Genre Prediction From Lyrics

By Anjaneya Teja Sarma Kalvakolanu, Nagasai Chandra and Bao Nguyen

I. Project Description and Breakdown

A. Project Description

They say Rock and Roll's lyrics are generally upbeat and positive, while country music lyrics are depressing. There should be a relationship between a song's lyrics and its genre and that what we want to explore. In this project, we want to build a tool to help users get the genre of a song if they input the lyrics of that song. Also, our system can automatically generate the reasons why it chooses that genre. On top of that, based on the dataset that we have, the system will recommend several songs related to the lyrics that users have given.

B. Project Breakdown

There are five main tasks that we need to solve in this project:

1. The Data: Cleaning and preprocessing the data "380,000+ lyrics from MetroLyrics." from kaggle.com.
2. Genre Classification: Extracting features from the lyrics of the dataset then training a classifier to predict song's genres.
3. Reason Generation: Automatically generating the reasons explain why our system predicts that genre
4. Song Recommendation: Finding the top songs that have lyrics similar to the input's lyrics.
5. User Interface: Using Django to build a web interface that allows users to easily interact with our system.

II. Tools and Libraries Used

NLTK library: Pre-processing the data

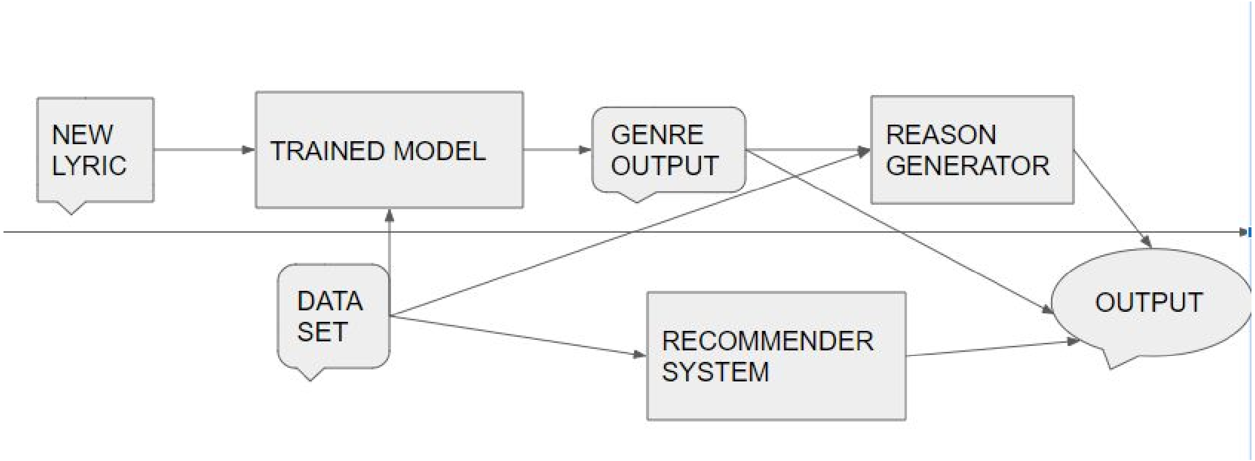
Keras library: Extracting TF-IDF, training model

Django library: Building the user interface

Pandas library: Reading, writing, visualize data

SKLEARN: Extracting TF-IDF vectorizer and cosine similarity

III.System diagram



A model is trained with the Dataset

1) When a new lyric is entered the trained model predicts the genre.

2) The predicted genre along with the dataset is used by the reason generator to generate support for the prediction.

3) The recommender system uses the data set to give a list of songs similar to the existing lyrics of the given song.

4) The output is a combination of all the above three aspects.

IV. Implementation

A. Data Preprocessing

For this project, we use a dataset from kaggle.com “380,000+ lyrics from MetroLyrics” of user GyanendraMishra [1] which includes 362,237 songs from 18,231 artists with six columns for each song: song index, title, year released, artist, genre, and lyrics. This dataset has a total of 10 different genres plus “Not Available”, “Other” genre. Many songs in this dataset don’t have lyrics included or their lyrics are not in the English language. Also, there is a heavily skewed in the number of songs in different genres, therefore, there is quite a lot of work that needs to be done in preprocess the dataset.

1. Removing Non-English songs

Since not all the songs in this dataset are English songs. therefore, there could be misleading when we analyze their lyrics since words are spelled correctly the same but from different languages can have a different meaning or words have the same meaning but spell differently in different languages. To removing the non-English songs, we use the “Stopword” corpus of NLTK to decide if a lyric is in English or not; basically by comparing the number of English stop words and the number of non-English stop words in this lyric. This algorithm helped us remove about 14,000 non-English songs.

2. Removing songs missing information

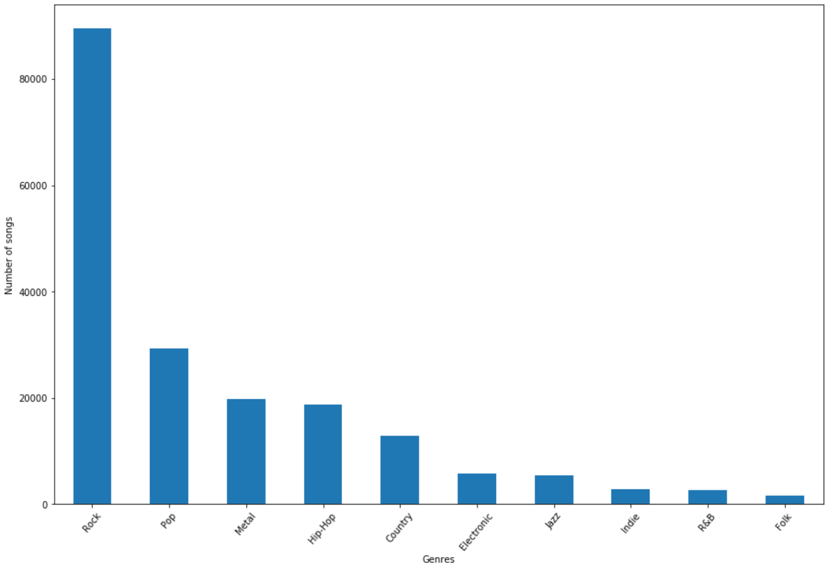
Since we focus on predicting the genre of a song based on its lyrics so we decided to remove songs with the genre of “Not Available”, “Other” or empty. We also excluded songs with empty lyrics or their lyrics are duplicated with other songs in this dataset. On top of that, we also remove the songs with the size of the lyrics are less than 200 characters since those lyrics don’t provide enough information for training. After this phase of the cleaning, our dataset now contains only 188909 songs.

3. Tokenize, Standardize, Lemmatize and Removing Stop Words

At first, we use the NLTK library to tokenize each word in the lyrics which are necessary for the upcoming steps of cleaning including standardizing the format of the lyrics which we transform both lowercase and uppercase words to lowercase and remove all special characters. Some words in the lyrics are also in different forms but refer to the same thing such as love, loved or loving, so that we do the lemmatization by converting all of them to the based form. And finally, we also remove all the stop words since they are extremely common and don’t provide valuable indicators of meaning or sentiment; therefore, they are not valuable for training.

4. Balancing the data

The above figure shows the number of songs for each genre. As you can see, The dataset was heavily skewed towards Rock while some other genres like Folk, R&B or Indie occupy only small parts of the dataset. This could lead our trained model to predict Rock more than other genres, especially when comparing to Indie, R&B or Folk. Therefore, after reserving 400 songs of each genre for the validation set, we



performed Oversampling and Undersampling of Sklearn library on each genre to a size of 15,000 songs.

B. Genre Classification

1. Extracting the features

After cleaning the data, we need to extract the features from those lyrics to begin training the model. We want to convert the raw text to relevant numbers which can present the information in the lyrics. We chose to present the lyrics of a song by the TF-IDF vector since it shows how important a word in a lyric, also adjusting some common words. We also extracted the TF-IDF of the song titles to see if it can be used as a feature of the training.

2. Training Model

Based on our research, we decided to train on Naive Bayes classifiers as the base since they are known to perform well in document classification and text analytics. We also tried to fit a logistic regression and a Neuron Network to see if we can get better performances. We also using Grid search with cross-validation to tune the hyperparameters of those models. Below is the accuracy of the validation set for different models and types of features.

	TF-IDF Lyrics	TF-IDF Title	TF-IDF Lyrics + Title
Multinomial Naïve Bayer	0.4235	0.278	0.3922
Logistic Regression	0.4257	0.2742	0.404
Neural Network	0.414	N/A	N/A

From the above results, the best accuracy that we could get is 0.425 which was trained by using logistic regression just 4 more better comparing to random prediction. We guess maybe the lyrics of a song and its genre are not highly correlated and there are some genres are similar in lyrics so the model does not perform well. For the purpose of the project, we decided to use the model was trained by Multinomial Naïve Bayer because of its performance since the accuracy looks similar between those methods.

C. Reason Generation

For this project to support our prediction, we generated a few supporting features that are

1. The number of words in a song of each genre
2. The length of a line of a song of each genre
3. No of unique words in the lyrics of a genre
4. No of unique words in a line of lyrics of a genre
5. Length of the punctuations
6. No of pronouns
7. No of nouns
8. No of verbs
9. No of Adjectives
10. The most repeated common words in a genre

For a new entered lyric the same values are calculated and are matched to the closest values generated for the whole dataset and the genre is mapped. If the genre is matched then the support is displayed or else that feature is ignored and the ones that support our prediction are generated.

D. Song Recommendation

As the dataset we have is enormous, we have decided to first sample the dataset before determining songs similar to the given lyrics. The key concept used in determining features to find similar songs is to get the Term Frequency - Inverse Document Frequency on the lyrics as documents. In these steps, we have considered n-grams of range(1-3) and made sure to exclude stopwords. With these features Cosine Similarity metric was used to determine the similarity indices of all songs in the sampled dataset. The webpage displays upto 5 similar songs with comparatively highest similarity measures including their scores.

E. User Interface

For the user interface, we used the Django framework that works on the principle of Model View Controller. The Django app is created which contains Url, views, templates and other files.

A URL is generated for the home page and the view is written in python for backend processing, this view returns the data processed into an HTML file and the output is displayed in the HTML.

When the submit button is clicked a new URL is triggered which in turn triggers a new view that processes the data to obtain the predictions support and recommendation results that are displayed on the HTML page.

A navbar is created to provide easy access to the home page to perform prediction for another lyric.

V. Conclusion and Future thought

The project is successful in predicting the genre , reasoning about the prediction, recommending the similar songs with similar lyrics. We used naive bayes , linear regression and neural network to predict the genre for classification. But it turns out to be very low correlation between the lyrics and the genre as well as the title of the song and the genre. To take this work forward, we first should collect a better dataset with very distinct genres. Also, using Convolutional neural networks that extract their own features for prediction which can give a better accuracy.

Appendix (Sample Outputs): This is in a separate document called genre_prediction_from_lyrics_appendix.pdf

References:

[1] “380,000+ lyrics from MetroLyrics” - GyanendraMishra here.