Florence, Jameson, and Zongrui
CS 5001, Spring 2021
Final Project Report

1 INTRODUCTION

Personal Context & Motivations

Florence: My love affair with music started as a very young age as I learned the piano and later on study the flute traversiere for a few years. As a child, I wanted to play the flute so badly that I saved all the money my grandmother gave me for birthdays and Christmases, year after year, until I could buy one. I also inherited my grandfather's violin later in life (I was not aware of it until I was an adult as he died a few months after my birth) and without surprise I am sharing my life with a professional musician whose instruments are the viola and the violin. I have a passion for a broad variety of music and I am always listening to different sounds and genre from the world. I am paying attention to the quality of the sound and the melody as much as the way to produce it... I have done some foley sound effects in my animation projects and I have a particular interest in haptic sounds applied in XR. Being able to manipulate images and sounds in my future work would be ideal and this is why this project is of particular interest to me.

Jameson: I've been involved with music for most of my life, whether it be middle school band, high school choir, or singing in the shower. The opportunity to combine my love of music with my academic studies in computer science is fantastic! I've always been interested in looking into the different ways that we classify music, genre being one of them. While song lyrics and topics definitely play into the definition of a genre, I think that instruments, musical structure, and other unquantifiable things such as the way a song "sounds" play a more significant part in how we view a given song's or artist's genre. However, lyrics still play a large part in determining genre: read the lyrics of a typical pop country song and a typical hip-hop/rap song and you will see a marked difference. I am therefore very curious to see how accurately we can predict the genre of songs!

Zongrui: I started to play cello when I was seven years old. When I was grade 9, I passed the highest level of cello certification exam. I also attended many orchestra music competitions, and I also performed many times in my high school and university. During the summer break from high school to university, I was interested in playing violin. Since I have a solid foundation in cello, I can perform some simple songs in a short time. But I haven't played cello after I go to university because I was busy in my homework. Now, when this music topic of final project comes to my minds, I feel it very interesting to combine music with computer programming. That will bring me a unique experience with music.

Project Question

There are many qualities of a song that can be analyzed to determine its genre, but we want to focus on the lyrics. Therefore, our question is:

Can a computational model predict the genre of a song by analyzing the lyrics?

2 ANALYSIS

Model Design

Taking the time and knowledge constraints of this project, we chose to create a binary classification model that would classify songs from two genres: country and hip hop. Since we have learned Bayes Theorem in class, we can extend this knowledge to our final project. Naïve Bayes Classifier is highly related to Bayes Theorem and we can use Bayes Theorem to create a Naïve Bayes Classifier.

Here is Bayes Theorem: $P(y|X) = \frac{P(X|y)P(y)}{P(X)}$. In our case, we need to calculate the probability of y (y is the class variable Hip Hop or Country Song) given by all features as X.

X is the set of all features $x1, x2, x3, \dots, xn$ of size n. The characteristic of Naïve Bayes Classifier is that it can make each feature independent and equally contribute to the outcome. Since each feature is independent each other, the Bayes Theorem can be written as $P(y|x1, x2, \dots, xn) = \frac{P(x1|y)P(x2|y)\dots P(xn|y)P(y)}{P(x1)P(x2)..P(xn)}$. So, it is available to calculate those values by looking at the dataset and plug in to the equation. For all class variables, no matter Hip Hop or Country, the denominator is all the same. So, the denominator can be removed to keep it easy and we use a proportion to represent this relation.

Generally, there are three kinds of Naïve Bayes Classifier: Bernoulli Naïve Bayes, Multinomial Naïve Bayes, Gaussian Naïve Bayes. Since the features are not continuous values, and we care about the number of times each feature occurs, we choose Multinomial Naïve Bayes to create our model.

Here is the next formula: $P(y|x1, x2, \dots, xn) \propto P(y)\prod_{i=1}^{n} P(xi|y)$. Finally, we need to calculate the probability for each hip hop and country song given by the those features. This can be achieved by the outcome of $P(Hip)\prod_{i=1}^{n} P(xi|Hip)$ and $P(Country)\prod_{i=1}^{n} P(xi|Country)$. Then, compare both results to see which one greater than another one. If the probability of hip hop is greater than the probability of country song, that y should be classified as hip hop and vice versa.

Acquiring Training Data

Before we began programming the model, we first needed to find a dataset with which to train the model. We found and used Mendeley Data's Music Dataset: Lyrics and Metadata from 1950 to 2019 (Moura, Luan; Fontelles, Emanuel; Sampaio, Vinicius; França, Mardônio (2020), "Music Dataset: Lyrics and Metadata from 1950 to 2019", Mendeley Data, V3, doi: 10.17632/3t9vbwxgr5.3). It contained many features for each song, but most important were the song genre and lyrics. Each song had a genre tag and the song lyrics were already tokenized and lemmatized, reducing the amount of processing required to correctly format our data for the scikit-learn Multinomial Naïve Bayes model.

Programming the Model

After acquiring the full set of data, we used the Python csv module to import the training data into our program. From there, we created functions that used loops to change the songs' genre tags to "0" for hip hop or "1" for country and split the single string of lyrics for each song into a list of individual words. We then made a function to iterate through each song's list of words and convert it to a dictionary for each song, turning the lyrics into the "bag of words" format for analysis. The key/value pairs in these dictionaries represented each unique word that appeared in the song and its frequency of occurrence. For example, the key/value pair {"must": 3} would indicate the word "must" occurred thrice in the song. A set of all the unique words within the complete dataset was also created to properly format the testing data for analysis.

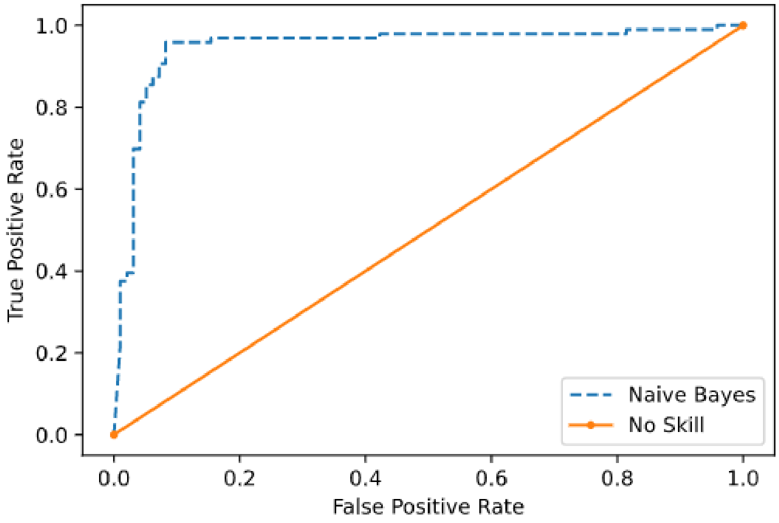
Once the data was converted to binary classes and bags of words, we split the data into two lists: one list containing the songs' classes and one list containing the songs' lyrics dictionaries. We used the sklearn DictVectorizer to convert the lyric dictionaries into feature vectors. Since the class list was already an array-type data structure, we simply used the numpy.array() method to create the class vector. Following the creation of the feature and class vectors, the sklearn MultinomialNB.fit() method was used to generate a trained machine learning model we could use to predict the genre (class) of other songs.

Acquiring Test Data

To thoroughly test our prediction model, we decided to use the Billboard Hot 100 Country and Hip Hop/R&B 2020 songs. Unfortunately, the Billboard website did not have a spreadsheet of this information available to download, so we manually transcribed each song and artist to a CSV file. We then registered with the Genius API to get a developer key in order to access Genius.com's database of song lyrics. Using the lyricsgenius package, we created a web scraping program to write the lyrics of each song to two text files: one for country and another one for hip hop. The csv module was used to read the lyrics into Python, after which we created a class vector for the test data genres and a feature vector for the song lyrics. To create the test feature vector, we needed to use the set of words in our training data to properly format our feature vector: any words in our test lyrics not in the training set were discarded, and any words in the training set not present in our test lyrics were added and given an occurrence value of 0. After converting these two lists into vectors, using methods identical to those used to create the training data vectors, we ran the feature vector into our model to make predictions.

Predictions & Results

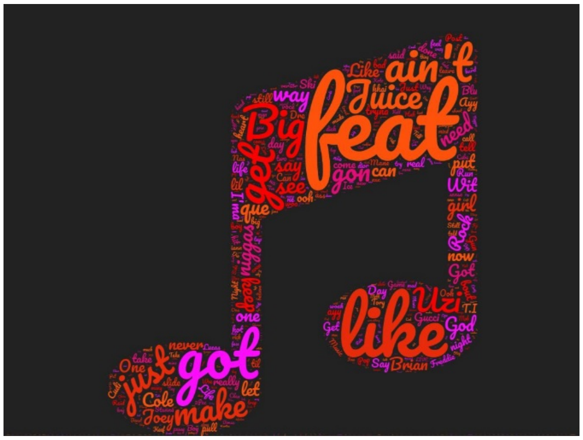
To generate a class prediction vector (actual y), we used the sklearn MultinomialNB.predict() method to generate an array of predicted classes, which we call the actual y results. Based on the actual y result and predicted y result, we used confusion_matrix function imported from sklearn.metrics library to make a confusion matrix. The true positive was 93, the false positive was 15, the true negative was 82, and the false negative was 3. Using the accuracy formula, we calculated $Accuracy\ = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} = 0.906$. That means we have successfully predicted 90.6% y binary result. Good prediction! Then, we introduced another machine learning tool, the Receiver Operator Characteristic (ROC) curve to test how good our model is. The ROC curve shows the trade-off between the true positive rate and false positive rate using different thresholds. We used the roc_curve function to generate a true positive rate and false positive rate for different thresholds and make the true positive rate and false positive rate as inputs to create a ROC curve by using matplot.pyplot library. The ROC curve that is closer to the top-left corner means a better prediction.



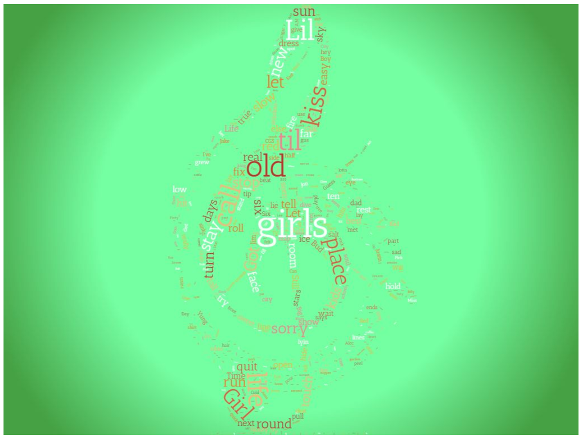
In our ROC curve, we can see it is close to top-left corner, which is not bad. Then, we we calculated the area under the ROC curve, which is AUC value. The AUC value is always between 0.5 and 1. A larger AUC value indicates better model performance. Our AUC value is 0.948, a very fantastic number. Both accuracy and AUC values show our prediction was successful.

Linguistic Insights

We visualised the word frequency that we got from our model using a wordcloud. Each genre is illustrated by an image. The bigger the word represented on this illustration, the more it is used in a genre:



Hip Hop wordcloud



Country wordcloud

We took the first 50 words that weigh the most in each genre.
We had to remove some words such as artists' names and some that appeared too many times in the data set. They didn't represent the musical genres.

When we analysed the results from the classifier, we found that some words are used in both genres, for example: " yeah", "love", "ain't", "like", "got", "get", "just", "know", "back", etc. They are mostly verbs and interjections. One word that was heavily used by both genres was the word "like". These findings are not surprising as a lot of lyrics in songs talk about feelings and actions.

This classifier helped us to discover an interesting fact. The word "feat", the most used word in Hip Hop, is used around 2,553 times. It is not used at all in Country music.
In comparison, "one" is the most used in Country music and it is only used 191 times.
We can see that there is a noticeable difference in the frequency of certain words between the two genres. Country lyrics have a tendency to use more vocabulary compared to hip hop lyrics. There is a repetitive pattern to Hip Hop that is heavily centered on specific words such as "feat", "yeah", "baby", "like", "lil", etc...
Country lyrics are more focused on vocabulary centered around morality such as "hell", "right" and "good". Another word that helps differentiate between the two genres is the word "beer" that is frequently used in country music.

In general, we can extrapolate that you can find the same words in both genres. However, the fact that the classifier distinguishes those top words represented with a wordcloud helps contribute to the high accuracy of 90% in our findings (you can see this in the area under the ROC's curve).

The false positive rate of around 15% is due to several factors. Certain words are common in both songs, which can be tricky, especially with the heavy use of words "like", "yeah", "ain't" ,"get" etc...so the two genres can be easily mistaken for one another in certain songs. Still, you are almost guaranteed to correctly predict all the country songs or hip hop songs. Not a lot are missing.

The fact that hip hop and country have such different lyrics really helps to get a true negative of 82% as some words such as “feat” for Hip Hop and “beer” for country are rarely shared. The false negative could be due to some names that are in common for both genres. For example, Justin Bieber is seen in both, and can be put in the country genre by mistake.

3 Conclusion

Results Summary

Given our model’s high level of accuracy at predicting whether a song is country or hip hop, the answer to our project question is yes! Computational models CAN accurately predict a song's genre by analyzing its lyrics.

However, there are some limitations. We used a binary classification model, which could classify a song into one of only two genres. Additionally, the two genres we selected (country and hip hop) are highly distinguishable from each other compared to other potential genres. If our model incorporated more genres (say, country, hip hop, rock, EDM, and pop) model would decrease. The bag-of-words model we used to prepare the lyrics is also limiting. Analyzing each word independent of the other words ignores the significance of word order and sequences, degrading the ability of the model to analyze the differences among genres.

There are also limitations with the data we used to train and test the model. It was challenging to find a substantial and thoroughly vetted data set of song lyrics. Songs are copyrighted material, making them more difficult to procure and distribute for any purpose, even academic. Additionally, there is no industry database of accurate, vetted song lyrics. Almost every lyrics site, including Genius (whose API we used to scrape our test data), relies heavily on user submissions and edits, leading to sometimes questionable accuracy, spelling, and completeness of the lyrics. The training and testing datasets were also small: we only trained with approximately 6,000 songs and tested with just under 200. Our model possibly ignored many other songs that could be more dissimilar from our test data, leading to an inflated accuracy score. The same is true for our test data; we could have chosen a small number of songs that serendipitously fit our trained model while ignoring songs that would be more challenging to classify.

Areas for Future Research

There are many opportunities for further exploration of song genre analysis using natural language processing and machine learning. We could use neural networks and deep learning to develop a model that can learn which words are most important and optimize itself to improve prediction accuracy. They would reduce the amount of work needed to lemmatize clean the data before training the model and allow words to be weighted by relative importance, not just by frequency. We could also combine different music features to make better predictions; for example, analyzing the audio waveform of a song to determine the tempo, key, and instrumentation in addition to the lyrics. This holistic song analysis could offset the limitations of analyzing each feature independently.

Personal Impact

Florence: From the project, I learned how to use the Bayes Theorem for prediction applied to a real life example. It helps me understand how to make a computational model to predict the genre of a song by analysing the lyrics. I realized that this method can be applied to a lot of things. I can use the NLTK module (natural language toolkit) for Python to create a Bayes classifier and interpret a binary classification model (based on 0 and 1).I learned what a ROC curve is (receiver operating characteristic) and how to interpret it. I learned what a word cloud is and how to make one.

This project made me realize the potential that machine learning can have. I would be interested in applying those principles in a system that could detect skin cancer in people. Creating a system that can allow millions of pictures to be fed into a computer and using it as a detection tool, could be lifesaving. I can imagine people having a scanner incorporated into their phones and using it as a medical device to check for early signs of the disease. It would be convenient and also so easy to use. People wouldn’t need to go to see a doctor. There would be less delay and people would be more likely to detect this disease early on thus increasing their chances of survival.

Jameson: I learned quite a bit from this project! It was very rewarding to see how a relatively simple concept like Bayes’ Theorem can be expanded to create complex machine learning models that can accurately make predictions using the English language! Programming the model sharpened my Python and coding skills, and it was very empowering to be able to build such an accurate model that combined advanced topics like machine learning and natural language processing after only one semester of formal instruction in basic computer science and discrete mathematics. It was also fun to learn about data scraping and file I/O when obtaining and processing the test lyric data. I now have a great interest in further pursuing machine learning and artificial intelligence academically and professionally! This project and report will be a great reference and foundation for even more machine learning projects in my future.

Zongrui: In this project, I learned the extension of Bayes Theorem, Naïve Bayes Theorem, and that will lead me to get in touch with machine learning. Through seeking for resources, I know how to build model to predict the data, which is very fun. Also, I learned what the ROC curve is and how to create ROC curve in python. This experience is valuable for me because it can practice my self-learning ability. I cannot learn everything inside the class. If I take the other classes and go to work in the future, I must have the self-learning skill to solve the problem. This project is the chance for me to know how to solve problems outside the class.