=control)

# train the SVM model

modelSvm <- train(label ~ ., data=classify\_data, method="svmRadial", trControl=control)

# train the random forest

randomforest <- train(label ~ ., data=classify\_data, method="rf", trControl=control)

```

```{r assessing machine learning level accuracy} results <- resamples(list(LVQ=modelLvq, GBM=modelGbm,

SVM=modelSvm,knn=modelknn, nnet=modelnnet, glm=modelglm, rf=randomforest))

```

```{r continued} bwplot(results) model\_evaluation("rf")

```

```{r which features are important?} plot(varImp(randomforest))

```