Music Genre Classification by Lyric Analysis

Purvank Patel
Kartik Prakash
Sarthak Anand

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.

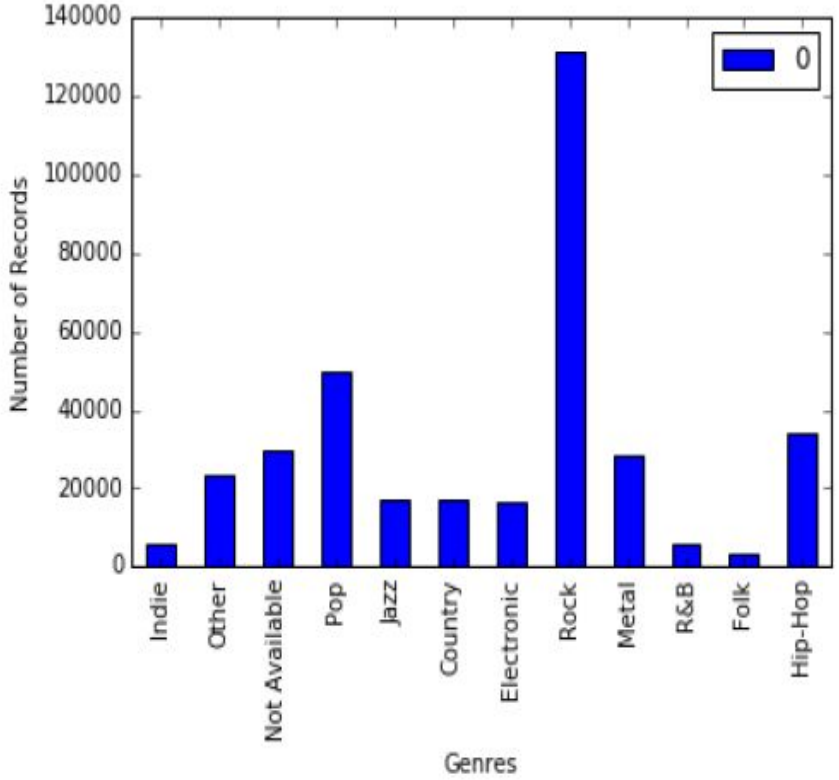
Data

- We scrapped songs data from songsLyrics.com, and metrolyrics.com. Also, we used a song dataset from kaggle.com.
- Our data included around 390,000 songs. Our data includes attributes like song, lyrics, year, artist, and a target attribute genre. For our task, we sampled 20,000 songs of each genre from our original dataset.

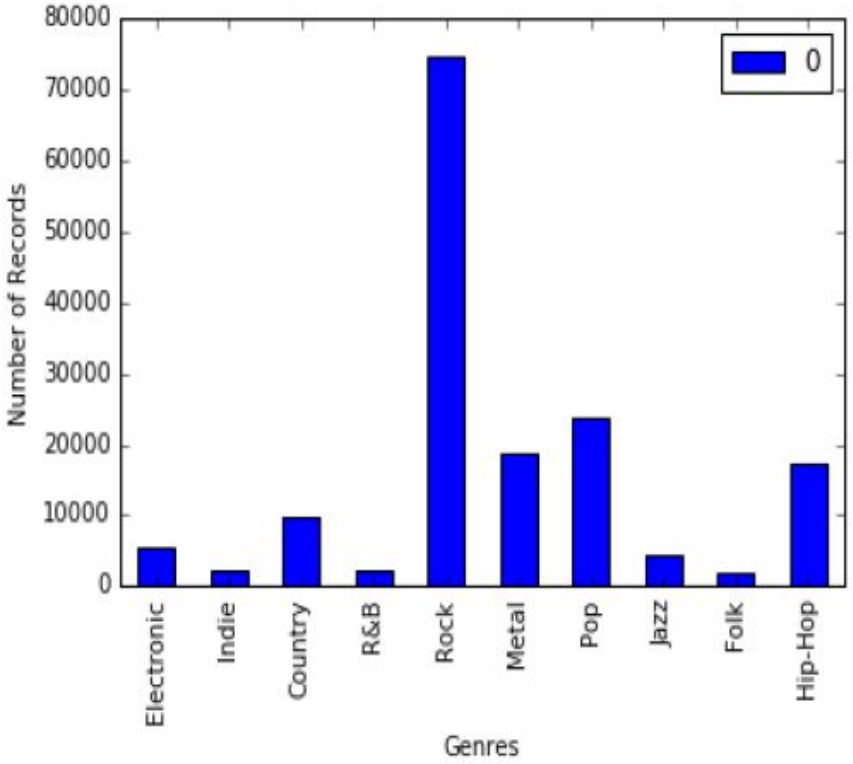
Pre-Processing

- Removed instances with genres like “not available” and “other”,
- Removed genres which didn’t have many instances.
- Removed unnecessary characters using regular expression.
- Removed stopwords using nltk’s english stopwords and stanford’s stopwords list.
- We stemmed tokens in each song using nltk’s Snowball stemmer.
- Some songs in our dataset had a non-english words. 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.

a) Before Preprocessing:

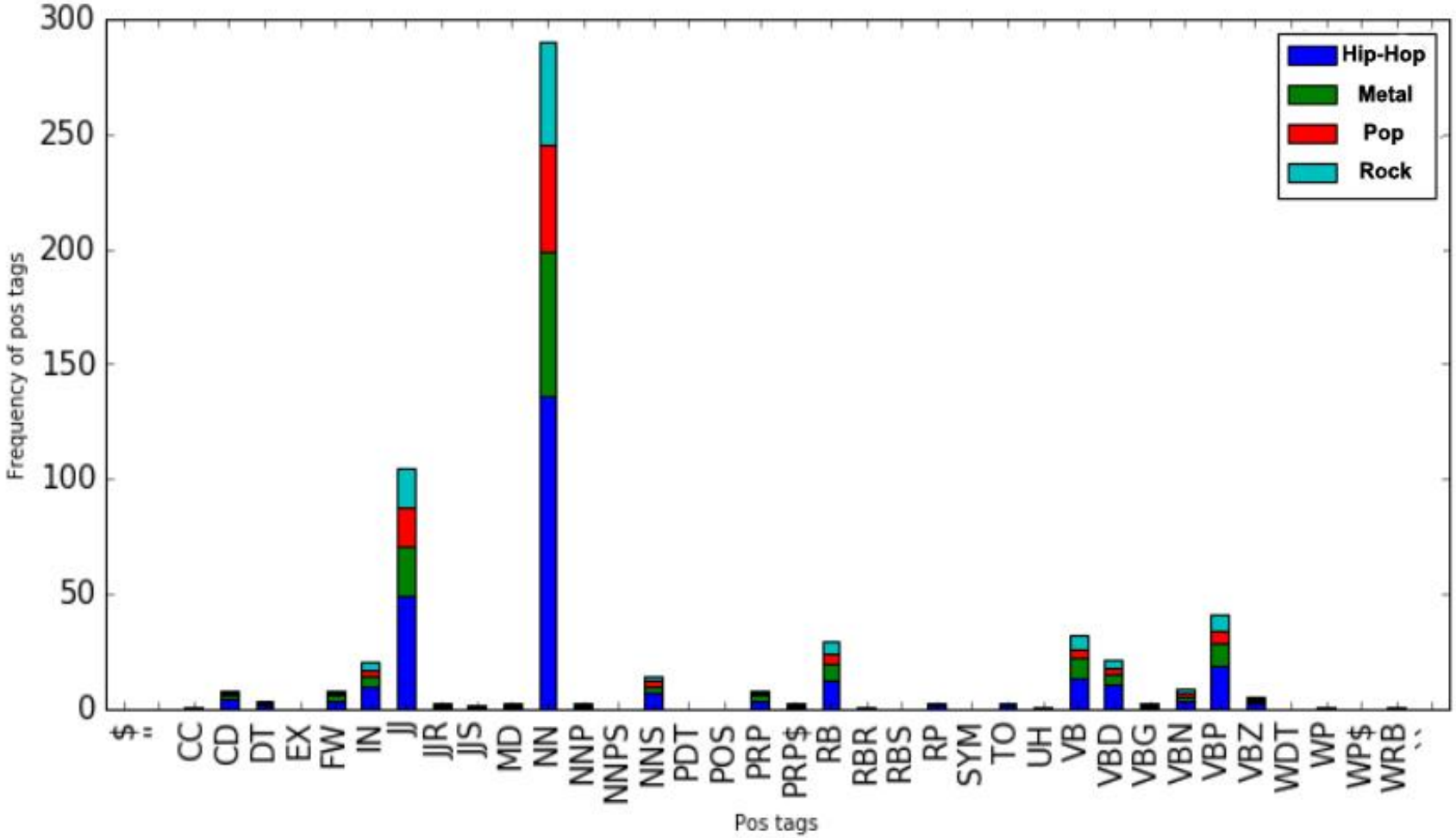


b) After Preprocessing:

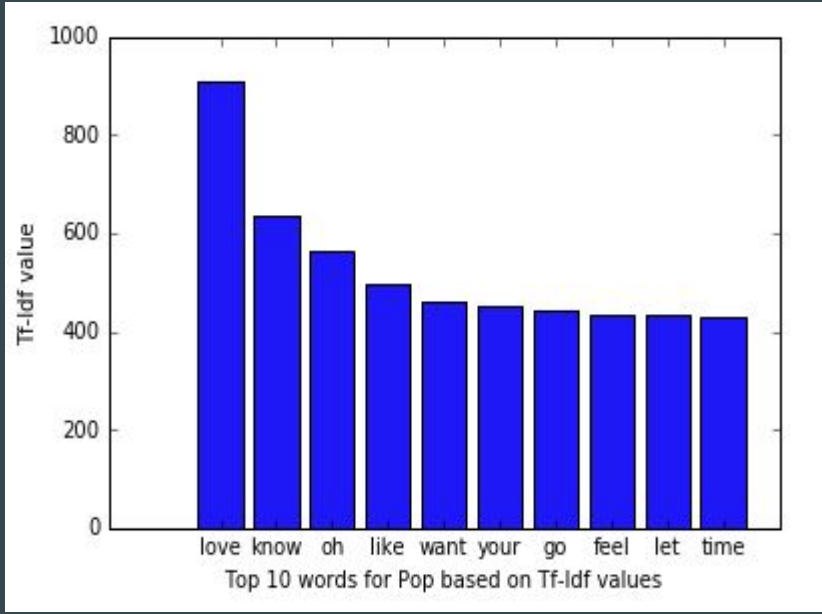


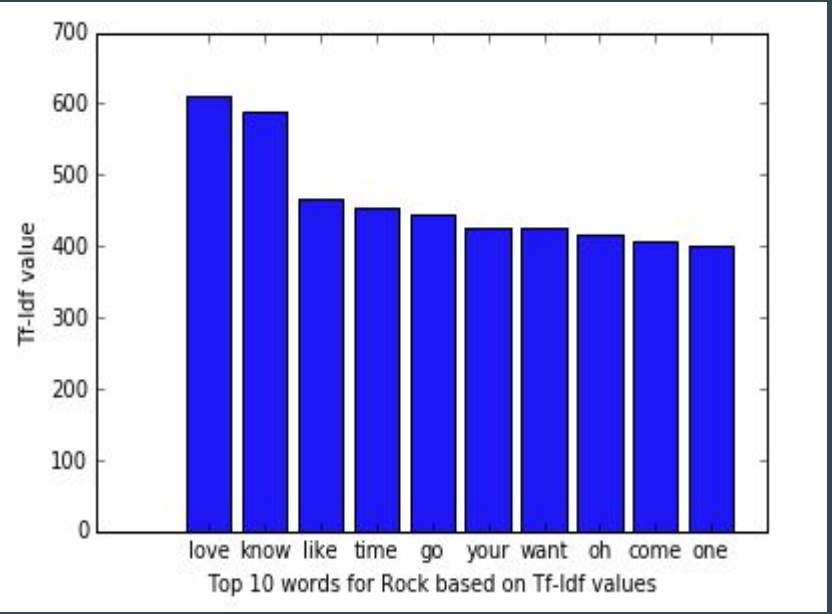
Features

- **Similarity with four genres**: We calculated the top 30 words in each genre using tf-idf. We created four different features named metal_similarity, pop_similarity, rock_simliarity, and hip_hop_similarity. 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**: Using nltk tokenizer, get used a normalized count of pos tags.
- **Word2vec**: We 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.

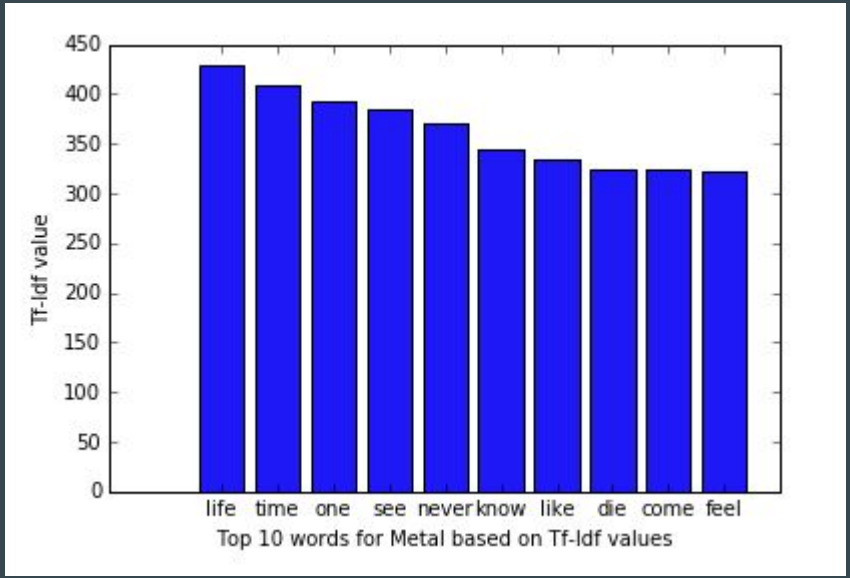

Hip-Hop
Metal
Pop
Rock
Frequency of pos tags
300
250
200
150
100
50
0
$ =
CC
CD
DT
EX
FW
IN
JJ
JJR
JJS
MD
NN
NNP
NNPS
NNS
PDT
POS
PRP
PRP$
RB
RBR
RBS
RP
SYM
TO
UH
VB
VBD
VBG
VBN
VBP
VBZ
WDT
WP
WP$
WRB
Pos tags

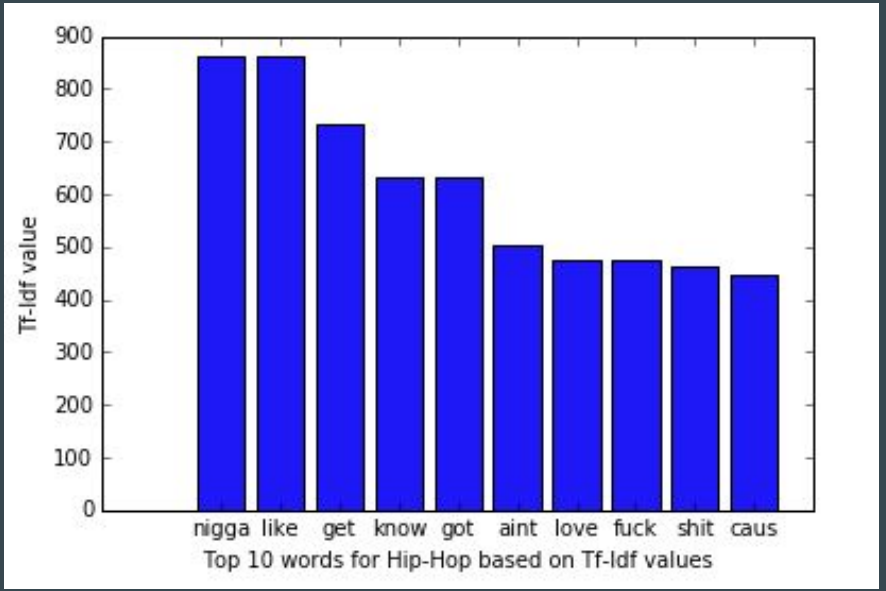
Tf-Idf Values





Tf-Idf Values





Experiments & Results

Models used:

- Dummy Classifier
- kNN CLassifier
- MLP Classifier
- Gradient Boosting
- Logistic Regression

Metrics Used:

- Accuracy
- F1 Score

Results Achieved

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

- 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

- 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

- 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

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

Thank You !