

**Music Genres Classification based on Lyrics**

El Alami Aymane

Supervised by:

Dr. Violetta Cavalli Sforza

CSC 5354 – NLP for Big Data

Submitter on:

May 5th, 2022

**Table of content:**

1. Introduction and motivation
2. Brief Literature Review
3. Resources and Tools
4. Process
5. Results
6. Conclusions and Future
7. Appendices
8. References

**List of Figures**

1. *Figure 1: Loading and Printing the Data*
2. *Figure 2: Plotting the distribution of the data in categories*
3. *Figure 3: function to remove punctuation, removing stop words, and stemming*
4. *Figure 4: Printing the training data with the length of every lyrics*
5. *Figure 5: vectorizing the data*
6. *Figure 6: Extracting the TF-IDF matrix*
7. *Figure 7: Split training and Testing data*
8. *Figure 8: Logistic Regression*
9. *Figure 9: Naïve Bayes*
10. *Figure 10: Support Vector Machine Classifier*
11. *Figure 11: Random Forest Classifier*
12. *Figure 12: K Nearest Neighbors*
13. *Figure 13: Comparing the performance of all models*

List of Acronyms

* NLP: natural Language Processing
* SVM: Support Vector Machine
* KNN: K Nearest neighbors
* AUI: Al Akhawayn university
* NLTK: Natural Language Toolkit
* TF-IDF: Term Frequency – Inverse Document Frequency

**Abstract**

Music genre classification has been a growing topic of research in the past few years. In this paper we are trying to compare multiple classification algorithms to automatically predict the genre of a song based on its lyrics. The main efforts have been towards pre-processing the songs lyrics and preparing it to be fed to the different models. Throughout the research performed while preparing for this project, many were more machine learning oriented in the sense that they were more taking into consideration features of songs like duration, instrumentality, acoustics, key, tempo and more. We are more interest in a Natural Language Programming (NLP) approach, so we are going to dive into one feature that is the lyrics of the songs.

We will be mainly using 5 different algorithms, Random Forest Classifier, Logistic Regression, Support Vector Machine Classifier, K-Nearest neighbors, and Naïve Bayes Classifier. These 5 algorithms have different approaches, which made it very interesting to see which approach would be the best. We are basically feeding them all the same Term Frequency – Inverse Document Frequency (TF-IDF) matrix as a representation of the lyrics of the songs after pre-processing them with stemming and lemmatizing and removing all the stop words and punctuation.

We then find that the overall accuracy has not been very high in any of them and that they were almost close in predicting the classes. At the end, we find that the Random Forest Classifier scored the highest accuracy with 66% accuracy.

1. **Introduction and motivation**

“The soul, like the body, lives by what it feeds on. The soul feeds on music” by J. G. Holland. This is a very accurate description of how music is important. It is therapeutic and speaks to each everyone of us. Therefore, it was important to me to work on a project related to music in general. Since my young age, music played a huge role in my life. I have been studying music theory and instrumentation for more than 5 years, and when I joined Al Akhawayn University in Ifrane (AUI), the first clubs or student associations I joined were music clubs (Rhythm Unity and Voices United Choir) and pursued them until I became the conductor of the choir. All of this to speak to my motivation to work on this project related to music.

Now, since music is very important and speaks to everyone of us in a different way, it also means that depending on the mood or the interests of a person, everyone like a different type of songs. Hence the idea of classifying the songs. From an NLP perspective, the lyrics can say a lot about a song in terms of words used and how they convey different meanings and give different insights on the theme or genre. Especially with the evolution of music throughout the years, it would be interesting to see what traditional algorithm can extract from the lyrics and how accurately they can predict the genre of the songs.

1. **Literature Review**

Before starting the project, I have tried to do a small literature review to see what people have already been working on and see the approach people are taking. Of course, the aim is not to exactly replicate what has already been done, but I wanted to see if people are actually interested in classifying music and how far did they take it.

The thing that I found is that very few people use a sole NLP approach, meaning that they only take into consideration the lyrics of the songs.

First, there is a paper from Aziz nasridinov called “A study on Music Genre Recognition and Classification Techniques”. It says that in general, the genre classification process of music has two steps: features extraction and classification. The first step is obtaining audio signal information, and then classifying the music into various genres according to the extracted features.it uses a combination of supervised machine learning and neural networks in addition to unsupervised learning. (Nasridinov, 2014)

Another paper that speaks about how important classifying music as it paves the way for many more applications for example to be integrated in a recommendation system. In this paper they mostly rely on audio files where they extract features and feed them to 8 different classifiers: Decision Trees, K Nearest Neighbors, Logistic regression, Naïve Bayes, Neural Networks, Random Forest, Support Vector Machine, XGBoost. And they were able to achieve an accuracy of 91%. (Nihalaani, 2021)

In this last paper talking about classifying music genres using NLP techniques, we find that they also take into consideration the audio part. They actually have many audio features taken into consideration like acousticness, danceability, duration, energy level, instrumentalness, speechiness, key, liveness, mode, tempo, loudness, time signature and valence. The classifiers used in this paper include Bernoulli Naive Bayes, AdaBoost, Bagging, Extra Trees, Decision Tree, Ridge, KNeighbors, Gradient Boost and Random Forest classifiers. The accuracy of their classification ranges from 44% with Bernoulli Naïve Bayes for example to 98% with Random Forest Classifier. (Guner, 2019)

Based on all this research, we can say that many of the published papers do take into consideration a lot of features about the audio, and some combine them with the lyrics. The idea of this paper is to actually focus only on lyrics and see if we can get close to the average performance of these models that use also the audio as part of the classification. Of course, lyrics alone won’t be enough because the same words could be used for both hate and love songs for example if we are talking about sentiment analysis. “Your Love is killing me” and “I want to kill myself because you’re not around me anymore” are two different sentiments but both have the word ‘kill’ but one is about finding love and the other is about lost love.

1. **Resources and Tools**

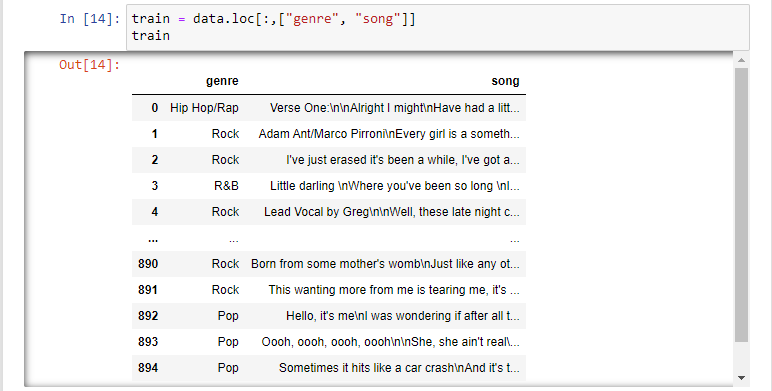
The main library used in this project was *SKlearn.* it is probably the most useful library for machine learning in python. The SKlearn library contains a lot of efficient tools for machine learning including classification, regression, clustering and dimensionality reduction. (Kunal, 2015) It very focused on machine learning algorithms, and provides free built-in statistical, mathematical, and general-purpose models. We are using the scikit-learn library to import feature extraction libraries and functions like vectorizers and TF-IDF transformer, classification models, data split functions, and models evaluation functions.

We used the Jupyter notebook for as Integrated Development Environment (IDE). It is an open-source web application that anyone can use to create and share documents that contain live code, equations, visualizations and text. Jupyter Notebook is maintained at the project jupyter. (Driscoll, n.d.) it is basically an interactive platform that allows developers to run snippets of code independently and is also downloadable within the anaconda environment along with various other development tools.

For the Text pre-processing, we are using Natural Language Toolkit (NLTK). It is a leading platform for building python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning. (Natural Language Toolkit , n.d.) We are using NLTK to import stop words corpora for data cleaning, tokenizers, and Porter Stemmer.

The pandas library is also used in our project, it is mainly there for any data manipulation. “Pandas” stands for “Python Data Analysis Library”. The term is derived from the term panel data also a term for multidimensional structured data sets. It is quite a game changer when it comes to analyzing data with python and it is one of most preferred and widely used tools in data manipulation if not the most used one. It is an open source free to use library. (Bronshtein, 2017)

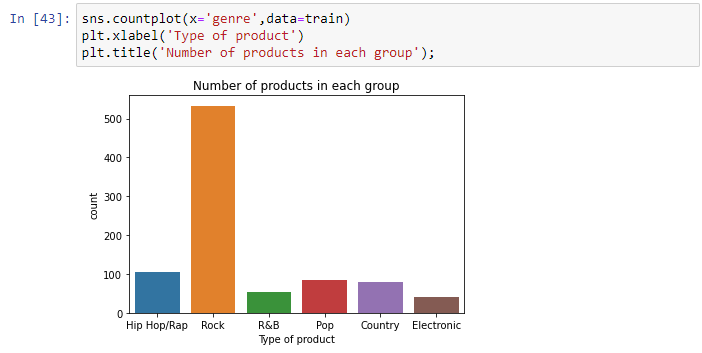
1. **Process**
2. Data



*Figure 1: loading and printing the data*

The data was obtained from a GitHub repository and can be found under Appendix A

* Data distribution:

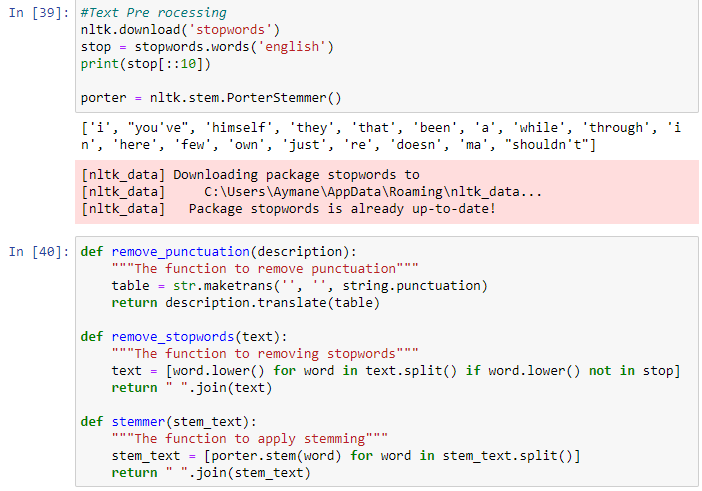


*Figure 2: Plotting the distribution of the data in categories*

this where I show the distribution of the data that I have by genres.

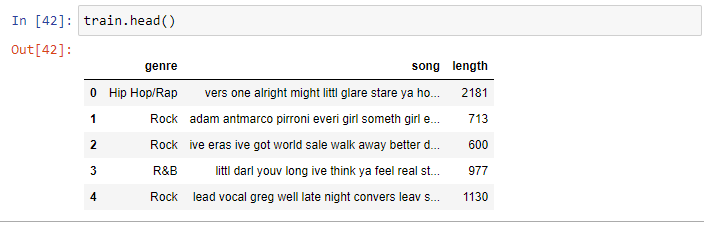
1. Data pre-processing

* Downloading stopwords from NLTK
* Initializing the PorterStemmer
* Defining functions:
* To remove punctuation
* To remove stopwords
* To stem the words



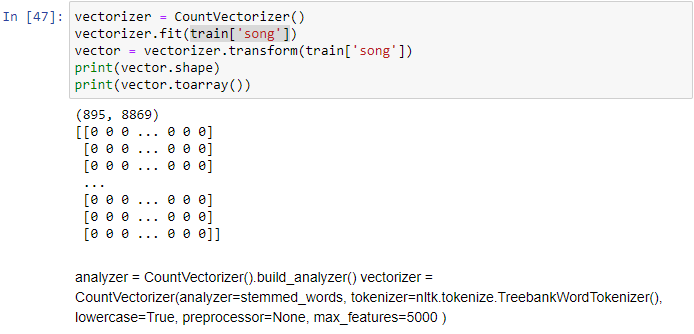
*Figure 3: function to remove punctuation, removing stop words, and stemming*

* Data after pre-processing



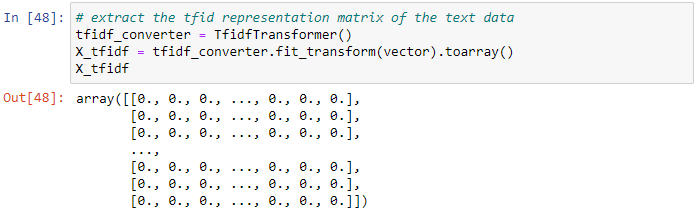
*Figure 4: Printing the training data with the length of every lyrics*

1. Data vectorization



*Figure 5: vectorizing the data*

1. Computing TF-IDF



*Figure 6: Extracting the TF-IDF matrix*

After vectorizing the songs lyrics, we extracted the TF-IDF matrix. That matrix will serve a representation of the lyrics of the songs. That will be the data fed to the models for classification.

1. We split the data into training and testing data

A picture containing diagram

Description automatically generated

*Figure 7: Split training and Testing data*

1. Now we feed the training and testing data to the models and evaluate them
   1. Logistic Regression

Initially, we had only the logistic regression model trained, tested and evaluated. These were the results

Table

Description automatically generated with low confidence

*Figure 8: Logistic Regression*

* 1. Naïve Bayes model

Text

Description automatically generated with medium confidence

*Figure 9: Naïve Bayes*

* 1. Support Vector Machine Classifier Model

A picture containing text

Description automatically generated

*Figure 10: Support Vector Machine Classifier*

* 1. Random Forest Classifier Model

Table

Description automatically generated with medium confidence

*Figure 11: Random Forest Classifier*

* 1. K Nearest Neighbors

A picture containing table

Description automatically generated

*Figure 12: K Nearest Neighbors*

1. **Results**

Now we move to comparing the performance of the models

Graphical user interface, text, application, email

Description automatically generated

*Figure 13: Comparing the performance of all models*

After running all 5 models: Logistics Regression, Random Forest Classifier, Support Vector Machine Classifier (SVM), Naïve Bayes, and K Nearest neighbor (KNN), we got each time a list of predicted genres of music that we call classes or labels. We then compare them to the y\_test array that contain the true labels or classes. Each time we compute the accuracy of the models in terms of predicting those specific labels.

Figure 13 shows all models results in terms of accuracy together. We put all of them into a data frame and sorted everything in an ascending order based on the accuracy score of their predictions.

we can obviously see that the overall performance of all the models wasn’t very high in terms of accuracy of predicting the music genres. However, we can see that the Random Forest Classifier scored the highest accuracy.

Based on the results and the performance of the models, we can clearly conclude that making predictions solely based on the lyrics of the songs is not very accurate and can incur many misclassifications.

1. **Future work**

As future work, to improve on the performance of the models in classifying the songs, I recommend including the other features that other papers and projects in the literature used like the duration, instrumentality, acoustics, key, tempo and others. I think that the more data we use to describe the songs, the more accurate the models would be in predicting the genres.

Of course, I did not do that as part of this project because I wanted to keep strictly from an NLP perspective and only work with lyrics.

I think that on a bigger scope of this project, the approach of include a lot of features to describe the songs would be more acceptable.

As a side note, I have tried the same approach on products descriptions, and it gave much better results with the highest accuracy at almost 92% accuracy. I can assume that it is because there wouldn’t be or there would very few instances where the TF-IDF matrix of a description of a product would be similar to different one. Whereas when it comes to songs, we can find many words in different songs even if they belong to different genres. This makes it more confusing for the models to accurately classify the songs.

**Appendix**

Appendix A: Data used in this project – [Link](https://github.com/rasbt/musicmood)

* + If the link is not accessible: https://github.com/rasbt/musicmood

Appendix B: My notebook of the project on my Github- [Link](https://github.com/ELALAY/NLP-Music-Classification/blob/912b40734ec4696dd6fdef14715681569bbe57e7/Music%20Genres%20Lyrics%20Classification.ipynb)

* + If the link is not accessible: <https://bit.ly/NLPFinalNotebook>

# References

Bronshtein, A. (2017, April 18). *A Quick Introduction to the "Pandas" Python Library*. Retrieved from Towardsdatascience.com: https://towardsdatascience.com/a-quick-introduction-to-the-pandas-python-library-f1b678f34673

Driscoll, M. (n.d.). *Jupyter Notebook: An Introduction*. Retrieved from Realpython.com: https://realpython.com/jupyter-notebook-introduction/#:~:text=The%20Jupyter%20Notebook%20is%20an,the%20people%20at%20Project%20Jupyter.

Guner, L. (2019). Classifying music genres and making suggestions with natural language processing. *ResearchGate*, 5.

Kunal. (2015, January 5). *Scikit-learn(sklearn) in Python – the most important Machine Learning tool I learnt last year!* Retrieved from analyticsvifhya.com: https://www.analyticsvidhya.com/blog/2015/01/scikit-learn-python-machine-learning-tool/

Nasridinov, A. (2014). A Study on Music Genre Recognition and Classification Techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 42.

*Natural Language Toolkit* . (n.d.). Retrieved from NLTK.org: https://www.nltk.org/

Nihalaani, R. (2021). Using Machine Learning to Classify Music genre. *International Journal for Research in Applied Science & Engineering Technology*, 9.