NATURAL LANGUAGE PROCESSING

Genre Classification using Song Lyrics Data

Rohan Chitte

Abstract

Majority of the commercial music platforms rely heavily on deep learning and natural language processing for the purpose of finding similarities between songs, classification of songs, new lyric generation, informational retrieval, gaining meaningful insights or efficiently analyze music. My main focus was to use raw lyrics data and classify them as one of the 5 genres (Hip-Hop, Rock, Pop, Country and Jazz). For this purpose, architecture such as Machine learning classifiers, Deep Neural Networks, LSTM and Bi-Directional Transformers were investigated. Various Data preprocessing and network optimization techniques were utilised and While, It was observed that keras BERT model and LSTM produced similar results, BERT worked the best than the rest.

1 Introduction

Music Industry is constantly evolving and so are its listeners across the globe. Very large no. of new tracks are released everyday. With such huge data, it becomes a complication for music streaming platforms such as Spotify or Gaana to provide music recommendations to its users. With progress in Machine Learning and Natural Language Processing techniques, music platforms rely heavily on these algorithms to ease the complication of making recommendation.

This research project is based on improving the scores of a lyric based genre classification, which in turn could help provide recommendations if no metadata is available.

2 Dataset

MetroLyrics Dataset, found in Kaggle's database library, is no longer available. Therefore, Metro Lyrics Dataset Gitlab Link was used as an alternate for this project. The data consists of 362237 song lyrics and its corresponding genres along with some additional features such as year and artist name.

3 Data Preprocessing

For the purpose of genre classification based on raw lyrics, I considered Lyrics and Genre as 2 Main features. All the rows containing null values were removed from the data set using notnull() function of pandas. There are total 12 unique genres: Pop, Hip-Hop, Not-Available, Others, Rock, Metal, Country, Jazz, Electronic, Folk, RB and Indie. Since the dataset was unbalanced (see Figure 1) and few of the genres like Not-Available and Others contained Non-English words and Non-ASCII Characters, only 5 genres were selected for the classification task. .These 5 genres are: Pop, Hip-Hop, Rock, Country and Jazz. String manipulation methods were used to remove punctuation and song-related identifiers like [Chorus] or "instrumental".

Nltk's predefined Stop words were used to remove stops words from lyrics. In addition, the text was lemmatized using nltk's lemmatizer. The cleaned version of the dataset was called df and it contained 185493 rows and 2 columns.

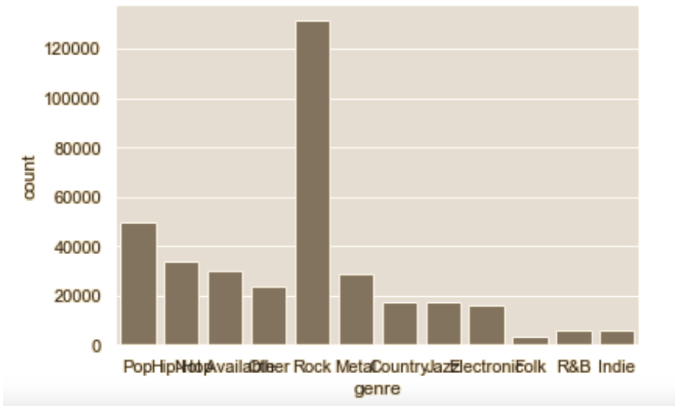

Figure 1 Data Histogram

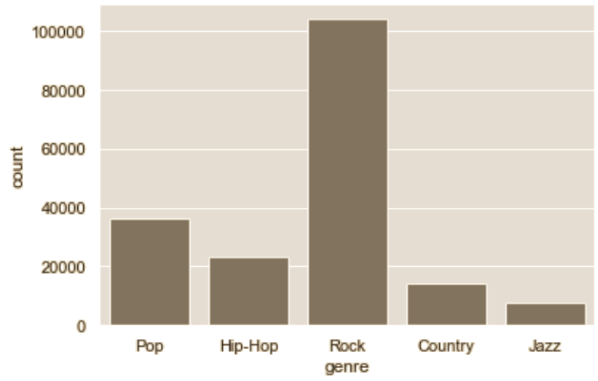

Figure 2 Data Histogram after cleaning

4 Approach

4.1 1. Machine Learning Algorithms and TfidVectorizer

Notebook Link

To get a quick overview of how well some of the conventional machine learning algorithms perform, MultinomialNB, RandomForestClassifier and a Dense Keras Network was used on the dataset. The data was split into training and validation set with 0.67:0.33 ratio using sklearn's train_test_split method. One hot encoding was used to convert genres into distinct numerical values. Next, Sklearn's TF-IDF Vectorizer was used to compute tf-idf scores on lyrics data and to fit the data in the machine algorithm for prediction. So, the training vector had the shape 124280 X 204679 and test vector had the shape 61213 X 204679.

4.1.1 MultinomialNB

Sklearn's MultinomialNB with alpha rate of 0.03 was used to fit the model. This model made predictions with accuracy score of 0.66%.

4.1.2 RandomForestClassifier

Sklearn's RandomForestClassifer with 300 no. of estimators was used to fit the model. This model made predictions with accuracy score of 0.69%.

4.1.3 Neural Network

Keras Sequential Dense network with 1 input layer, 1 hidden layer with 64 neurons and 1 output layer was used with relu and softmax as activation function. Categorical Crossentropy and adam were used for the loss function and optimizer respectively. This model made predictions with accuracy score of 0.68% after 5 epochs with batch size of 128.

4.1.4 Conclusion

While RandomForest Classifier and Neural Network had similar accuracy scores of roughly 68-69%, The Neural Network's training loss decreases and the validation loss increases for the neural network. So, it tends to over fit and is not able to generalise well.

4.2 2. Glove Embeddings with LSTM

Notebook Link

To reduce the cost of training a new deep learning model every time, it is efficient to use a Pre trained word vectors. For this project, I used Glove's 6 Billion tokens and 300 Dimensions Model called glove.6B.300d. Glove builds word embeddings in a way that a combination of word vectors relates directly to the probability of these words' co-occurrence in the corpus. Before loading the pre trained model, keras's tokenizer() was used to convert text into integer sequences and pad_sequences was used to prepare sequences of similar length.Next, the pretrained model was loaded and an embedding index was created.For every word, a weight was assigned using the embedding index and an embedding matrix of size 220815 x 300 was obtained for the training purpose.

Long Short Term Memory networks (LSTMs) are a special kind of Recurrent Neural Network (RNN), capable of learning long-term dependencies. Many research showed that LSTM works better than the traditional RNN. So, I decided to use Deep network that comprises of a Glove's Embedding layer fed to a LSTM cell with 128 neurons and a dropout rate of 0.2 to prevent over fitting. Global MaxPooling layer to extract less dimensions. Finally a Dense layer with 64 neurons with relu and an output layer with 5 neurons with softmax activation function. Categorical Crossentropy and adam were used for the loss function and optimizer respectively.

4.2.1 Hyperparameter Tuning

The biggest challenge is the hyper parameter tuning because there are many kinds of hyper parameters such as batch size, drop out rate, layer neuron size, activation function, cost function etc. Because, the dataset was huge, a lot of computation

effort and space was required for the training task. For ex: This lstm model took 5 days to run 5 epochs. Therefore, I decided to set the hyperparameters according to the conventional wisdom that shows a certain choice of results in model tends to not only generalize better results but also converge faster. Keeping the network simple and not too dense was ideal for this project to avoid over fitting and for faster training. Batch size was kept not too small or too large. Adam known for its faster convergence and overall good performance was selected. A grid search approach was attempted but keeping time constraint in mind, it was not feasible and hence aborted.

4.2.2 Results and Conclusion

Confusion Matrix Notebook Link

As expected, the LSTM model with pretrained word vectors significantly improved the performance of the dense network. After 5 epochs, the model gained an accuracy of 70.45%.(see Figure 3) From the confusion matrix, it was observed that the model was able to make good predictions for Rock and Hip-Hop where as the predictions made for jazz and pop were only about 25 and 35 percent accurate respectively. This is partly because of the limited training data available for these genres. Overall, the training loss and validation loss decreases with a minimal gap and stability between the two final loss values suggesting a good fit learning curve. However, a prolonged training of this good fit might lead to an overfit.

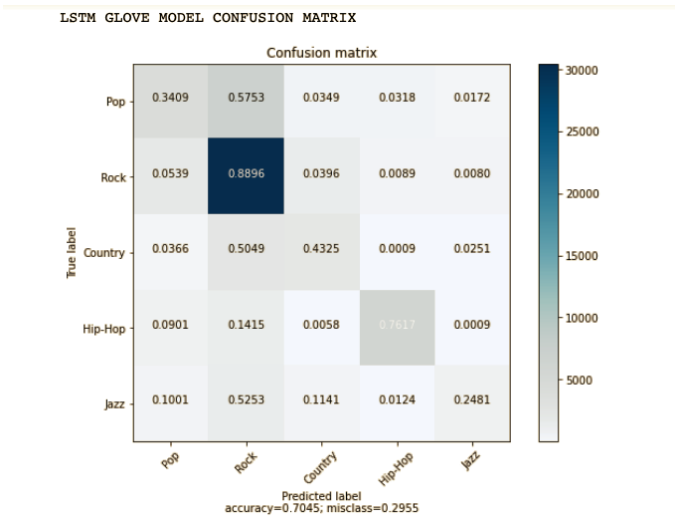

Figure 3 Confusion Matrix of LSTM with GLOVE word embeddings

4.3 3. Bert Embedding

Notebook Link

BERT is one of the leading NLP models in 2021. It is widely used for answering questions, named entity recognition and other tasks related to general language understanding. Bert has helped push the benchmarks for various projects in the

past and therefore for this project a compact bert model with 4 hidden layer and 512 neurons was chosen. These models are intended for environments with restricted computational resources. This bert model has been pre-trained for English on the Wikipedia and BooksCorpus and a pre-processor that is compatible with this BERT encoder was chosen. A 0.1 drop out layer, CategoricalCrossentropy loss function and adamw optimiser which is an improved version of adam was used with training epochs = 5+2 were used as hyperparameters.

4.3.1 Hyperparameter Tuning

Hyperparameter tuning challenges remains similar to LSTM Model, therefore the most conventional wisdom that shows a certain choice of parameters resulting in an overall good performance was chosen. A grid search approach was attempted but keeping time constraint in mind, it was not feasible and hence aborted.

4.3.2 Results and Conclusion

With Bert model, the classification accuracy for the validation dataset improved by 0.85% compared to LSTM model. After 5 epochs, the model gained an accuracy of 71.38%. The model was trained for 2 more epochs but the results didn't improve as the model started to overfit. Overall, the training loss and validation loss decreases with a minimal gap and stability between the two final loss values suggesting a good fit learning curve.

4.3.3 Improving Results with Data Augmentation

Notebook Link

Imbalanced data is often a very difficult challenge. Data Augmentation is one approach to amend lack of data. Nlpaug with a bert model and hyperparameter aug max = 3 was used to add more training data for jazz genre. Nearly 12000 artificial jazz song lyrics were added to the pre existing training data. The final training data had 160368 song lyrics. Histogram was used to visually spot the increase in the data. Evidently, the training and the validation accuracy both increased with increase in data. Because, the accuracy for the unseen data is more important, it was observed that the accuracy had increased from 71.38 to 71.58% which is significant 0.20% hike.

5 Final Classification Conclusion Table

Model	Best Accuracy Achieved
Multinomial Naive Bayes	66.60%
Random Forest Classifier	69.16%
Dense Neural network - 1 Hidden layer	70.14%
LSTM + Glove Embeddings	70.45%
BERT	71.38%
BERT + Data Augmentation	71.58%

Table 1 Accuracy Table

Project Github Link

Comparing the higest achieved accuracy of 71.58% with some of the benchmark scores found on the web for the Metrolyrics dataset. It is apparent that bert with data augmentation has attained and improved the benchmark scores.

Referred Benchmark Score Link 1
Referred Benchmark Score Link 2 - 69%
Referred Benchmark Score Link 3- 71%

5.1 Additional Data Analysis

Notebook Link

5.1.1 Word Cloud

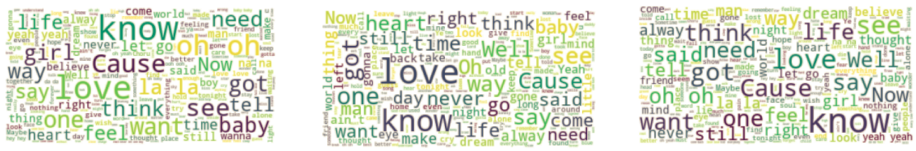

Figure 4 Pop — Country — Rock

Figure 5 Hip-Hop — Jazz

From the word clouds as one would expect, words such as "Love","life","Know","Heart" were most widely used for almost all the 5 genres. However, an interesting observation can be made in the Hip-Hop genre that although some of the words mentioned earlier occurred frequently, a lot of words suggesting emotions of anger, hatred and overall negativity is visible.

5.1.2 Semantic Analysis

NLTK's sentiment package was utilised in order to detect the influence of each genre on sentiments.

The figure (see Figure 6) confirms the analysis from word cloud that hip-hop is inclined towards more negativity than the rest of the genres. While rock promotes some level of negativity, jazz is the most positive kind. Country and Pop have similar effect and motivate more positively than negatively.

References

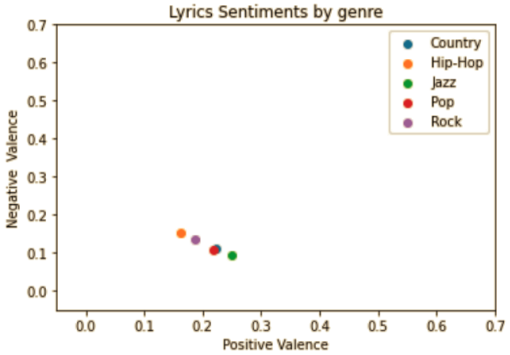

Figure 6 Sentiment Analysis of lyrics based on different genres