IST 736 Final Project

Jeffrey Chao, Eric Gillis, Charles Vanleuvan

Hip Hop Lyric Comparison

Introduction

In 2018, hip hop became the most streamed music genre in the United States, across all major platforms. Hip hop was a largely underground sound in the late 1970's in the south Bronx, New York, and quickly evolved into an international phenomenon by the year 2000. Throughout the 1980's, hip hop spread throughout neighborhoods across the United States via mixtapes and corner freestyling. It is a genre of music that draws heavily on cultural influences, past musical influences, and regional shoutouts which allow listeners to connect with artists from their home state.

The hip hop sound has evolved so much over 5 decades that distinct sub genres have formed. The sound across the decades has also seen a dramatic change, both in production type as well as lyrical content. Starting on the east coast, in New York, hip hop artists in the 1980s in NY had a sense of consciousness about their lyrics and delivered rhymes with a free-flowing rhythm. Contrasted with the west coast sound, known as G Funk or Gangsta Rap, the west coast hip hop sound utilized more synthesized beats along with introducing political commentary into lyrics. In the south, such as Atlanta and Miami, hip hop developed a club style sound, pioneered by DJs and artists such as Uncle Luke and the 2 Live Crew. In the Midwest, artists utilized high paced beats with quickly delivered rhymes to create a "chopper" sound.

The vast and fast-growing differences between regional hip hop sounds has warranted an investigation into the lyrical composition of songs across US regions and decades. The fruits of such an investigation would answer the question: Can a subgenre of hip hop be predicted from the lyrics? By analyzing the differences in diction, word frequency, and lyrical content across the regions and decades, differences between such focus areas can be identified and understood. The top music streaming platforms, such as Amazon, Apple, and Spotify, are likely using lyric content to build recommendation systems to suggest new songs to listeners based on the text content of the songs they are currently listening to. By fully understanding the lyrical makeup of the hip hop songs across these different regions and decades, a pathway to building a song recommendation system can be established.

Analysis and Models

Obtain Data

The Genius Lyrics API was used to obtain song lyrics for free. After creating a developer account and receiving a token credential, this API enables users to retrieve album, artist, lyrics, and song information for songs in the hip hop genre. The developers at Genius have also created a Python package called lyricsgenius that allows the user to run the API natively within a python file.

```
import lyricsgenius
genius = lyricsgenius.Genius(token)
```

Figure 1 – Calling the Genius lyrics API in Python

The resulting artist and song information is delivered in JSON format. Parsing the JSON object for the lyrics text will provide the desired lyrics. Billboard is an international music charting organization that informs the public on the popularity and rankings of songs. The Billboard records were filtered to retrieve all top hip hop songs for all time to build a list of hip hop songs to search for lyrics. Since Billboard only began keeping record of top hip hop songs since 1989, that year is the first year in the dataset. The song artist and title were then supplied to the Genius API to retrieve the lyric text.

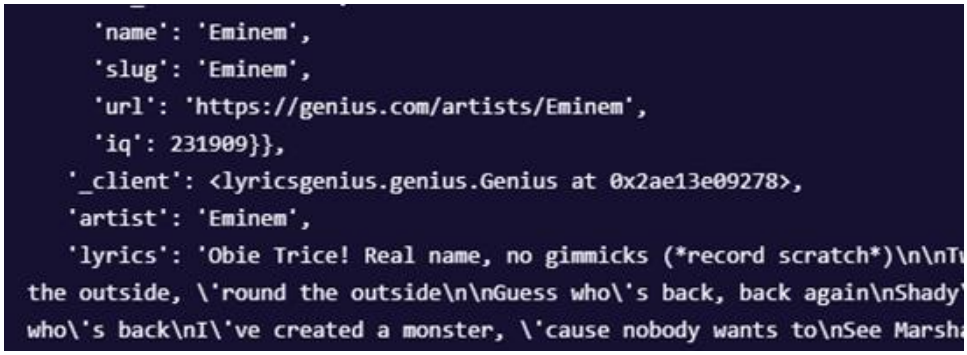


Figure 2 – The JSON object results from the API call

To build a corpus of lyrics, the results of the mass API search was placed into a Python Pandas data frame, with the lyrics stored as a column, shown below.

Title(from Genius)	Searched Song	Artist(from Genius)	Searched Artist	Lyrics	Date	State
Hotline Bling	Hotline Bling	Drake	Drake	you used to call me on my you used to you used to yeah you used to	10/10/2015	CAN
Summer Sixteen	Summer Sixteen	Drake	Drake	looking looking looking looking looking looking looking looking look	2/20/2016	CAN
Godâ€™s Plan	God's Plan	Drake	Drake	and they wishin and wishin and wishin and wishin they wishin on m	2/3/2018	CAN

Figure 3 - Lyrics data frame, with labels

Scrub Data

This dataset features manually annotated labels, as there was no feature in the Genius API to retrieve artist home state location. The category labeled was the main artist's home state, abbreviated with the standard US state abbreviations. The international artist labels are CAN for Canada, EU for Europe, and JAMAICA for Jamaica. The home state was identified for each artist via Wikipedia. The home states were then further categorized prior to implementing the Naïve Bayes model by grouping them into regional categories. The breakdown for the regions is shown below:

East	West	South	Midwest
NY	CA	GA	IL
MA	WA	FL	MO
NJ		NC	MI
PA		LA	OH
DC			MS

To format the data frame so that it is suitable for Naïve Bayes modeling, text vectorization was performed. This was done with the vectorizing functions supplied by the machine learning library Scikit Learn. Binary, Term frequency, and Normalized Term frequency vectorization were achieved with custom built vectorizing functions that utilized the prebuilt vectorizing functions from Scikit Learn. In all cases, the data frame lyrics column was cleaned to removed non-ASCII characters via a simple lambda expression with regular expressions. The leading and trailing whitespace for the lyrics text field was also

removed via a lambda regular expression. Then, by fitting the lyrics column to the vectorizing function, a term document matrix was produced. The binary term document matrix contains a Boolean "1" or "0" to indicate whether or not the term was included in that song, the term frequency matrix contains the count of that term in the song, and the normalized term frequency contains the normalized term frequency relative to how often that term occurs throughout all songs. The normalized matrix is shown below:

LABEL	act	actin	acting	action	actions	activate	activated
GA, CAN	0	0	0.021845	0	0	0	0
CA	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0
CA	0	0	0	0	0	0	0
CA, NC	0	0.027432	0	0	0	0.0402331	0
CA, NC	0	0.027432	0	0	0	0.0402331	0
NY	0	0	0	0	0	0	0
NY	0.026804	0	0	0	0	0	0
NY	0	0	0	0	0	0	0
NY	0	0	0	0	0	0	0
NY, CA	0	0	0	0	0	0	0
NY	0	0	0	0	0	0	0

Figure 4 – Term document matrix for the normalized data

In a term document matrix, each row represents a song, and each column is a word from the whole dictionary of words used across all songs in the data set. In the vectorization process, the region labels were added to the term document matrices via a for loop. Since the JSON result also included erroneous string formatting text, such as "\n" to indicate new lines, that also was removed from the lyrics prior to the vectorization process.

Typically, numbers are removed from a dictionary that will be used to generate a term document matrix. However, since rappers are prone to using numbers to represent where they are from or speak in a "code" to identify their crew, it was hypothesized that leaving numbers in the dictionary would be beneficial to model development.

KMeans Clustering

After vectorization, the data set is transformed into a term document matrix and ready for inspection. The first inspection of the data set involved clustering to observe how the data inherently groups itself. With four regional US labels, it was hypothesized that 4 clusters are ideally present in the data set. KMeans clustering was performed with the clustering functions provided by Scikit Learn. Using the entire dictionary of words in all song lyrics, clustering was iterated 10 times, each time incrementally increasing the number of cluster centroids created. This process is done to identify an "elbow" point, which is the point on a plot of SSE vs. Clusters where the decrease in SSE appears to hit a limit. SSE is the Sum of Squared Errors, which is the sum of distances for each sample in a centroid to the mean center of the centroid. Smaller SSE results in a "tighter" centroid, where the samples are very closely related to the other samples in that centroid. A plot of the SSE vs. Clusters is shown below:

Figure 5 – KMeans cluster elbow plot for term frequency data

In the plot for this data set, 3 or 5 clusters appears to be the ideal number of clusters. Since 4 regional labels are being used, this is closely aligned with the hypothesis, but not an exact match. Clustering is not a supervised machine learning model, however, and does not provide predictions. It simply shows how the data set naturally presents itself.

LDA Topic Modeling

Latent Dirichlet Allocation (LDA) modeling was also performed on the term document matrices. LDA is used to model the topics that are naturally present in the data set and see how separated these topics may be from each other. It was hypothesized that songs from different regions may separate themselves into distinct topics, topics which are unique and relevant to the region the artist is from. However, it was observed that the topics generated were not regionally focused, but artist or song focused. The topic bubbles are displayed graphically on a 2D plot, and the most relevant terms for the topic as well as the entire corpus are shown on the right, in the image below:

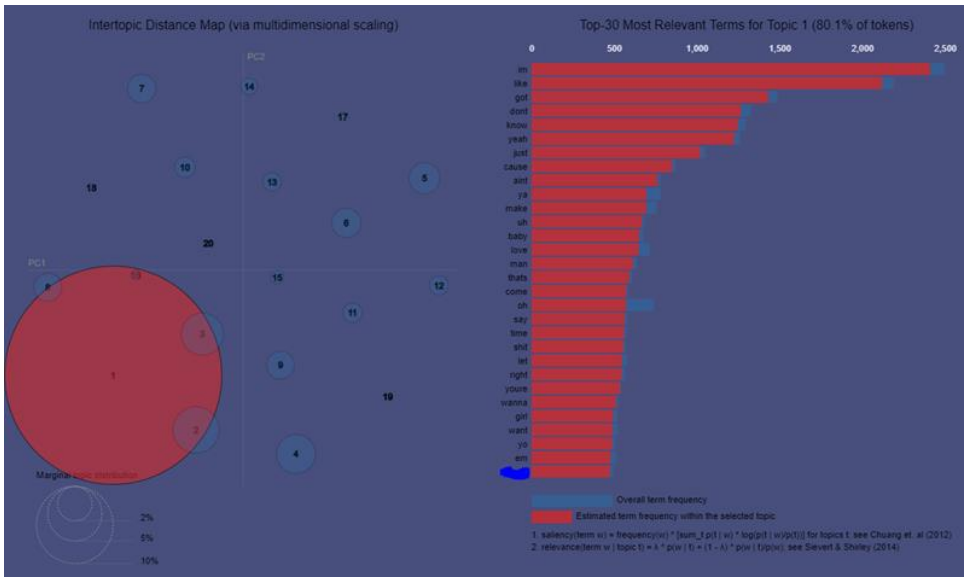


Figure 6 – LDA topic modeling, showcasing the top topic

The largest topic, which made up 80% of the tokenized words included many of the common "song" words, such as "baby", "love", or "man". These words are the foundational terms used to build hip hop songs, so it is clear why they would be so relevant across all songs.

The other topics, which occupy small, non-intersecting space in the 2D plot of the topics, show that they are made of the terms that are extremely specific to an artist or song. For example, the artist Rakim, known for larger, more complex words in his freestyle type raps, has a topic dedicated entirely to his songs in the data set. His relevant terms are shown below:

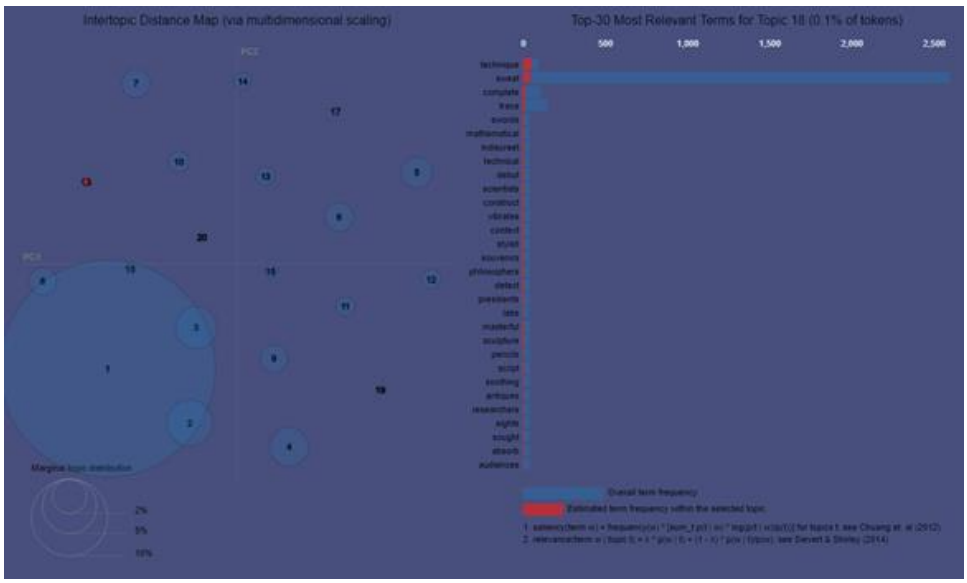


Figure 7 – LDA topic modeling, showcasing Rakim's topic

Meanwhile, the song "Jump" by Kriss Kross, which asks the listener to jump, so heavily featured the words "jump", "gotta", "make", that these words formed a separate topic just for Kriss Kross. This topic is shown below:

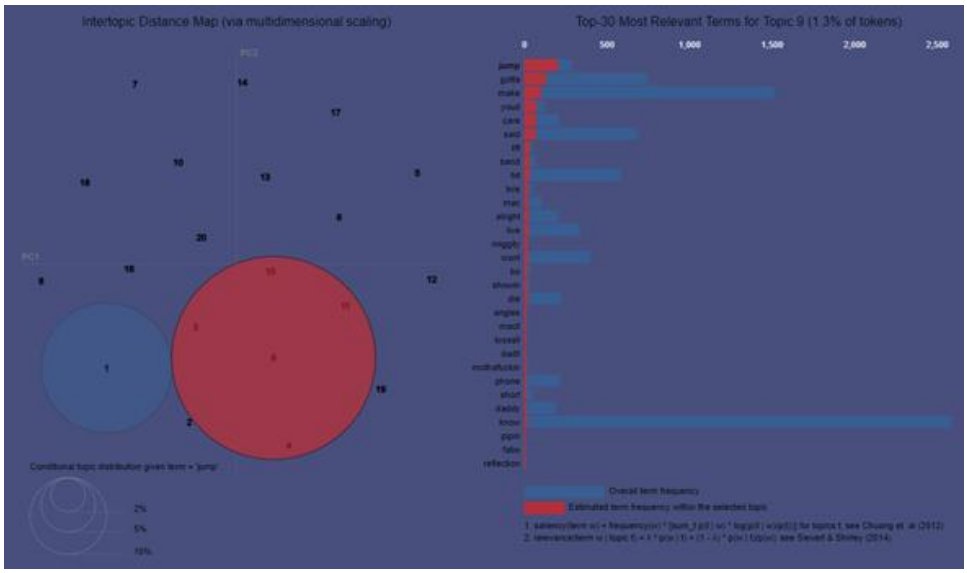


Figure 8 – LDA topic modeling, showcasing Kriss Kross topic

LDA topic modeling provided insight into how the Naïve Bayes model will perform, given the context that most of these songs used the same terms, like “I’m”, “got”, “know”. Overall, artists and songs would generate their own topic if the words used in the lyrics are repeated often enough as well as incredibly specific and unique. However, the size discrepancy between the top topic and the rest indicates that most songs utilize the same terms, which will make modeling and prediction difficult.

Exploratory Data Analysis

To get a better understanding of the data prior to modeling, the word frequency can help show what is there. There are multiple ways to show this, but one of the more effective and visually appealing ways to do this is with a word cloud. Within the purpose of this analysis there is opportunity to be creative with these word clouds, as the hip hop lyrics can be explored through looking at the frequencies by region and by decade.

For this to be properly done, the data needed a small amount of cleaning first. With the data frame read in, the first step is to remove any duplication of the songs. There are examples in the data of a song being featured at the top of the *Billboard* Charts more than once each year. Due to that, they can appear twice, doubling the frequency of some words. To prevent this, a drop_duplicates() command was ran on the “Searched Song” column.

The next step to prepare data for filtering by region. The “State” variable describes where the main artist and any featured artists come from or where they started out in their hip-hop careers. These are initially combined into one column but can be split out based on the comma into separate columns.

	Title(from Genius)	Searched Song	Artist(from Genius)	Searched Artist	Lyrics	Date	State
0	No Lie	No Lie	2 Chainz	2 Chainz featuring Drake	eardrummers mike will made it yah tru 2 chainz...	9/8/2012	GA, CAN
1	Me So Horny	Me So Horny	2 Live Crew	2 Live Crew	whatll we get for 10 dollars every ting you wa...	11/4/1989	CA
2	Banned in the U.S.A.	Banned in the U.S.A.	2 Live Crew	2 Live Crew	government of the people for the people by the...	8/25/1990	CA
3	Unreleased Songs [Full Discography List]	Dear Mama / Old School	2Pac	2Pac	1\tall bout u original version 2\tall out/die ...	3/11/1995	CA

Figure 9- Lyrics data frame with original "State" variable format.

	Title(from Genius)	Searched Song	Artist(from Genius)	Searched Artist	Lyrics	Date	State	Art.State	Ft.State1	Ft.State2	Ft.State3
0	No Lie	No Lie	2 Chainz	2 Chainz featuring Drake	eardrummers mike will made it yah tru 2 chainz...	9/8/2012	GA, CAN	GA	CAN	None	None
1	Me So Horny	Me So Horny	2 Live Crew	2 Live Crew	whatll we get for 10 dollars every ting you wa...	11/4/1989	CA	CA	None	None	None
2	Banned in the U.S.A.	Banned in the U.S.A.	2 Live Crew	2 Live Crew	government of the people for the people by the...	8/25/1990	CA	CA	None	None	None
3	Unreleased Songs [Full Discography List]	Dear Mama / Old School	2Pac	2Pac	1\tall bout u original version 2\tall out/die ...	3/11/1995	CA	CA	None	None	None

Figure 10 - Lyrics data frame with "State" variable split.

This split allows the visualizations to be centered on where the primary artist is from rather than all who are featured on a track. After the split of that variable is completed, that origin state for the primary artist can be matched against a dictionary that pair a state or country abbreviation with a region name. For example, California (CA) and Washington (WA) are linked to the West Coast Region and Maryland (MD) and New Jersey (NJ) to the East Coast Region. This matching to the dictionary was stored in a new column in the data frame named “Region”. It’s from this column that the data frame would be subset to look at the regions.

	Title(from Genius)	Searched Song	Artist(from Genius)	Searched Artist	Lyrics	Date	State	Art.State	Ft.State1	Ft.State2	Ft.State3	Region
0	No Lie	No Lie	2 Chainz	2 Chainz featuring Drake	eardrummers mike will made it yah tru 2 chainz...	9/8/2012	GA, CAN	GA	CAN	None	None	South
1	Me So Horny	Me So Horny	2 Live Crew	2 Live Crew	whatll we get for 10 dollars every ting you wa...	11/4/1989	CA	CA	None	None	None	West Coast
2	Banned in the U.S.A.	Banned in the U.S.A.	2 Live Crew	2 Live Crew	government of the people for the people by the...	8/25/1990	CA	CA	None	None	None	West Coast
3	Unreleased Songs [Full Discography List]	Dear Mama / Old School	2Pac	2Pac	1\tall bout u original version 2\tall out/die ...	3/11/1995	CA	CA	None	None	None	West Coast

Figure 11 - Lyrics data frame with “Region” column added and mapped.

At this point, the data frame can begin to be subset by the regions and the decades. For the regions, new data frames denoting the desired region were created by pulling in the data that were in the rows where the “Region” column matched the selected region. This resulted in data frames for the East Coast, West Coast, Midwest, South, and World regions. To subset for the decade, a similar process occurred. First, the “Date” column was changed to datetime format. Then, subsets were made for each decade, pulling in data based on rows that matched the criteria for a range of dates. Once completed, there were now data frames for the 1990s, 2000s, and 2010s.

With these data frames ready for each region and decade, the next step was to create the word clouds. This can be done by importing the wordcloud package, but to bring a little more life to the word clouds, the Image module from the PIL library was imported as well as ImageColorGenerator from the wordcloud package. These packages allow a word cloud to be masked to a shape while having custom colors and fonts for the word provided.

If working with a custom color palette, the first step for making a word cloud is to define the colors to be used in a small function. These are set by choosing values based on HSL (Hue-Saturation-Lightness) colors and storing them as a list within the function. If choosing a custom font, the path to where it is installed on the local machine is saved under the variable name “font_path”. Then, the path to the image is stored as the “mask” variable, where that Image module loaded in earlier imports the image data. The word cloud parameters are then set, removing any desired stopwords, setting the background colors, defining the width and height based on the stored “mask” variable, defining the font and the colors, and ultimately selecting the text to use – which in this case was the “Lyrics” column. Finally, several commands were run from matplotlib that generated the word cloud image itself. This was done for each region by the previously subset date. An example of the result is below.

```
## Custom color for South
def multi_color_func_S(word=None, font_size=None,
                       position=None, orientation=None,
                       font_path=None, random_state=None):
    colors = [[49, 100, 47],
              [186, 100, 47],
              [27, 100, 47],
              [320, 100, 47]]
    rand = random_state.randint(0, len(colors) - 1)
    return "hsl({}, {}%, {}%)".format(colors[rand][0], colors[rand][1], colors[rand][2])
```

```
## South
font_path = "C:\\Users\ergil\AppData\Local\Microsoft\Windows\Fonts\Blank River.ttf"
mask = np.array(Image.open("C:\\Users\ergil\Documents\SU iSchool MS in Applied Data Science\Classes\IST 736 - Text Mining\Fin
S_wordcloud2 = WordCloud(stopwords = stopwords, background_color = "white",
                         width = mask.shape[1], height = mask.shape[0],
                         font_path = font_path, mask = mask, mode = "RGBA",
                         color_func = multi_color_func_S).generate(' '.join(lyrics_South["Lyrics"]))
```

```
# Display the generated image:
plt.figure(figsize = (30,30))
plt.imshow(S_wordcloud2, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Figure 12 - Code and word cloud output for the "South" region.

The same process was done for each the decades, although the colors chosen were based off matplotlib's color library.

```
## 1990s
font_path = "C:\\Users\ergil\AppData\Local\Microsoft\Windows\Fonts\SivarPro.otf"
mask = np.array(Image.open("C:\\Users\ergil\Documents\SU iSchool MS in Applied Data Science\Classes\IST 736 - Text Mining\Fin
nineties_wordcloud2 = WordCloud(stopwords = stopwords, background_color = "black",
                        width = mask.shape[1], height = mask.shape[0],
                        font_path = font_path, mask = mask, mode = "RGBA",
                        colormap = "YlGn").generate(' '.join(lyrics_nineties["Lyrics"]))
```

```
# Display the generated image:
plt.figure(figsize = (30,30))
plt.imshow(nineties_wordcloud2, interpolation='bilinear')
plt.axis("off")
plt.show()
```

Figure 13 - Code and word cloud output for the 1990s decade.

Once the word clouds were generated for each of the regions and the decades, they were then loaded into Adobe Photoshop to save them as a single file. For the U.S. regions, they were positioned within an outline of the U.S., defining the regions more clearly with some border lines. For the decades, they were primarily positioned so that they were lined up properly. The results are seen below.

Figure 14 - Final word cloud for U.S. Regions.

Figure 15 - Final word cloud for World Region.

Figure 16 - Final word cloud for the decades.

Modeling (Bernoulli, Naïve Bayes, and SVM)

In order to prepare for a 5-fold cross-validation test, a 1-fold cross-validation test was first conducted as a proof of concept. This process was started by splitting the region labeled dataframe into training and test samples by using the train_test_split method from sklearn, saving one-fifth of the dataframe for testing while training the model on the remaining four-fifths of the rows. The labels were then separated from both the training and test frames and saved as their own variables. The training data and training labels were then fit to the multinomial naïve bayes model; this model was used to predict the values from the test data and then these predicted labels were compared with the actual test labels. Once this was done successfully, the proof-of-concept code provided a blueprint to evaluating the 5-fold cross-validation scores for all the subsequent models.

Additionally, the data was further scrubbed to improve the model results. In the initial modelling process, it was noticed that several of the models provided trivial results by predicting every song to be from one region. Upon closer inspection, this bias was caused by an imbalanced dataset where one region was much more heavily populated than all the other regions. Since the list of songs that were included in the dataset came from the Billboard number-one song of the Hot Rap Singles chart, it was not possible to fairly select additional songs for the under-represented regions. To resolve this, the dataset was subsampled by under-sampling the over-represented regions to match the region with the least amount of representation in the dataset. The resulting balanced subsampled dataset was for all the regional models.

Before Subsampling

Region	Song Count
East Coast	141
South	78
West Coast	34
Midwest	32

After Subsampling

Region	Song Count
East Coast	32
South	32
West Coast	32
Midwest	32

Figure 17 - Comparison of Song Counts for Each Region, Before and After Subsampling

Next, the KFold and cross_val_predict methods from sklearn were imported to do the full 5-fold cross-validation on both the region labeled and decade labeled data frames. This was done by specifying KFold to shuffle the rows and split the data in 5 different subsections (or folds). Each of these folds

would then be used as the testing data for in their individual splits, resulting in an aggregation of predictions to see the overall percentage of prediction errors in the overall 5-fold cross validation prediction. This process was repeated for each of the models: Bernoulli naïve bayes, multinomial naïve baye, and SVM with different kernels.

Results

Testing the nine different models of Bernoulli naïve bayes, multinomial naïve bayes, and SVM with linear, radial-based, and polynomial kernels between count vectorized and TF-IDF vectorized region datasets and then comparing the 5-fold accuracy results shows that the most accurate model for this dataset was the multinomial naïve bayes with count vectorization. However, this model still only had a 5-fold cross-validation accuracy of **36.72%**, only a few percentage points higher than both linear kernel SVM models and a mere 10% better than the expected results from random labeling. These results seem to indicate that the capability to distinguish a song's region based solely on the lyric vocabulary is extremely difficult, given the current dataset.

Model	5-Fold CV Accuracy
Bernoulli Naïve Bayes	26.56%
Multinomial NB (Count Vectorizer)	36.72%
Multinomial NB (TF-IDF Vectorizer)	22.66%
SVM: Linear Kernel (Count Vectorizer)	34.38%
SVM: Linear Kernel (TF-IDF Vectorizer)	35.16%
SVM: RBF Kernel (Count Vectorizer)	17.19%
SVM: RBF Kernel (TF-IDF Vectorizer)	14.84%
SVM: Polynomial Kernel (Count Vectorizer)	13.28%
SVM: Polynomial Kernel (TF-IDF Vectorizer)	14.06%

Figure 18 - Comparison of 5-Fold CV Accuracy between Region Models

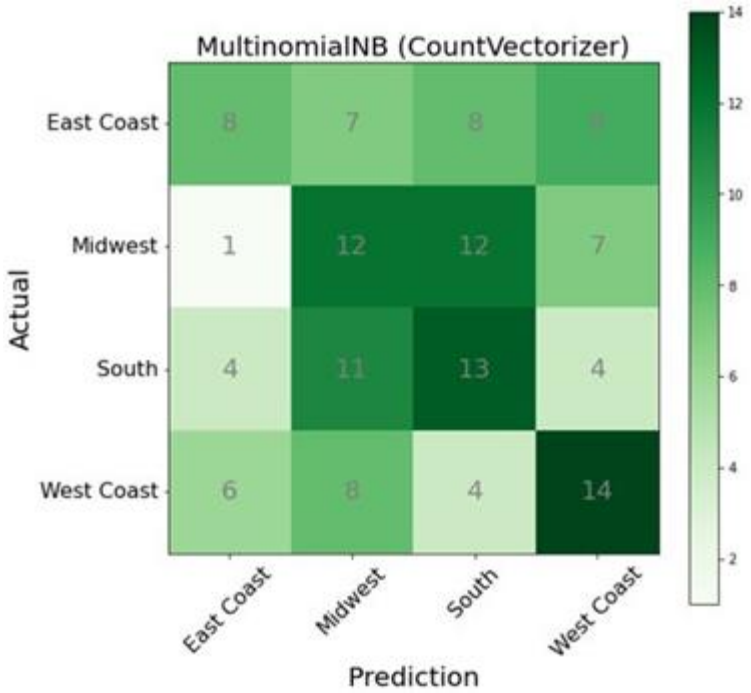
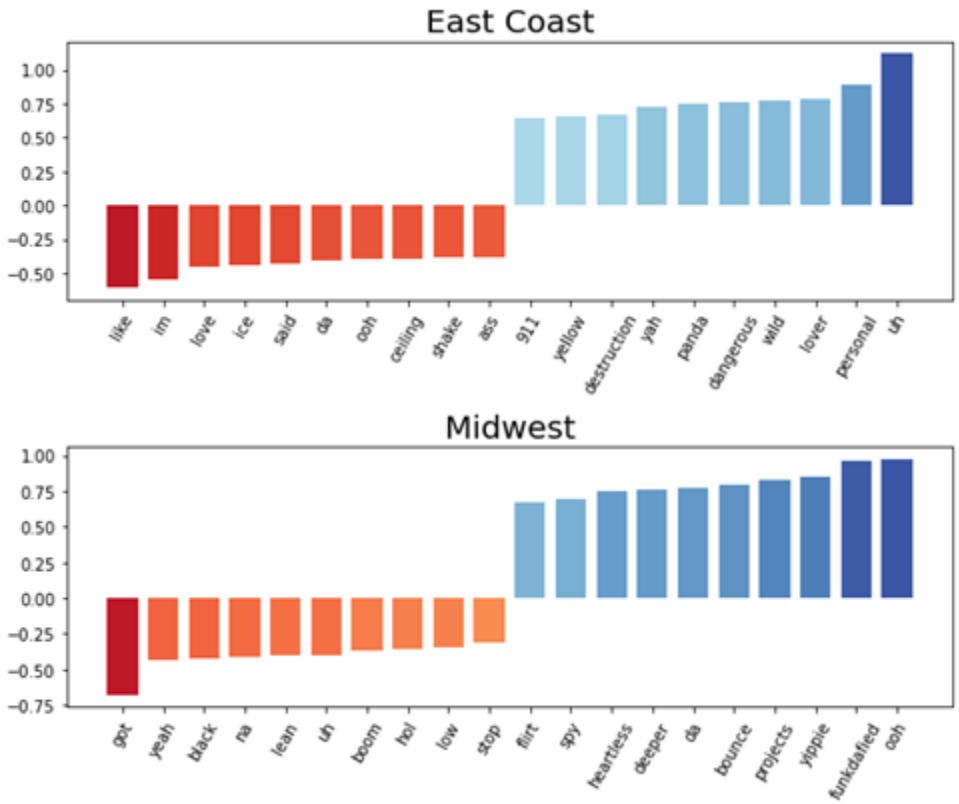


Figure 19 - Confusion Matrix for Most Accurate Region Model (Count Vectorized Multinomial NB)



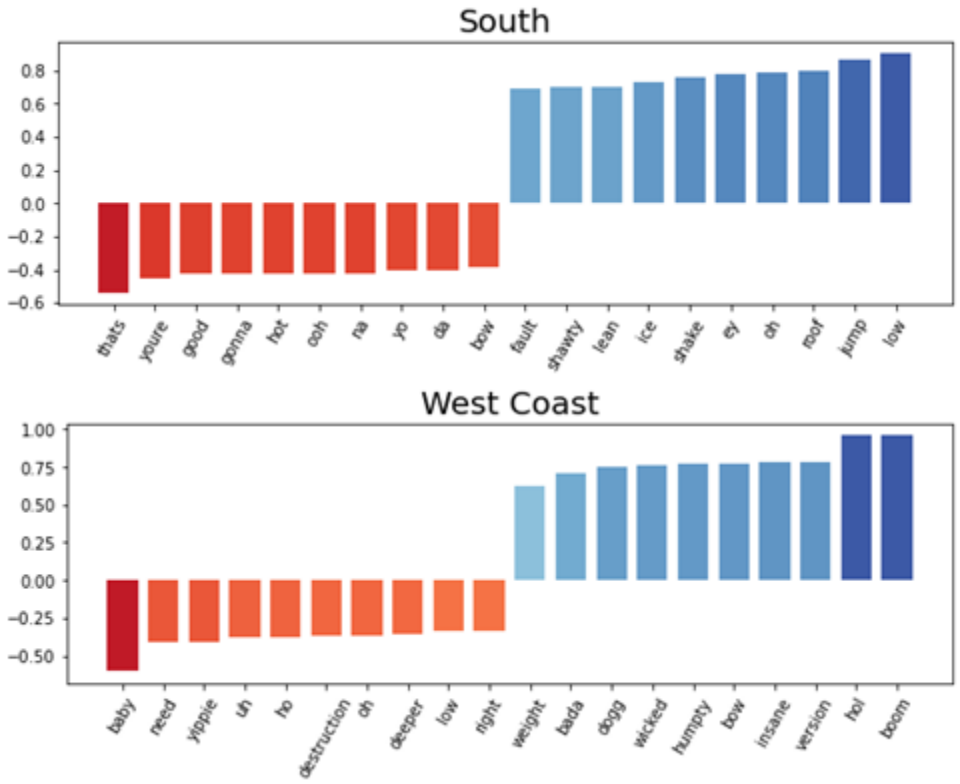


Figure 20 - Top Features for Region Modeling (by region)

Repeating this modeling process on the decade labeled vectorized dataset and then comparing the 5-fold accuracy results shows that the most accurate time period model for this dataset was also the multinomial naïve bayes with count vectorization. This model still had a 5-fold cross-validation accuracy of **70.22%,** much higher than the expected results from random labeling. These results seem to indicate that it is possible to distinguish a song's time period based solely on the lyric vocabulary. Additionally, the other model that also performed well above the baseline was the linear kernel SVM using TF-IDF vectorized data.

Model	5-Fold CV Accuracy
Bernoulli Naïve Bayes	56.89%
Multinomial NB (Count Vectorizer)	70.22%
Multinomial NB (TF-IDF Vectorizer)	49.33%
SVM: Linear Kernel (Count Vectorizer)	58.22%
SVM: Linear Kernel (TF-IDF Vectorizer)	66.22%
SVM: RBF Kernel (Count Vectorizer)	42.22%
SVM: RBF Kernel (TF-IDF Vectorizer)	37.78%
SVM: Polynomial Kernel (Count Vectorizer)	21.78%
SVM: Polynomial Kernel (TF-IDF Vectorizer)	27.11%

Figure 21 - Comparison of 5-Fold CV Accuracy between Decade Models

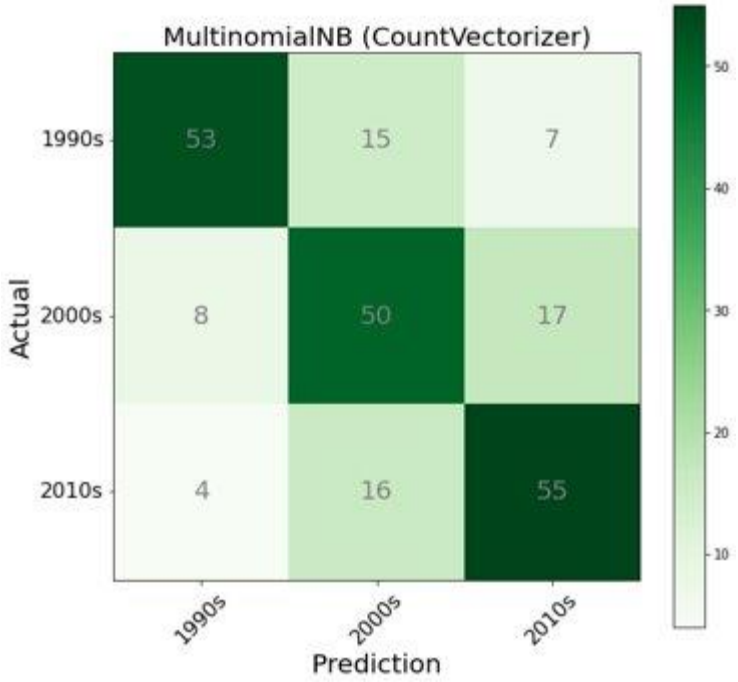
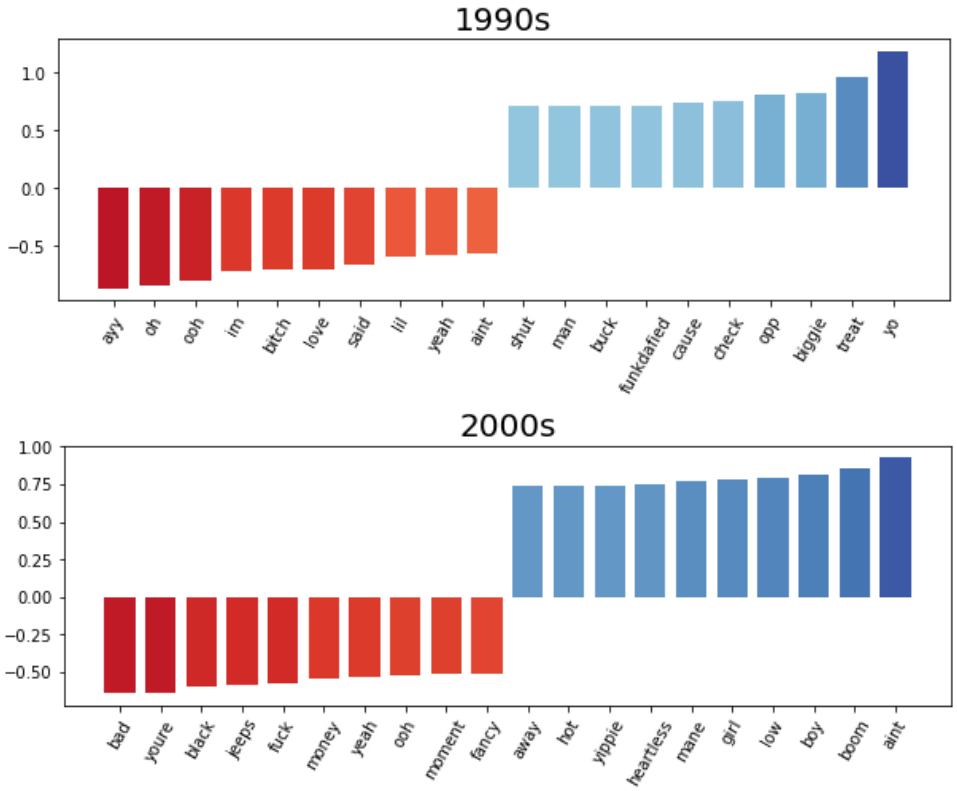


Figure 22 - Confusion Matrix for Most Accurate Decade Model (Count Vectorized Multinomial NB)



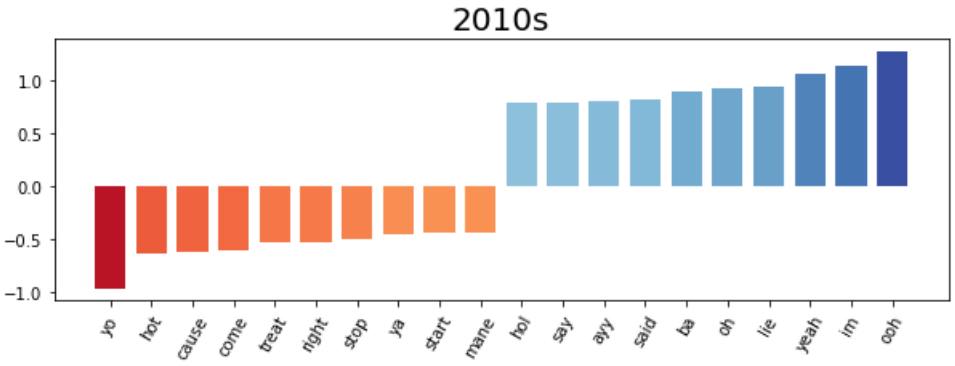


Figure 23 - Top Features for Decade Modeling (by time period)

Conclusion

Although hip hop began almost 40 years ago to humble beginnings, it quickly sprouted into a multibillion-dollar industry that has transformed past its musical roots into an increasingly lucrative lifestyle. And with the rise of a dominant industry comes the pursuit of companies that will find more ways to make the genre more accessible and available to the massive potential audience that it has yet to be exposed to it. For example, the differences in diction, word frequency, and lyrical relations across regions/decades could offer a large amount of insight in identifying differences between these categories. This has led many large tech corporations like Spotify, Apple Music, Amazon Prime, or Google to begin actively seeking text modeling methods that can use these underlying connections between lyrics and categories to provided enhancements on their music recommendation systems – even if the user is listening to a new artist that does not yet have an established genre.

Based on the results of modeling the vectorized lyrics data to both the region and decade labels for songs on *Billboard*'s number-one for Hot Rap Single chart, it was determined that the lyrical vocabulary of hip-hop songs between regions does not differ enough to create an accurate prediction model. However, the vocabulary choice between different decades has enough differences to distinguish between these time periods; for example, identifying that the song "Self-Destruction" is from a different decade that the song "Old Town Road", even though both songs were staples of the *Billboard* number-one for an extended period.

Overall, there were some improvements for identifying the region of a song by its lyrical vocabulary that could be gleamed from the data modeling and its results. First, a larger dataset could be used to get a more accurate portrayal of the lyric usage throughout a region. Due to the subsampling that resulted from the imbalanced starting dataset, each region relied on only 32 songs to build the entire feature set. For future analysis, the dataset could be constructed by selecting many songs from each region based on their regional or local popularity; this would ensure that every region is equally represented without removing any additional data. Another improvement that could be made is seeking the inputs from a subject matter expert on this topic, such as a regional linguist or another academic in a similar field. The guidance from this type of expert could help to shape the features that should be emphasized and the features that should be omitted, rather than using a default list of stop words built into a python package. Like other prominent document classification problems, sometimes common stop words can be important indicators of an author (or region's) writing style.

References

Exploratory Data Analysis Links:

Guide to Creating Word Clouds - https://towardsdatascience.com/create-word-cloud-into-any-shape-you-want-using-python-d0b88834bc32

Mapping Dictionary Values to Dataframe Columns - https://kanoki.org/2019/04/06/pandas-map-dictionary-values-with-dataframe-columns/

Wordcloud Documentation - https://amueller.github.io/word_cloud/cli.html

U.S. Outline Map - https://vemaps.com/united-states/us-02

Matplotlib Colormaps - https://matplotlib.org/stable/tutorials/colors/colormaps.html

DaFont Freely Downloadable Fonts - https://www.dafont.com

https://newsroom.spotify.com/2018-12-04/the-top-songs-artists-playlists-and-podcasts-of-2018/

https://docs.genius.com/

https://www.billboard.com/