# Music Genre Classification by Lyric Analysis

Purvank Patel (A20380792) ppatel104@hawk.iit.edu

Kartik Prakash (A20376570) kprakash1@hawk.iit.edu

Sarthak Anand (A20389087) sanand13@hawk.iit.edu

## Abstract

Song lyrics show very specific properties such as rhyming verses and having different frequencies for certain parts of speech. Each genre possesses properties specific to its genre.
This is where Lyric Analysis can be used to take advantage of those properties to efficiently classify songs by their genre. In our project we focus on developing a classifier that classifies songs only on the basis of its lyrics. By analysing our dataset, we worked on developing specific features that would help our classifier to distinguish between songs of different genres. After feature selection, we implemented various machine learning algorithms such as Logistic Regression, Neural Networks, Gradient Boosting and k Nearest Neighbors on top of these features. We achieved interesting results and also understood a lot about the semantics of different genres by working on this project.We also experimented with several other classifiers such as RandomForestClassifier, Naive Bayes etc but did not report the results for these classifiers.

## Introduction

Classification of music is a very important and heavily researched task in the field of NLP. Previous research in this field has focused on classifying music based on mood, genre, annotations, and artist. All the approaches either used audio features, lyric as text or both in combination.

Genre classification by lyrics is itself a clear Natural Language Processing problem. The end goal of NLP is to extract some sort of meaning from text. For music genre classification, this equates to finding features to classify music using lyrics.

There are a wide range of scholarly and commercial applications for automated music genre classifiers. For example, classifiers could be used to automatically analyze and sort music into large databases. Music recommendation systems could be used to automatically analyze a user's likings and recommend appropriate songs to listen. Music classifier can be used to recommend music based on the mood of users. Similarity analysis which is a part of music classifiers could be used to detect pirated music copies. These are only some of the many applications of music classification systems.

## Background/Related Work

1. *Integration of Text and Audio Features for Genre Classification in Music Information Retrieval by Robert Neumayer and Andreas Rauber* [1]

The approach to solving the problem in this paper involves using lyrics as well as audio features in order to classify song genres using a corpus created from audio and song lyrics file from of a collection of music. The audio features of the corpus were computed using models such as Rhythm Patterns , Statistical Spectrum Descriptors and Rhythm Histograms. The lyric features were computed using bag of words and a tf-idf.

2. *Multimodal Music Mood Classification using Audio and Lyrics by Cyril Laurier, Jens Grivolla, Perfecto Herrera* [2]

The approach used in this paper revolves around identifying the mood (Angry,Happy,Sad,Relaxed) of the song based on audio and lyrics. The audio classification is done using SVM, Logistic Regression and Random forest on tonal, rhythmic and temporal descriptors of songs. The lyric classification was done by applying k-NN on bag of words model .

3. *Song Genre and Artist Classification via Supervised Learning from Lyrics by Adam Sadovsky,Xing Chen* [3]

The approach used in this paper is similar to what we are doing. We are going to use similar features such as part of speech, and bag of words. The difference between our approach and this paper is that they have used only maxent, and svm, whereas we are going to experiment with different models such as k-NN, gradient boosting, and neural networks.

4. *Semantic Analysis of Song Lyrics by Beth Logan, Andrew Kostisky and Pedro Moreno* [4]

This paper relies on the similarity of documents or in this case songs in order to classify them. The author uses PLSA to determine the most popular words for each genre then classifies new instances on songs based on the occurrences of those words. The author tries this classification using various number topics (concepts) from PLSA each with and without stemming.

5. *Musical Genre Classification by ensembles of audio and lyrics features by Rudolf Mayer and Andreas Rauber.* [5]

In this paper, it uses a very small dataset of around 3000 songs, whereas we are using a dataset which has a good representation of different genres. We are doing model selection and feature set selection manually, whereas in this paper they are using ensembling methods to choose the best performing feature set and classification algorithm.

## Approach:

### ***Pre-Processing*:**

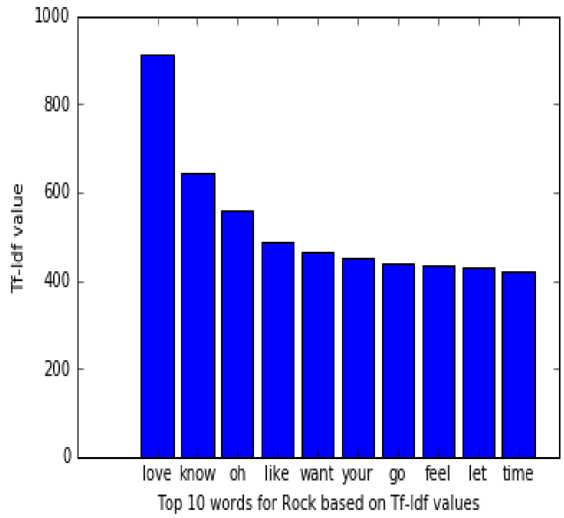
- For the pre-processing part, we first deleted some instances from our data which contained genres like “not available” and “other”, also we deleted instances of genres which didn’t have a good numbers of instances to represent the class.
- After that, we removed stopwords using nltk’s english stopwords and stanford’s stopwords list. We stemmed tokens in each song using nltk’s Snowball stemmer. Also, we have removed unnecessary characters using regular expression.
- Some songs in our dataset had a non-english words, to fix these problem, we used a package called ‘ftfy’. Using ftfy, we have fixed the encoding of the text, and also we removed instances which had a non-english words even after we fixed the encodings.
- We removed word such as ‘Chorus’ and ‘Verse’ which represent different parts of a song.
- In the end, we used regular expression to remove parentheses and square brackets from our songs lyrics.

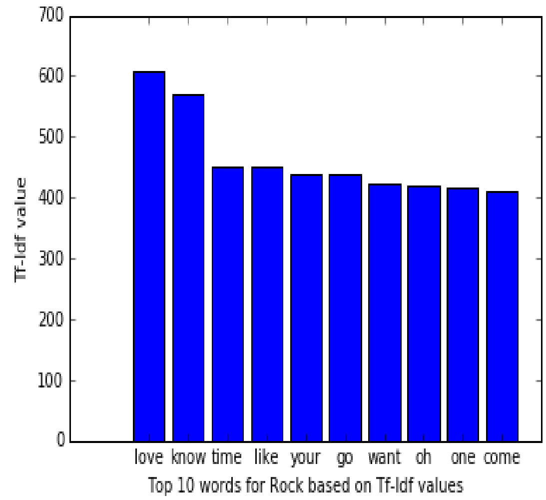
### ***Feature Selection*:**

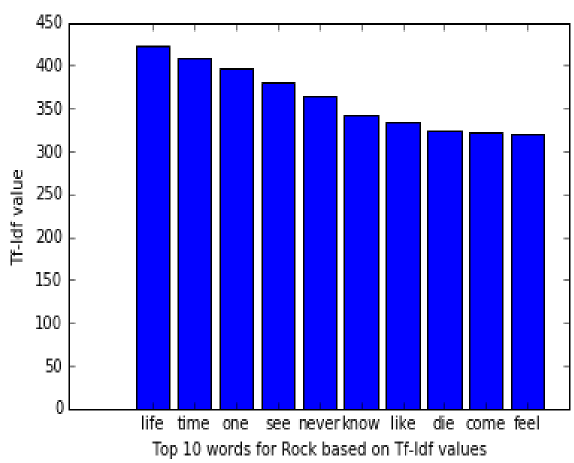
- **Similarity** with four genres: we have created four different features named metal_similarity, pop_similarity, rock_simliarity, and hip_hop_similarity. To calculate these similarity values, we first calculated top-30 words of each genres based on their tf-idf values, after that for every song in our data, we calculated tf-idf scores of each token in a song, and if a token appeared in any of the top-30 words of any genre we used its tf-idf value to calculate the cosine similarity with the tf-idf value of that token in a particular genre in which it appeared.

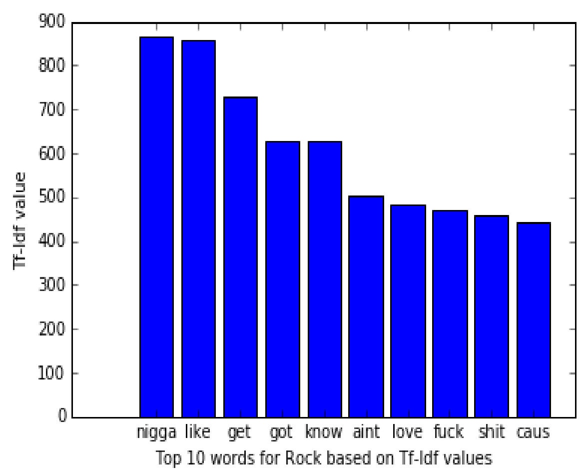
- **Pos tags**: We used nltk tokenizer to tokenize each song in our data, and on those tokens we have used nltk pos tagger to get pos tags. We have count the number of occurrences of a particular pos tag in a particular song, and normalized it, and used it as a feature.

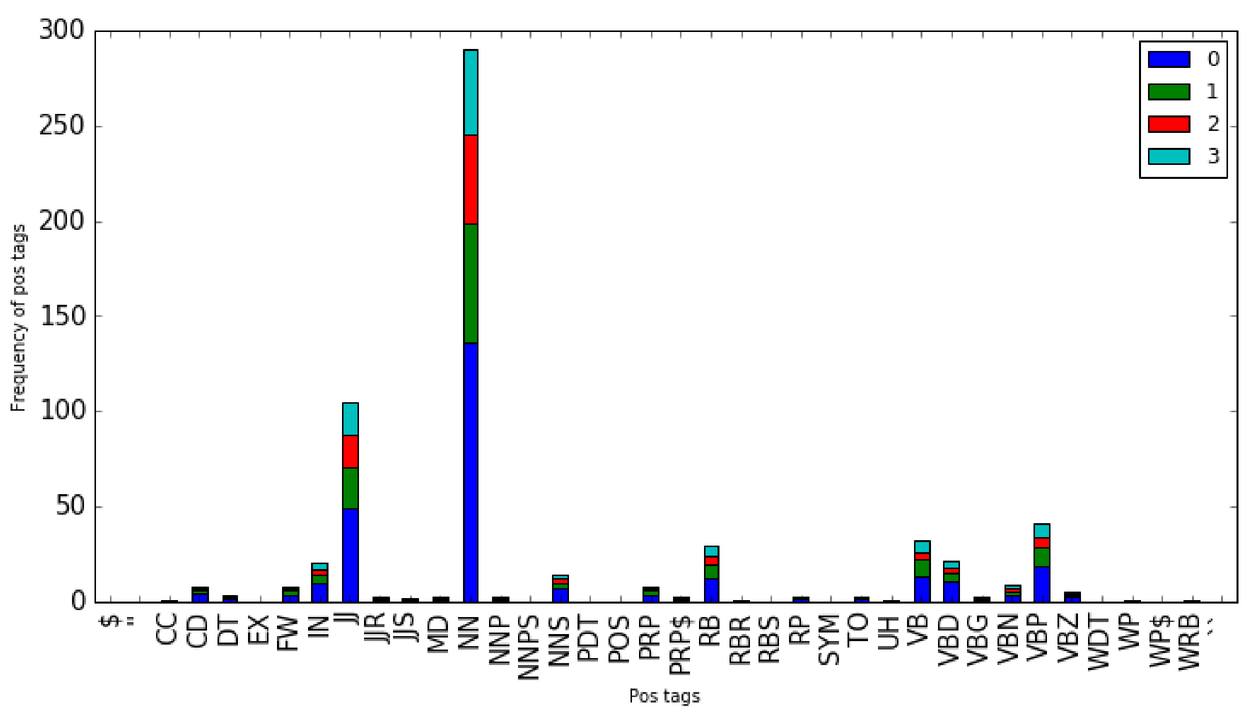
- **Word2vec**: We have trained a word2vec model on the whole dataset, and brown corpus. After training word2vec model, we used it to generate word2vec vector of each token in each song. In the end, we took an average of all the vectors to get a word2vec vector of a song.




(Number of different pos tags per genre)

***Classifiers :***

- Dummy Classifier:  A simple baseline classifier that predicts using simple rules. It is only useful to use as a baseline classifier(from sklearn.dummy)
- kNN Classifier: K-nearest neighbor classifier algorithm is used to predict the target label by finding the nearest neighbor class. (from sklearn.neighbors)
- MLP Clasifier: This model optimizes the log-loss function using LBFGS or stochastic gradient descent. (from sklearn.neural_network)

Log Loss function: -log P(yt|yp) = -(yt log(yp) + (1 - yt) log(1 - yp))

Gradient descent: $\theta_{ij}^{t+1} \leftarrow \theta_{ij}^{t} + \eta(\lambda_j(1 - p(y \mid \vec{h}; \vec{\lambda})))yx_i$

- Gradient Boosting Classifier: GB builds an additive model in a forward stage-wise fashion; it allows for the optimization of arbitrary differentiable loss functions. In each stage 'n_class' regression trees are fit on the negative gradient of the binomial or multinomial deviance loss function.
- Logistic Regression: It uses a sigmoid function for predictions.

$$f(t) = \frac{1}{1 + e^{-t}}$$

## **Experiment**.

We initially started with the data set from Kaggle and then later on we scrapped the raw data from SongLyrics.com. Our data consists of song lyrics and their genres with approximately 200 thousand instances out which we took 20000 instances for each genre to simplify computation. The four genres we decided to classify are:

- Hip Hop
- Metal
- Pop
- Rock

For Baseline estimation we have used Dummy CLassifier. For evaluation we have used following metrics:

- Accuracy Score
- F1 Score

From the our experiments, we achieved the following results:

| Model | Accuracy | F1 Score | Avg Cross Val Score | Parameters |
|---|---|---|---|---|
| Dummy | 0.24 | 0.24 | - | - |
| Naive Bayes | 0.495 | - | - | Default |
| kNN | 0.54 | 0.52 | - | Neighbors=4 |
| kNN | 0.52 | 0.50 | 0.5361 | Neighbors=3 |
| GradBoost | 0.67 | 0.68 | - | Default |
| GradBoost | 0.69 | 0.70 | - | Max Depth=5 Estimators=25 |
| Random Forest | 0.6305 | - | 0.6315 | Default |
| Logistic Regression | - | - | 0.6422 | Default |
| Neural Networks | 0.6995 | 0.699 | 0.7052 | Default |
| SVM | 0.6364 | - | - | Default |

***Confusion Matrices:***

Label Encoding:
0 - Hip Hop   1- Metal  2 – Pop  3 - Rock

- **Naive Bayes**

| **Prediction** | **0** | **1** | **2** | **3** | **All** |
|---|---|---|---|---|---|
| **Actual** | | | | | |
| **0** | 4341 | 1085 | 621 | 602 | 6649 |
| **1** | 223 | 5341 | 412 | 615 | 6591 |
| **2** | 609 | 2564 | 1622 | 1793 | 6588 |
| **3** | 317 | 3732 | 736 | 1787 | 6572 |
| **All** | 5490 | 12722 | 3391 | 4797 | 26400 |

- **Support Vector Machines**

| Prediction | 0 | 1 | 2 | 3 | All |
|---|---|---|---|---|---|
| Actual | | | | | |
| 0 | 5376 | 275 | 517 | 446 | 6614 |
| 1 | 264 | 5110 | 584 | 658 | 6616 |
| 2 | 662 | 1021 | 3831 | 1082 | 6596 |
| 3 | 593 | 1966 | 1529 | 2486 | 6574 |
| All | 6895 | 8372 | 6461 | 4672 | 26400 |

- **Logistic Regression**

| Prediction | 0 | 1 | 2 | 3 | All |
|---|---|---|---|---|---|
| Actual | | | | | |
| 0 | 5342 | 219 | 445 | 603 | 6609 |
| 1 | 223 | 4450 | 854 | 1063 | 6590 |
| 2 | 525 | 987 | 3662 | 1459 | 6633 |
| 3 | 495 | 1430 | 1450 | 3193 | 6568 |
| All | 6585 | 7086 | 6411 | 6318 | 26400 |

**Conclusion:**

After analysis of tf-idf values and confusion matrix we came to know how similar rock and metal songs are. Most of the Classifiers were also predicting wrong labels among these two genres. For the future work, we can use some more features such as parse trees, word endings, and length of a song to distinguish between these two genres and further increase accuracy of different classifiers.

**References:**

1. http://link.springer.com/chapter/10.1007/978-3-540-71496-5_78
2. http://ieeexplore.ieee.org/document/4725050/
3. https://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf
4. https://pdfs.semanticscholar.org/95ed/bdb583a87278c3757993a5b8078d1013bb3c.pdf
5. http://ieeexplore.ieee.org/document/5952341/
6. https://pdfs.semanticscholar.org/e658/ec86e033aae370ba680118a04431071cafe1.pdf
7.http://cs229.stanford.edu/proj2012/BourabeeGoMohan-ClassifyingTheSubjectiveDeterminingGenreOfMusicFromLyrics.pdf