Alec Arroyo
3/21/22
Final Project

Music Data Classification

Introduction:

The purpose of the final project was to find value from a dataset using machine learning models. What we choose to use for the final project was intaking a Kaggle dataset containing music lyrics of various artists and train and test models to predict what genre of music a certain song is. This project is targeting upcoming and startup music companies and labels, that are eager to sign growing artists and make money based off of the best's artists for the most popular genres of music. Predicting the genre of music based on song lyrics could be very beneficial to the music company. One reason could be to only sign artists are currently categorized in the most popular genres of music. This would help them grow more quickly which could generate more revenue. Another reason for this task could be to help starting artists declare a music genre that they fit in with or categorize with. If there is a new artist that doesn't belong to a certain genre of music, you could read in their song lyrics and using the models developed see what category they belong to.

To determine the best solution for predicting genres of music, the two algorithms that fit best with this task are using Naïve Bayes classification, and Support Vector Machines (SVM). We will be intaking the dataset and applying preprocessing and vectorization techniques that will help shape the data into a useable format that can be trained and tested on for analysis. Some of the preprocessing steps that will be ran will include lowercasing words, applying stemming, removing stop words, removing punctuation/whitespaces/nulls, and then finally vectorizing the dataset. After that, the data will be split into training and testing datasets, and we will apply them to the models first using the holdout method, then after using cross validation. When applying cross validation, we will be setting the number of folds equal to 3.

The dataset that was chosen for this classification came from a Kaggle dataset that contained two different CSV files, one related to the Artist of a music genre, and one related to the lyrics of a song for each genre of music. The first dataset for artists contained 3,200 rows of data and the lyrics dataset contained 209,522 rows of data. This gave a large enough sample size to work with, when splitting our data for analysis. In addition, in terms of the number of different genres of music there were to work with, there were 6 different types. These 6 genres of music included Samba, Funk Carioca, Hip Hop, Sertanejo, Pop, and Rock. This allows a wide variety of different ranges of music, and even includes different languages of music such as English, Spanish, and Portuguese.

Method:

To start with the two CSV datasets, they were loaded in Python using the read_csv function and then visualized it at a first glance:

	Artist	Songs	Popularity	Genre
0	10000 Maniacs	110	0.3	Rock
1	12 Stones	75	0.3	Rock
2	311	196	0.5	Rock
3	4 Non Blondes	15	7.5	Rock
4	A Cruz Está Vazia	13	0.0	Rock
5	Aborto Elétrico	36	0.1	Rock
6	Abril	36	0.1	Rock
7	Abuse	13	0.0	Rock
8	AC/DC	192	10.8	Rock
9	ACEIA	0	0.0	Rock
10	Acid Tree	5	0.0	Rock

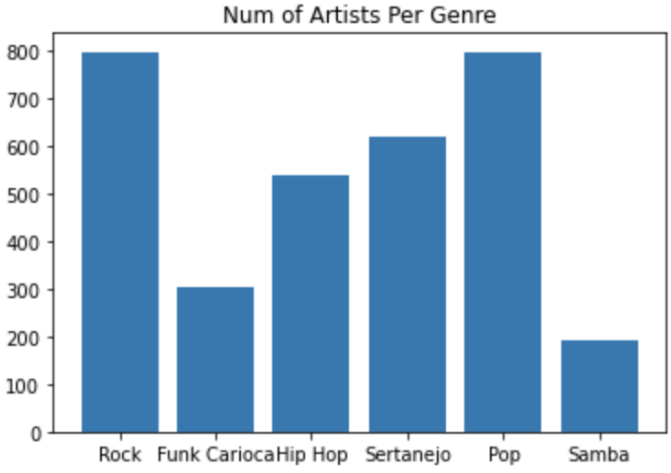
Artists Dataset

	Song	Lyric
0	More Than This	I could feel at the time. There was no way of ...
1	Because The Night	Take me now, baby, here as I am. Hold me close...
2	These Are Days	These are. These are days you'll remember. Nev...
3	A Campfire Song	A lie to say, "O my mountain has coal veins an...
4	Everyday Is Like Sunday	Trudging slowly over wet sand. Back to the ben...
5	Don't Talk	Don't talk, I will listen. Don't talk, you kee...
6	Across The Fields	Well they left then in the morning, a hundred ...
7	Planned Obsolescence	[music: Dennis Drew/lyric: Natalie Merchant]...
8	Rainy Day	On bended kneel've looked through every window...
9	Anthem For Doomed Youth	For whom do the bells toll. When sentenced to ...
10	All That Never Happens	She walks alone on the brick lane,. the breeze...

Songs/Lyrics Dataset

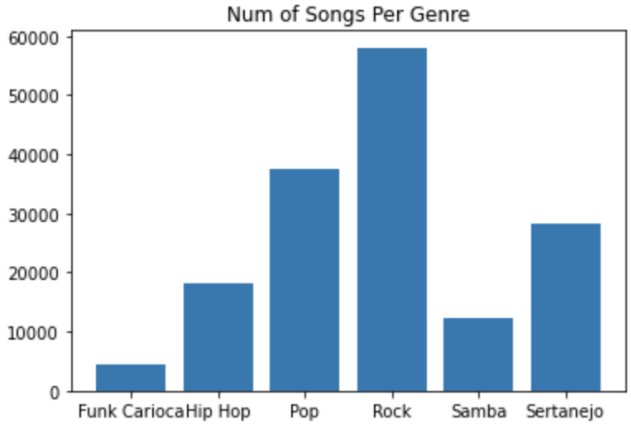
From the screenshots, you can see that the Artists dataset has four different fields named artist, songs, popularity, and genre. You can also see that the Lyrics dataset has some columns as well named song and lyric. To get an understanding of the number of different artists per genre of music, we created a table displaying that information:

	Genre	Total_Count
0	Samba	193
1	Funk Carioca	302
2	Hip Hop	537
3	Sertanejo	617
4	Pop	796
5	Rock	797



You can see that the representation of each genre of music is seemed to be skewed towards Pop and Rock, and not well representative of Samba. However, in order to do our analysis and get a better understanding of the data, we will need to combine the two datasets together in order to use the lyrics field to predict the genre of music. To do that, we used the pandassql python packages to write a SQL query to combine the datasets together:

	Artist	Songs	Popularity	SName	Lyric	Genre	Idiom
0	Luan Santana	187	17.2	"A"	Tá em dúvida. Não sabe se é normal gostar de d...	Sertanejo	PORTUGUESE
1	Mutantes	123	1.0	"A" e o "Z"	Eu sou o começo. Sou o Fim. Sou o A e o Z. Eu ...	Rock	PORTUGUESE
2	Foxy Brown	74	0.0	"Oh Yeah" By Foxy Brown	[Verse One]. I'm the most critically acclaimed...	Hip Hop	ENGLISH
3	Barbie Kue	8	0.0	"Sei Lá..."	Se um dia olhar pro lado. Eu não vou mais esta...	Rock	PORTUGUESE
4	Tiziano Ferro	160	2.1	"Solo" e' Solo Una Parola	Il cuore è andato in guerra ma la vita. non l'...	Pop	ITALIAN
...	...	...	...	...	...	...	...
161284	Ariana Grande	155	246.8	Goodnight n Go	Tell me why you gotta look at me that way. You...	Pop	ENGLISH
161285	Girls' Generation	248	0.4	쉼표 (Fermata)	Maeumi swineun dosi. eopseul georan geol ara. ...	Pop	None
161286	BigBang	175	1.0	착한 사람 (a Good Man)	Chakhansaramieosseum angeuraetjyo. Geureona na...	Hip Hop	None
161287	BigBang	175	1.0	천국 (heaven)	saranghae nan neol gieokhae. HEAVEN. SING IT T...	Hip Hop	None
161288	BigBang	175	1.0	하루 하루 (haru Haru)	YEAH. FINALLY I REALIZED. THAT I'M NOTHING WIT...	Hip Hop	None



From the screenshots we can see the dataset has been combined and it contains 161,289 rows. You can also see from the bar chart a better view of the data representation for each genre of music. From that chart, it looks like there's more representation for rock and pop, then the genre of the others especially Funk Carioca. This leads to an unbalanced dataset where there is too much or too little representation of one or more categories. This will need to be taken into account, when running our analysis and making conclusions. Now that the data was loaded and observed, there were several issues that needed to be addressed. Some of the issues for the Lyrics field included lowercase words, lack of stemming, lots of stop words, and blanks and whitespace that need to be removed. This would lead to the preprocessing step of the data.

In order to do this, a function was defined called vectorize that contained all preprocessing steps that needed to take place. This included first making all words lowercase, then adding stemming using the PorterStemmer function, and then removing stopwords. To remove stopwords, the NLTK package was applied where stopwords were removed in English, Spanish, and Portuguese. After, it proceeds to remove punctuation, empty strings, and whitespace. After applying the function to the final dataset, we get the following values:

	Artist	Songs	Popularity	SName	Lyric	Genre	Idiom
0	Luan Santana	187	17.2	"A"	tá em dúvida não sabe se é normal gostar de do...	Sertanejo	PORTUGUESE
1	Mutantes	123	1.0	"A" e o "Z"	eu sou o começo sou o fim sou o a e o z eu sou...	Rock	PORTUGUESE
2	Foxy Brown	74	0.0	"Oh Yeah" By Foxy Brown	verse one im the most critically acclaimed rap...	Hip Hop	ENGLISH
3	Barbie Kue	8	0.0	"Sei Lá..."	se um dia olhar pro lado eu não vou mais estar...	Rock	PORTUGUESE
4	Tiziano Ferro	160	2.1	"Solo" e' Solo Una Parola	il cuore è andato in guerra ma la vita non lho...	Pop	ITALIAN
...	...	...	...	...	...	...	...
161280	MV Bill	92	1.7	é Nós E A Gente	a paz não precisa ser um sonho basta o respeit...	Hip Hop	PORTUGUESE
161281	Furacão 2000	57	2.8	é O Kit	é o kit é o kit furacão se liga mané é o kit t...	Funk Carioca	PORTUGUESE
161282	Fundo de Quintal	315	4.6	ô Irene	ô irene ô irene ô irene ô irene vai buscar o q...	Samba	PORTUGUESE
161283	Vremya I Steklo	5	0.0	На Стиле	а мы на стиле даже и не думай чтото отменять у...	Pop	RUSSIAN
161284	Ariana Grande	155	246.8	Goodnight n Go	tell me why you gotta look at me that way you ...	Pop	ENGLISH

158695 rows × 7 columns

We see that there were numerous rows removed due to the preprocessing steps making the total row count of 158,695 rows. Now that we had the data preprocessed, we can split the data into training and testing datasets.

Results:

To start, there were two arrays created that each represented the lyrics field and the genre field labelled X and Y. Then, using the train_test_split function from the sklearn.model_selection package, we split the data 60% training and 40% testing. After splitting the data, the total row count for the training dataset was 95,217, and the total row count for the testing dataset was 63,478. To get an idea of the representation of the genres of music, a table was created for it:

```
[['Funk Carioca' 'Hip Hop' 'Pop' 'Rock' 'Samba' 'Sertanejo']
 [2589 11011 22447 34998 7217 16955]]
```

Num Training Ex Per Genre

You can see from the number of training examples per genre of music, that there is more representation of Pop at 22,447 and Rock at 34,998 than the other categories. You can also see that there is a small portion of Funk Carioca at 2,589. This is from the dataset itself not having a large enough representation of Funk Carioca and having too many examples of Pop and Rock. However, the datasets will work for training and testing purposes on the models. Now that the splitting is set up, it is time to apply them against the models.

Starting with the Naïve Bayes model, the feature was set up using a unigram Boolean vectorizer, with a minimum document frequency equal to 5. This was done using the CountVectorizer function provided from the sklearn.feature_extraction.text package. The minimum document frequency was set to 5 to exclude any words in the vocabulary that have the total count of frequency less than 5 times. In doing this, we remove any unwanted words

that are not important. Once the vectorizer was set up, we then applied the vectorizer on the training dataset using the fit_transform function that will fit the data and transform it. After that we used the vocabulary constructed from the training data to vectorize the testing dataset.

The next step was to apply the vectorized training data on the Naïve Bayes model by using the MultinomialNB function provided by the sklearn.naive_bayes packages. We created the model for it and then used the training data to train the model. To view the holdout raw accuracy of the model after training it, we ran the score function to test the model and give us the accuracy. From doing this, we got an accuracy of 62.96% from the holdout method. To try and train a better model, we then ran cross validation on the dataset setting the number of folds equal to 3, which then gave us an accuracy of 71.05%. This is a significant increase as compared to using the holdout method. After running that, we used the classification_report function to view the precision, recall, and F1-score of the accuracy:

	precision	recall	f1-score	support
Samba	0.45	0.73	0.56	1751
Funk Carioca	0.80	0.55	0.65	7183
Hip Hop	0.69	0.20	0.31	15073
Sertanejo	0.67	0.80	0.73	23042
Pop	0.53	0.50	0.52	5065
Rock	0.57	0.94	0.71	11364
accuracy			0.63	63478
macro avg	0.62	0.62	0.58	63478
weighted avg	0.65	0.63	0.60	63478

From the measures, we can see that Funk Carioca had the best precision score at 80% while Samba had the worst precision score at 45%. From the recall, we can see that Rock had the best recall score at 94%, while Hip Hop had the worst recall score at 20%. Finally, from the F1-score we can see that Sertanjo had the best FI-score at 73%, while Hip Hop had the worst F1-score at 31%. If we look at the confusion matrix for the model, we can get a greater detail on how the model was scoring each genre of music:

	P:Samba	P:Funk Carioca	P:Hip Hop	P:Sertanejo	P:Pop	P:Rock
T:Samba	2546	95	0	2344	67	13
T:Funk Carioca	77	1280	3	385	3	3
T:Hip Hop	268	856	3976	419	738	926
T:Sertanejo	315	275	0	10683	69	22
T:Pop	700	194	697	2293	3026	8163
T:Rock	873	152	306	2753	505	18453

From the results we can see that for the Samba genre of music, most of the predictions were correct getting 2,546 right. It seems that the second most similar genre to Samba according to the model was Rock at 873 times falsely predicting it as Samba. When we look at Funk Carioca using Naïve Bayes model, we see it was correct for 1,280 instances. We can also see that the model predicted Hip Hop as being very similar to Funk Carioca even though it was

wrong. For hip hop, it was very accurate predicting it correctly for almost all instances at 3,976. When viewing Sertanejo, we see that the model was all over the place in predicting it correctly confusing it mostly with Rock, Pop, and Samba. For Pop, the model was very accurate in predicting it correctly at 3,026 times, and was constantly falsely predicting it for Hip Hop next at 738 times. Finally, for Rock we saw a large number of cases when the model correctly predicted it correctly at 18,453 times.

After using Naïve Bayes to predict genre classification form the lyrics data, it was then time to try Support Vector Machines to see if the accuracy would improve. To start, since we already had the training and testing dataset ready, all we had to define was the model. To do that, we ran the LinearSVC function provided by the sklearn.svm package. We defined the model and set the value of C equal to 1, which represents the strength of the regularization. After doing that we fit the SVM model to the training data and tested the values to see the initial score using the holdout method. When we ran it, we got an accuracy of 68.16% which is better than the holdout value for the Naïve Bayes model. To see if we could get the accuracy to improve, we ran cross validation on the model setting the number of folds equal to 3. In doing this we get an accuracy of 71.05% similar to the naïve bayes cross validation accuracy. However, when looking at the full decimal value, there is a slight difference at the end. When we look at the precision, recall, and F1-score of the model we get a greater detail:

	precision	recall	f1-score	support
Samba	0.72	0.66	0.69	1751
Funk Carioca	0.79	0.72	0.75	7183
Hip Hop	0.53	0.53	0.53	15073
Sertanejo	0.72	0.74	0.73	23042
Pop	0.61	0.58	0.60	5065
Rock	0.76	0.79	0.78	11364
accuracy			0.68	63478
macro avg	0.69	0.67	0.68	63478
weighted avg	0.68	0.68	0.68	63478

From the screenshot you can see the precision, recall, and F1-score against all 6 genres of music for the SVM model. For the precision, Funk Carioca had the best precision score at 79%, while Hip Hop had the worst precision at 53%. For the recall measure, Rock had the best score at 79%, while Hip Hop once again had the worst at 53%. Finally, for the F1-score, Rock has the best F1-Score at 78%, while Hip Hop had the worst F1-score at 53%. From these results it looks like for the SVM model, it was able to predict Rock the best averaging around 77.7% accuracy across all measures. On the other hand, it did the worst job at predicting for Hip Hop given, it scared 53% in all three measures which was the lowest of all the genres. When we check the confusion matrix of the model, we get a more descriptive view of how the model categorized each genre of music:

	P:Samba	P:Funk Carioca	P:Hip Hop	P:Sertanejo	P:Pop	P:Rock
T:Samba	2957	72	66	1065	503	402
T:Funk Carioca	123	1155	59	182	151	81
T:Hip Hop	71	69	5144	100	1099	700
T:Sertanejo	860	151	40	8993	795	525
T:Pop	436	94	795	841	7957	4950
T:Rock	393	54	373	616	4541	17065

From the confusion matrix we can see that the model did a little bit of a better job predicting Samba correctly than the Naïve Bayes model did predicting it 2,957 times correctly. For Funk Carioca, the SVM model predicted it very accurately for 1,155 times. When looking at Hip Hop, it looks like it predicted it correctly for 5,144 songs and then there were 795 times where the model mistakenly picked Pop for Hip Hop. When looking at Sertanejo, the results are very skewed for all predictions, but it mostly did a good job predicting at 8,993 times. We can see from Pop that it did a good job predicting most of the testing lyrics at 7,957, while also mistaking Rock and Hip Hop the most for Pop at 4,541 and 1,099. Finally, when looking at rock, it seems it did the best job at predicting with 17,065 times predicting it correctly, and having Pop come in second as mistaking it for that genre 4,950.

After observing the model's effects on the unbiased dataset, we began to apply the same measures on a balanced version of the dataset. In order to decrease the number of lyrics data for Pop and Rock and increase the amount of data for Funk Carioca while also balancing the other genres, we can apply oversampling to all the genres, to be able to have a fair representation of each genre of music. To do this, we used the RandomOverSampler function provided by the imblearn.over_sampling package to sample with a sampling strategy of "not majority" which means to constantly populate and sample the data to all the genres of music besides the biggest one. This will allow the population of data for each type of genre of music to be equal to the biggest one for a fair representation of each. From the counts before and after using oversampling, you can see the difference:

	count(*)	Genre
0	4340	Funk Carioca
1	18194	Hip Hop
2	37520	Pop
3	58040	Rock
4	12282	Samba
5	28319	Sertanejo

Unbalanced Dataset

	count(*)	Genre
0	58040	Funk Carioca
1	58040	Hip Hop
2	58040	Pop
3	58040	Rock
4	58040	Samba
5	58040	Sertanejo

Balanced Dataset

You can see from the screenshot that before the counts were very different for each genre of music, and after using oversampling the counts are constant at the largest group which was originally Rock music. Now that the data was finally balanced, we can apply this dataset to the same models using the same approach of splitting the data, vectorizing it, training and testing the models, and getting the accuracy scores/confusion matrix for each. Starting with the Naïve bayes model, we split the dataset using the train_test_split function, ran it against the

unigram Boolean vectorizer with a minimum document frequency of 5, and fir the model using the MultinomialNB function against the balanced training dataset. Then we fit the model and ran predictions to get a raw accuracy score of 68.18%. Compared to the unbalanced dataset, this accuracy seemed to increase by about 8%, and when we look at the precision, recall, and F1-score we see a greater detail of the score:

	precision	recall	f1-score	support
Samba	0.79	0.86	0.83	23151
Funk Carioca	0.89	0.57	0.70	23283
Hip Hop	0.62	0.23	0.34	23241
Sertanejo	0.56	0.79	0.66	23330
Pop	0.75	0.73	0.74	23007
Rock	0.60	0.91	0.72	23284
accuracy			0.68	139296
macro avg	0.70	0.68	0.66	139296
weighted avg	0.70	0.68	0.66	139296

From the screenshot we can see that the accuracies for all the genres of music for each category are much higher than the ones when we used the unbalanced dataset to do our analysis. For the precision measure, we see that Funk Carioca scored the best at 89%. For the recall measure, we see that both Rock and Samba music did the best with accuracies of 91% and 86%. Finally, when looking at the F1-score you can see that Samba ran the best at an 83% accuracy. One other note to make is that the supports for each of the genres of music are close to each other which means there was a fair representation of each genre for the training and testing dataset. Compared to the unbalanced data results, the supports were very skewed towards the ones with the biggest representation of data. When we look at the confusion matrix, we can see how the model predicted each genre:

	P:Samba	P:Funk Carioca	P:Hip Hop	P:Sertanejo	P:Pop	P:Rock
T:Samba	16767	609	4	5334	228	65
T:Funk Carioca	674	19954	56	2355	76	36
T:Hip Hop	997	2939	13298	1202	2236	2611
T:Sertanejo	1032	910	0	21132	164	46
T:Pop	1578	557	1128	2912	5429	11637
T:Rock	1270	231	406	2353	676	18394

From the confusion matrix you can see that Funk Carioca seemed to be the most accurate prediction as most of the guesses were correct. Also, it looks like Sertanejo was the hardest to predict for this model as all of the genres were guessed in the thousands for mistaking it for the actual correct label. The last portion of testing was to use the balanced dataset to predict the music genre using Support Vector Machines. The process was the same as before where we split the dataset, used the unigram Boolean vectorizer, and trained and tested the model using the LinearSCV function with a C equal to 1. After training and fitting the model to make predictions we saw an initial holdout accuracy of 87.18% which is substantially better than using the unbalanced dataset. It seems when using the SVM model on the balanced

dataset compared to the unbalanced one, it performed much better. When we view the precision, recall, and F1-score we can see further detail of the scores:

	precision	recall	f1-score	support
Samba	0.98	1.00	0.99	23151
Funk Carioca	0.93	0.95	0.94	23283
Hip Hop	0.71	0.70	0.71	23241
Sertanejo	0.76	0.69	0.72	23330
Pop	0.91	0.97	0.94	23007
Rock	0.91	0.92	0.92	23284
accuracy			0.87	139296
macro avg	0.87	0.87	0.87	139296
weighted avg	0.87	0.87	0.87	139296

From the scores you can see the model was very good at predicting most of the genres of music correctly using the SVM model. For the precision, the highest percentages consisted of genres Samba at 98%, Funk Carioca at 93%, and then Pop and Rock at 91%. For the recall measure, Samba was 100% accurate, while others slightly lagged behind them such as Pop at 97% and Funk Carioca at 95%. Finally, for the F1-score, Samba was once again at the highest measure of 99%, while Funk Carioca and Pop trailed behind at 94%. When we view the confusion matrix for it, we see a greater detail of the predictions:

	P:Samba	P:Funk Carioca	P:Hip Hop	P:Sertanejo	P:Pop	P:Rock
T:Samba	22375	25	15	370	154	68
T:Funk Carioca	9	23142	0	0	0	0
T:Hip Hop	49	43	22086	69	668	368
T:Sertanejo	937	122	40	21346	542	297
T:Pop	580	118	976	859	16349	4359
T:Rock	535	78	592	700	5281	16144

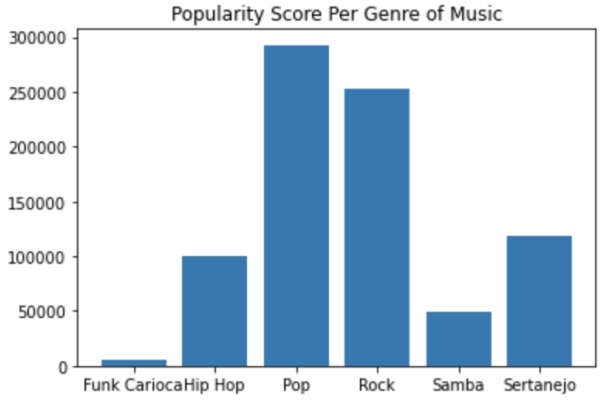
From the confusion matrix, you can see that Funk Carioca was rarely mistaken for any other genre of music besides its own. You can also see that Rock and Pop seemed to be the hardest to predict as they seem very similar to each other based on the matrix. Also, it looks like Samba was only predicted for Samba mostly and no other genre as you can see the times it mistook it for Samba when it was wrong was very low. Overall, it looks like using Linear SVM on the balanced dataset was by far the best model for predicting music genre.

Conclusion:

When viewing the dataset after preprocessing and vectorizing the data, it was apparent that there was some unbiased data where there were too many entries for Pop and Rock, and too few entries for Funk Carioca. When we viewed the results from both the Naïve Bayes model and the SVM model, we saw the accuracies were very low. However, when we applied oversampling to the data and reran the models, the accuracy improved dramatically by an 8-15% increase in score. From the data visualizations found from the data, there were some key.

Findings that can be used of value in the music industry. For example, when looking at a visual of the popularity of a song based on the genre of music, we were able to get an idea about the types of genres of music from these six, that are most popular. Using the sqldf function, we were able to get the sum of all popularity counts using the Popularity field, and group it by genre to get the following displays:

	sum_pop	Genre
0	5727.7	Funk Carioca
1	99648.4	Hip Hop
2	292820.5	Pop
3	252171.3	Rock
4	48876.2	Samba
5	118477.0	Sertanejo



From the visual we can see that when we add up the popularity scores for each song and group it by genre of music, Pop and Rock seem to excell against the other categories. This type of information can be used for a music company who is just starting to hire new and upcoming artists. They could use these visualizations to inform them to mostly hire new artists that are classified in a Pop or Rock genre of music. They could also be aware to avoid artists that work in genres that are of lower popularity. Information like this is essintial to a company, where there main goal is to hire artists that will generate the most amount of views and money.

From the models that we ran against the balanced dataset, we can see that the Linear SVM model excelled past the Naïve Bayes model in doing its analyiss and predictions. With the holdout accuracy of 87.18% that was reported from the SVM model and the accuracy of 68.18% from the Naïve Byaes model, it is clear to use the SVM one in the future. However, there were some similarities between the two models. For example, the supports were relatively the same since we were using a balanced dataset for analysis. Also, for the F1-score, Samba had the highest accuracy for both models as it was the most balanced between the precision and recall compared to the other genres. When comparing the confusion matrixes between both models, we can see that Sertanejo was the hardest to predict for Naïve Bayes and Rock was the hardest to predict for SVM. However, both models kept mistaking Pop for Rock and vice versa, as their correct and incorrect guesses were very close to each other. This could indicate that they use a lot of the same types of lyrics between one another as they seem to be realted. One last note to make was that the Naïve Bayes model predicted Hip Hop the best, while the SVM model predicted Funk. Carioca the best.

One reason for a music company to use these models would be new artists that do not declare themselfs to any genre of music. You could intake the songs they've created and use the trained models to predict the genre of music the songs mostly associate to. You could also build around those songs that lean towards populrity, and suggest them to make more of them

with that style. One future suggestion to the company would be to use this model and adjust it to make the accuracy better, so when they are intaking new songs, the model can predict better.

References:

Kaggle website: https://www.kaggle.com/datasets/neisse/scrapped-lyrics-from-6-genres?select=lyrics-data.csv