n0vah917 / dsc-phase-4-project-v2-3 Public

forked from learn-co-curriculum/dsc-phase-4-project-v2-3

View license

0 stars 3 forks

Star Watch

Code Pull requests Actions Projects Wiki Security Insights Se

main

This branch is 4 commits ahead of learn-co-curriculum:main. Contribute Sync fork

 n0vah917 Add files via upload ... now 18

View code

README.md

Classifying a Song's Genre Using Lyrics

Project Overview/Summary

The focus of this project is to provide a rudimentary tool for music recognition not through the song waveforms but through lyrics. This would be of value to companies like Spotify or Pandora, who can use this capability in order to improve the way people find music through their recommendation systems. Instead of searching by artist/era or album, a potential future recommendation system would be able to find a set of artists/songs who fit similar vibes/moods.

Lyric data from Ariana Grande, Eminem, Drake, Post Malone, Taylor Swift, Maroon 5, and Selena Gomez's entire discography was used in order to develop a NLP model that will be able to identify which song belongs to which genre. The genres for identification for this project would be Rap and Pop.

Stopwords were removed, and all tokens in the model were lemmatized in order to consolidate via root words. The TF-IDF (Term Frequency - Inverse Document Frequency) Vectorizer is used in conjunction with the Multinomial Naive Bayes classifier, in order to create a model that successfully classifies genre to a song.

Differentiating genre solely on a song's lyrics has massive capabilities when it comes to the future of music recognition. The resulting model was able to successfully assign a given genre to a lyric with an accuracy of 92% on training data and 88% on testing data. As a side goal, this project will be able to identify common themes and verbiage that artists tend to use.

Business Problem

An artist can be most understood through their lyrics and the messages they convey to their listeners. Generation defining artists like Kendrick Lamar and Frank Ocean captivate their audiences with poetic skill, painting vivid pictures within 16-bar verses. An artist's growth, both in musical talent and personal growth can also be observed their subsequent releases. As an example, the beginning of Drake's rap career started off with raw, unfiltered "backpack-rapper" influenced lyrics, heavily relying on his unique charisma as a defining feature.

To build off this example, day-one fans of Drake have noticed his lyrical style evolve throughout his 16-year long music career. From emotional tributes, to party bangers, to multi-genre experiemental tracks, themes that express sadness, fame, infatuation, and desensitization can be identified through his word choice. In terms of other artists, when I listen to a Frank Ocean song, I can expect to be sad for hours. When I listen to a Disturbed song, I can expect to be angry, or hyped up. When I listen to a Marvin Gaye song I can expect to learn about the world of injustice he saw through his lyrics.

With the plethora of music available nowadays, some might say the market is oversatured. From my personal experience, it can become increasingly harder to search for music that I actually want to listen when I'm in a given mood. Searching by an artist on Spotify is sometimes unpredictable/unfruitful, as artists like Drake have different phases of music theme and style throughout their extensive discography. While public users make playlists titled "Sad Jams" or "Gym Hype" on Spotify, more often than not, the music within these playlists doesn't typically align with when I'm actually looking to listen to. Even on some level, it would be interesting to see music search capabilities can be improved. Using something as rudimentary/simple as a search bar is antiquated at this point.

Lyric data spanning Ariana Grande, Eminem, Drake, Taylor Swift, Maroon 5, Post Malone, and Selena Gomez's artist discography through 2021 will be aggregated/summarized in order to classify a given song lyric text body to its corresponding genre. An NLP (natural language processing) model will be created that will be able to accurately identify each song's respective genre, as well as identify themes/choice words artists tend to use in the process. The output of this project would be a precursor to a larger project for music recognition/recommendation through the use of an artist's themes.

Data Understanding

Lyrics from the lyric aggregate site, Genius, were found pre-scraped from a Kaggle dataset. All packages for interpretation/modeling are imported in this step. Each row in the initial dataset represents an individual song, with its corresponding artist, song title, album title, year of release, and date of publishing. Upon observation, the song lyric text is represented in the Lyric column, as a long string separated by spaces and punctuation.

Across the 7 artists contained in the dataset, there are 2,137 total songs. The split of songs present each genre of interest are about even, 1,084 identified as Rap, and 1,053 identified as Pop.

While Rap and Pop are not always mutually exclusive in terms of style, I personally found these genres to be the most divergent within the ones that were present in the overall available dataset. Since the goal of this project is genre classification, it is important that the chosen genres don't have too much assumed overlap with lyrical content/subject matter.

Drake, Eminem, and Post Malone were identified as the Rap artists, and Taylor Swift, Ariana Grande, Maroon 5, and Selena Gomez were identified as the Pop artists.

On average, Eminem has 707 words per song On average, Drake has 499 words per song On average, Ariana Grande has 374 words per song On average, Selena Gomez has 335 words per song On average, Taylor Swift has 357 words per song On average, Maroon 5 has 350 words per song On average, Post Malone has 423 words per song

Now that general metrics regarding the dataset are identified for context, the dataset can now be prepped for modeling.

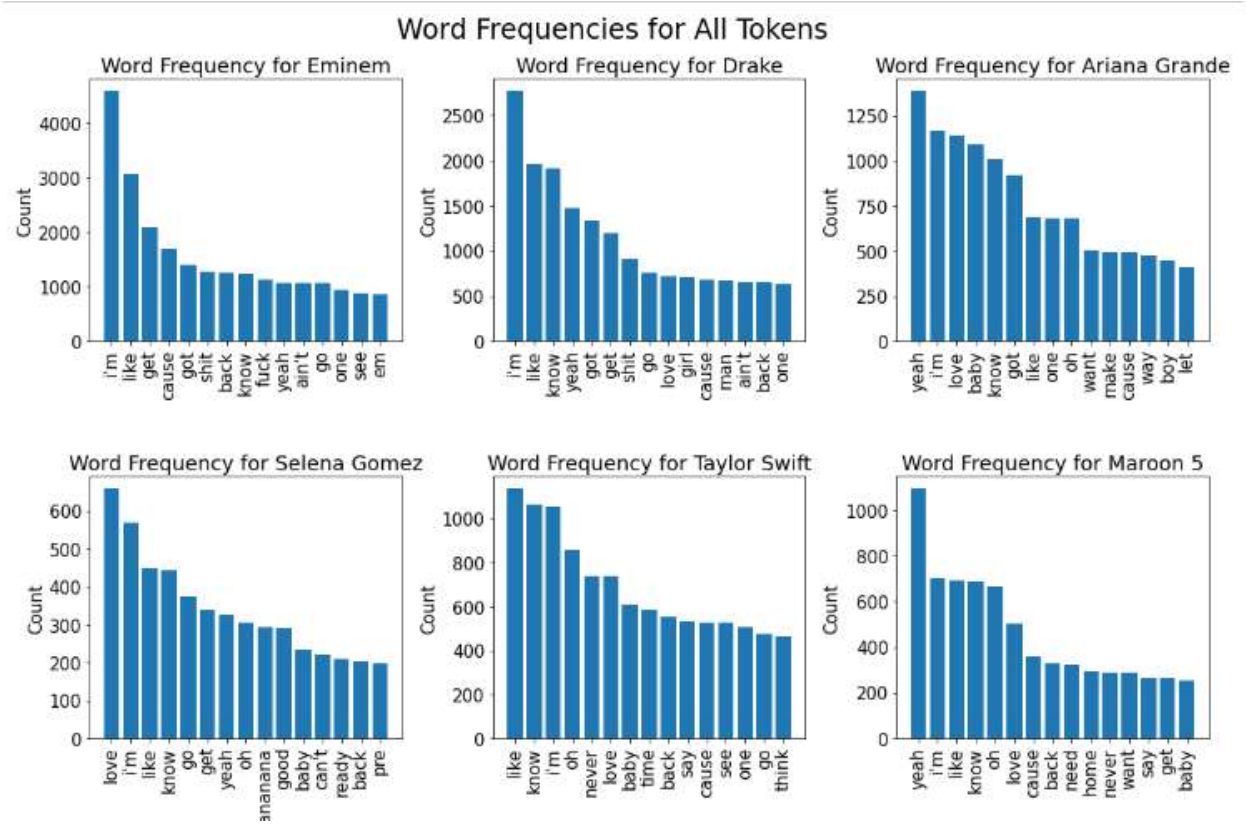
Data Preparation

The first step in prepping the dataset to be summarized is to break each lyric (currently stored as a text string in the overall table) into individual words, separating by commas, then re-populating as a list in each row in the dataframe. Below, the regular expression pattern is used within the tokenizer to split words, looking for non-letter characters as boundaries while preserving apostrophes within words where relevant. Success is observed due to row 6 still containing "i'm" as an individual word, rather than splitting at the apostrophe.

Prior to importing the dataset, it was observed that most words within the lyric text were already in lowercase, but to account for potential outliers, all words are converted in the following step. It would not be desirable to count 2 words as distinct if their only differing factor is case (e.g. "I'm" vs "i'm".)

In the following step, English stopwords were imported and a function was created to remove all stopwords within each list of lyrics. When doing a form of classification, it is important to remove stopwords as there are bound to be sets of words that are common amongst any body of text (e.g. "and","the). Keeping these words makes it harder for the eventual model to differentiate between two sets of text and make the proper classification.

In order to get a better sense of the data, now that the words have been tokenized and stop words have been removed, the following code is used to visualize the distribution of words within each available artist's lyrics. From the respective charts, it can be observed that there are still some commonalities amongst top words for each artist, even after stopwords have been removed. If these pose an issue, they can be removed in later iterations of modeling in order to improve performance, but for now, “i’m” and “like” are removed.



It is blatantly evident that within the rap artists (Drake, Eminem, and Post Malone), curse words are far more prevalent in their lyrics, while none are present in the pop songs. Within the pop songs, "love" seems to be a common word, taking a top 10 slot across all artists.

Doing a quick check amongst the distinct words that are present within the dataset, it seems as if removing stopwords caused the overall word count to drop to a great degree.

Now that the words are tokenized and prepped for modeling, a final count is done of the set of words present in each genre. Surprisingly, the count is about even between the two genres. It was expected that Rap would have a significantly higher amount of distinct words, considering their average word count was higher overall as well.

The Rap discography has 1080 distinct words The Pop discography has 1026 distinct words

Data Modeling

In order to initialize modeling, the data is split into train and testing groups so that the models can be evaluated in the presence of new data.

Model 1: Count Vectorizer + Multinomial Naive Bayes

In order to further prep the data, a count vectorizer is initialized. Vectorizers are necessary for this process, as it provides a way to convert text data, currently in the form of lists, into a form that a model can understand. The Count Vectorizer is a rudimentary vectorizer, which solely counts the presence of a word in any given document. The min_df is specified as 30, an arbitrary number for our baseline model. The min_df value of 30 implies that all words in the vectorizer need to be in the document base at least 30 times. Similarly, the max_df is specified as 1000, which implies that words are only considered as identifiers for the model if they appear in less than 1000 distinct documents. Words that appear too frequently across the base are not desirable, as their inclusion may decrease resulting performance.

]:

	able	across	act	actin	acting	actually	admit	afraid	age	ago	...	ya	yeah	year	years	yes	yet	yo	york	young	youre
686	0	0	0	0	0	0	0	0	0	0	...	1	2	0	0	1	0	0	0	0	0
394	0	0	0	0	0	0	0	0	0	0	...	0	5	0	0	0	0	0	0	0	0
816	0	0	0	0	0	0	0	0	0	0	...	1	1	0	0	0	0	0	0	1	0
674	0	0	0	0	0	0	0	0	0	0	...	1	17	0	0	1	0	1	0	0	0
39	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	0	0	0	0	0	0

Below, a Multinomial Naive Bayes model is instantiated for our model to be fit on, and the plotted confusion matrix helps visualize how well the model did. The MultinomialNB model is essential for text classification, as it involves using conditional probability across the different categorizations, and is generally good for classifying on distinct features.

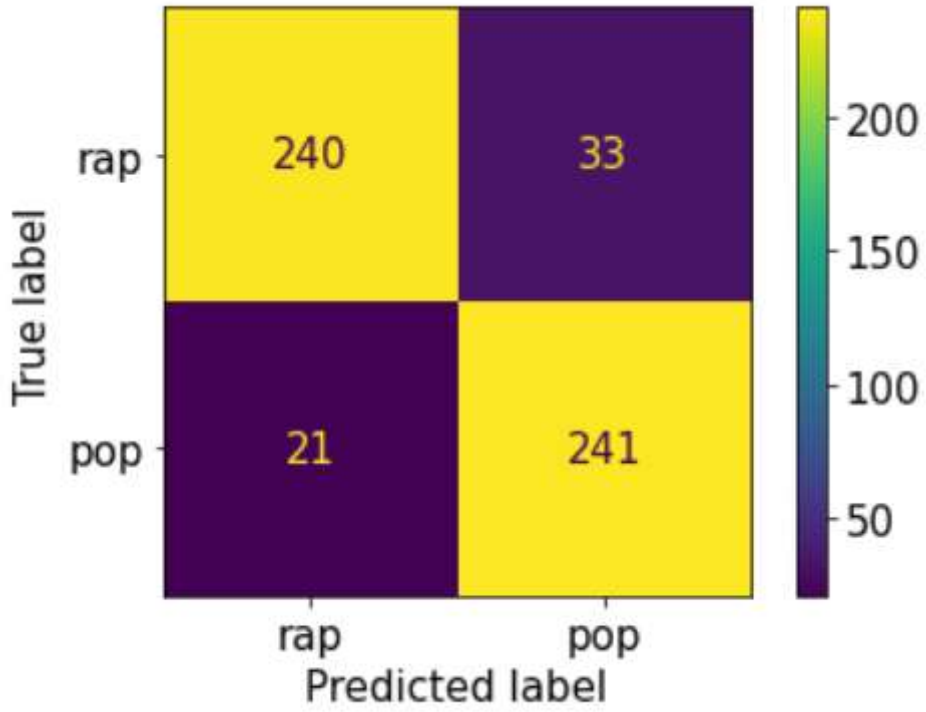
The corresponding confusion matrix, seen below, shows how the model classifies the test/train data using the Multinomial Naive Bayes classifier through counts of true/false positives/negatives.

LEGEND: Rap in True label, Rap in Predicted label - True Negative

Rap in True label, Pop in Predicted label - False Positive

Pop in True label, Pop in Predicted label - True Positive

Pop in True label, Rap in Predicted label - False Negative



Below, results of the first model can be seen through the classification report. The classification report outputs 3 metrics of critical importance: precision, recall, accuracy, and f1 score. These main metrics help understand how well the model performed in terms of classifying each song genre as a rap/pop.

Precision: Precision measures how precise the predictions are. Precision is calculated by taking the number of true Rap classifications, and dividing by the number of Rap song classifications predicted by the model.

Recall: Recall measures what % of the games identified as Rap were actually identified correctly. Recall is calculated by dividing the number of total true Rap classifications identified by the model, by the number of actual Rap songs.

Accuracy: Accuracy evaluates all predictions within the model. Accuracy is calculated by dividing the total correct predictions from the model (both Rap/Pop) by the total records in the original dataset.

F1 Score: F1 Score takes both precision and recall into account to get a cumulative score. The formula is calculated as follows: 2(precision recall)/(precision + recall). This metric is a good measure of overall model performance, as a high F1 score implies both precision and recall are both high as well.

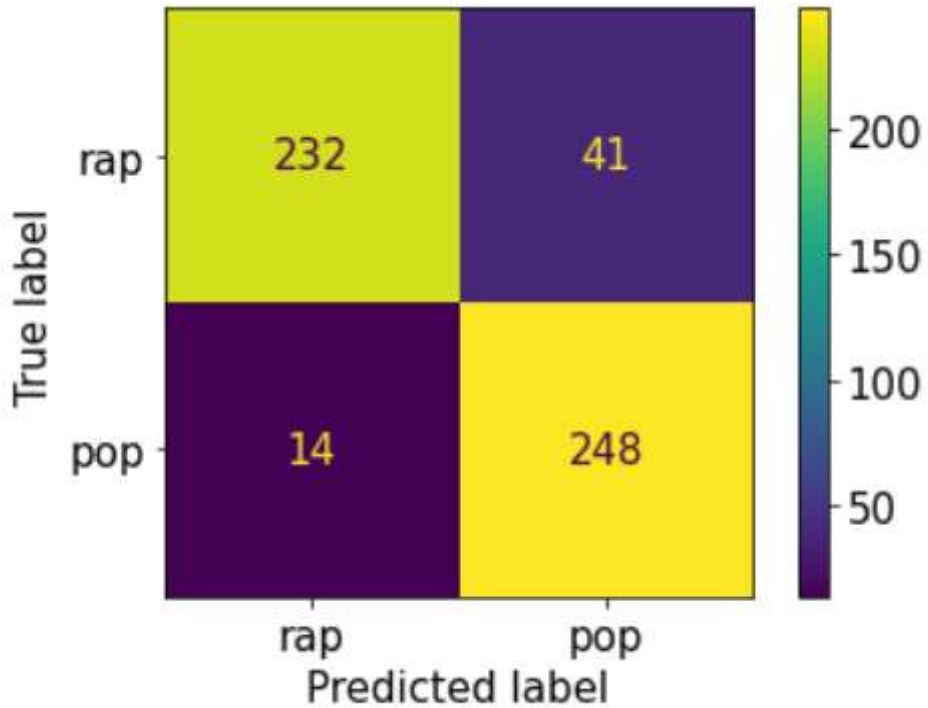
As seen below, the initial model performs relatively well on the test data, seen from the predicted values using the trained model. F1 score is relatively high for both Rap/Pop classifications, implying succesful precision and recall values as well. Looking at the overall accuracy, across all observations, our model performed exceptionally well too.

Model 2: TF_IDF Vectorizer + Multinomial Naive Bayes

For the second iteration of the model, since the Multinomial Naive Bayes classifier seems to be an ideal fit for this type of text data, the vectorizer will now be changed in order to see if model performance will improve. This time, a TF-IDF vectorizer, also known as Term Frequency-Inverse Document Frequency, will be used. This type of vectorizer is ideal for use in the case of multiple text documents, and due to the nature of the calculation, it implies rare words contain more information than common words. Term Frequency (TF) represents the presence of a word in a document as a fraction, while inverse document frequency (IDF) represents the prevalence of a word across all documents. By multiplying the two, a value for the importance each word has in the entire document base is obtained.

Similarly to the Count vectorizer, the min_df and max_df parameters will be kept the same in order to compare similar setups.

As seen through the slightly improved recall scores within both the test data and training data, we can see evident model improvement with the use of the TF-IDF vectorizer.



Model 3: TF-IDF Vectorizer + Random Forest Classifier

Keeping the TF-IDF vectorizer, as it has been identified to have a clear improvement over the Count vectorizer, another classifier is tried for the sake of curiosity, the Random Forest Classifier. This model works by using several decision trees in order to make a classifying decision. In prior projects, the Random Forest was seen to have relatively good performance on determining game outcomes for the NBA.

As seen in the following classification report, recall of Rap songs went down to a great degree, showing that the application of this new model tended to favor the Pop song classifications, across both the test and training data. The Random Forest Classifier will not be considered for the final model iteration, and the MultinomialNB model will be continuously improved upon.

Model 4: Lemmatizing and Removing Additional Stopwords - TF-IDF and Multinomial Bayes

As seen in prior word count distributions across the artists, it was observed that some words like “get” and “got” were considered as distinct words although semantically, they have the same meaning. In an effort to improve prior model performance, words will be lemmatized in order to consolidate based on their root words. Below, the part of speech is tagged to each token word, and the root words of each are found. Additionally, more common words amongst the genres are included in order to decrease similarity amongst the genre sets. Word frequencies are revisualized, each genre/artist looking more unique.

Model performance decreased slightly in terms of recall, possibly due to the recategorizing of words due to lemmatization. Overall word count dropped ~100 from consolidation. Similar to the Random Forest model, slight skewing towards Pop classification is observed. Consolidating "get" and "got" may have decreased performance, as those words are individually clear identifiers of Rap songs, rather than Pop songs. There is a slight preference for this model, even though it decreased performance, as now all root words are properly classified/identified.

Model 5: Gridsearch on Random Forest Classifier - TF-IDF

In an effort to improve model performance for the Random Forest classifier, a grid search is used in order to identify potential parameters, to see if application would outperform the Naive Bayes model. Results as depicted below show that while falsely identified Pop songs were mitigated a bit, and overall recall performance improved from the initial Random Forest model, the Naive Bayes model still outperforms the Random Forest model.

Classifier Results Evaluation

When considering a matter as sensitive as music genre, the distribution of words chosen across artist's song lyrics have a large impact on how their music is perceived and recognized by cultures and communities. The final model created, using a Multinomial Naive Bayes classifier with a TF-IDF vectorizer, was able to predict song genre given song lyrics with an accuracy of 88% for test data and 92% for training data.

A classification report and confusion matrix of the classifier model can be seen below across the test/train datasets.

For the test data, recall was far higher than precision for Pop predictions, which implies the model was better at correctly selecting true Pop songs, rather than predicting whether a song was truly Pop/Rap. On the contrary, recall for Rap songs was far lower. This means the model tends to favor assigning false positive Pop songs. The tendency to designate false positive Pop songs can be overlooked since the recall % of overall genre predictions were high, which is the main target metric of success within the model.

For the training data, similar trends can be seen within precision, recall, and f1-score for Rap/Pop predictions, but the model was slightly better with the data it was trained on. Both test/training predictions having ~90% accuracy implies that there is negligible overfitting on the final model.

Below, the consolidated/lemmatized data for word distributions across the artists in the dataset can be observed. Model performance being largely successful can be rationalized by simply looking at the evident difference in word composition between artists.

Observed Profanity

A common thread linking most rap music since its inception in the late 80s/early 90s is the prevalence of profanity. For the rap artists included in the data set this is evident, especially with Eminem, who is notorious for being a "NSFW" artist. A large part of this tendency may be holdover influence from the "gangster rap" movement in the 90s, music containing themes of violence/perceived vulgar subject matter. Between the rap and pop genres, it is undeniable that the prevalence had a large effect on the classifier's success. Pop music is generally perceived as family-friendly, radio-friendly music that is palatable for all ages through all mediums, hence its popularity. While profane words took several of the top 15 slots across the rap artists, no profanity was seen in any of the pop artist lyrics.

Perceived Gender

In hindsight, gender did have an inherent influence on the success of the results. The rap genre was solely comprised of male artists, and aside from Maroon 5, the pop genre was solely comprised of female artists. While music should ideally be universal, where anyone regardless of gender can relate on some level with a song, the word frequency shows an evident disconnect from this idea. This is to be expected, as artist write music about their own personal experiences, one of the most prevalent topics being about heartbreak/love. In the male rap artists (who are known as heterosexual), some songs can be seen directed to/referencing women; the use of female-oriented words like b**** (while problematic) and girl are seen in all rappers' top slots. Similarly in pop songs, words like baby and boy and love are far more prevalent. This inherent bias seen through gender likely had a great effect on the successful model classification.

Recommendations/Conclusion

The model was able to successfully assign the majority of songs within the testing dataset to their respective listed genres, solely through the use of song lyrics. While an 90% accuracy across both test/training data may be seen as a poorly performing model when considering perceived blatant differences between artists, there is a bit of overlap in subject matter between mainstream Pop and Rap songs especially when considering the artists in the dataset.

Some may say the line is blurred for rappers like Post Malone and Drake, who are frequently observed experimenting with new instruments/lyrical styles in their music in an effort to appeal to wider audiences. Similarly, Pop artists frequently have features by Rap artists in their recent songs, examples of cross-genre collaboration seen with Katy Perry & Snoop Dogg, Adam Levine and Kanye West, BTS and Nicki Minaj. More definitive/improved model performance may be seen when using data sourced exclusively from 90's music, when the genres were observed as largely being mutually exclusive.

The findings of this model show that there is inherent correlation with lyrical subject matter/composition and perceived genre. Regarding the stakeholders, Spotify and Pandora, genre recognition through lyrics has massive implications on how music recommendation can be reworked/improved. The ability to identify commonalities through songs that have similar lyrical themes can expose users to music that truly fits their mood, improving customer experience and retention. While this project was a very rudimentary look into a new lens of song classification, the results and framework of this model could be extended to a new form of general music classification/song recommendation. Since the experience of listening to music is so fundamentally nuanced from person to person, being able to succesfully recommend exactly what someone is looking for has limitless applications for the future of music streaming.

Next Steps

Inherently, subject matter, cultural shifts, and pop culture have changed drastically within the last 30 years, largely due to the advent of the internet. It would be interesting to see this model performed on music lyrics from the 90s, when the rap and pop genres were widely seen as distinctly different. From a broader perspective, the artists selected within this dataset can all be classified as mainstream pop music to some. Outside of the lens of rap and pop, different genres could be also compared, such as country and soul, to see if there may be inherent overlap/differing characteristics.

Another route of improvement would be to classify using bigrams as a supplement. This factor was not addressed during the final iterations of the model as the words involved, regardless of context, were distinct enough to make successful classifications, seen by the final performance. Tying this back to the prior point regarding gender, it would be interesting to see the context in which women are talked about in either genre, and if implementing those specific bigrams would improve performance further.

Another route of improvement would be to think about genre from a wider perspective. Similar to the point brought up in the conclusion, This would involve information-gathering to understand public sentiment towards a set of songs. Being able to classify songs based on the induced feeling within a listener would improve current recommendations systems to a great degree. Additional columns describing mood could be created in the dataset and implemented in a new model.

Releases

No releases published
Create a new release

Packages

No packages published
Publish your first package