Genre clustering of music reviews

PML Project 2

Tudor Andrei Dumitrascu

January 19, 2021

1 Introduction

The aim of the project is to cluster music reviews based on genre. This will be achieved by employing two clustering algorithms and evaluating their performance. Along those two, a baseline will be established using random prediction and a RandomForest algorithm.

2 Dataset

The dataset consists of album reviews of various music albums from the Pitchfork website. [3]

The dataset is highly unbalanced, with most of the samples belonging to one class. See Fig. 1

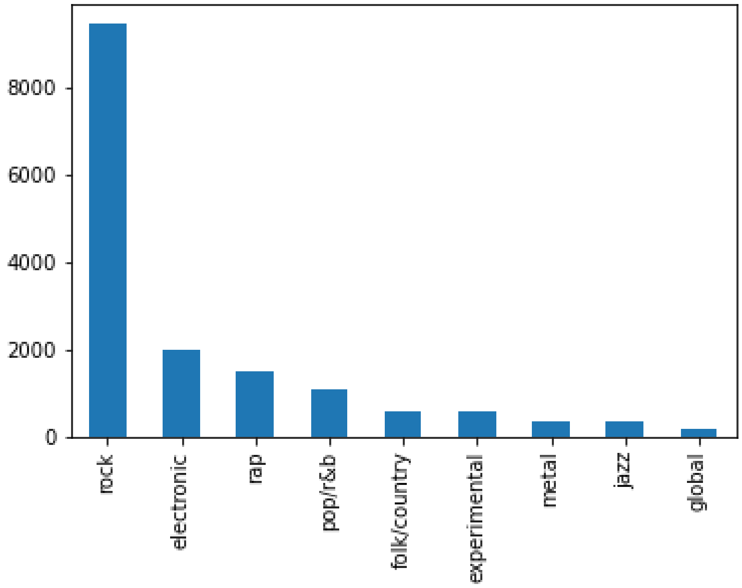


Figure 1: Genres Distribution

In an attempt to balanced the dataset, two strategies were employed:

- Sampling from the majority dataset, using an upper limit on the number of samples.

- Sampling from the majority dataset, using an upper limit and removing the classes with low number of samples (i.e. less than 400)

The last sampling strategy worked a bit better than the rest. If all the samples would have been used, then the majority class would have distorted the dimensionality reduction results. The upper limit in this case it 1000 samples.

2.1 Preprocessing

In order to use the dataset some preprocessing steps were taken:

1. remove the \xa0 symbol When loading the csv file the xa0 whitespace symbol appeared as text and it had to be removed
2. remove symbols Removes all the symbols that are not alphanumeric or whitespaces, along with the dash ('-') and underscore ('_')
3. stemming This removes all the suffixes and prefixes of the words. There are multiple stemming algorithms, and the Snowball Stemmer was used in this case.

For a better result, lemmatization could have been used for the last step. This was the initial idea but was replaced with stemming due to the fact that the lemmatization of a entry (i.e. a review of an album) took too long, and considering the fact that the dataset contains 18k rows, it's not a feasible solution.

For the final preprocessing step, the whole dataset was processed using the TF-IDF algorithm.

With the following parameters:

- minimum document frequency: 10
- maximum document apparitions: 100
- maximum features: 150

  They were set so that from each category only the relevant words were kept, in order to create a distinction between the genres.

2.2 PCA and TSNE

In order to test multiple hypothesis, the data was also reduced in dimensions using the PCA and t-SNE, down to two components.

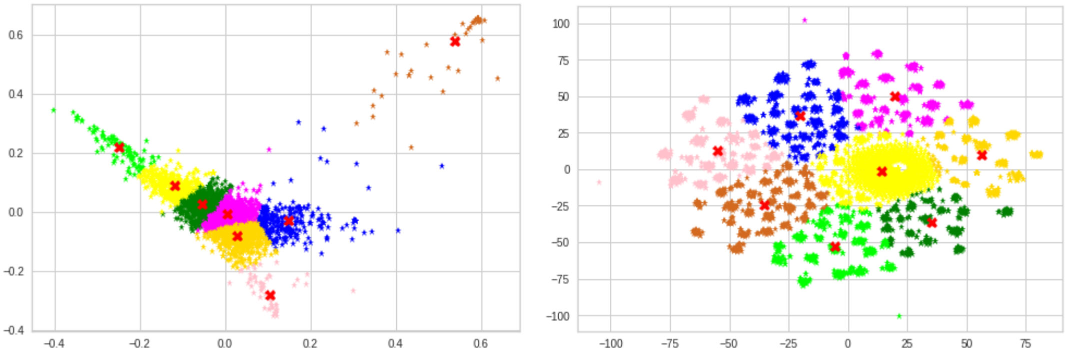
In some of the cases it greatly improved the outcome, especially for the DBSCAN algorithm.

3 Methods

In order to have a better overview of the performance of the clustering algorithms, a supervised method was used and also the probability of choosing at random were computed.

3.1 Kmeans

Because the number of clusters it is known, we just set it as a parameter to function and train the algorithm on the data.



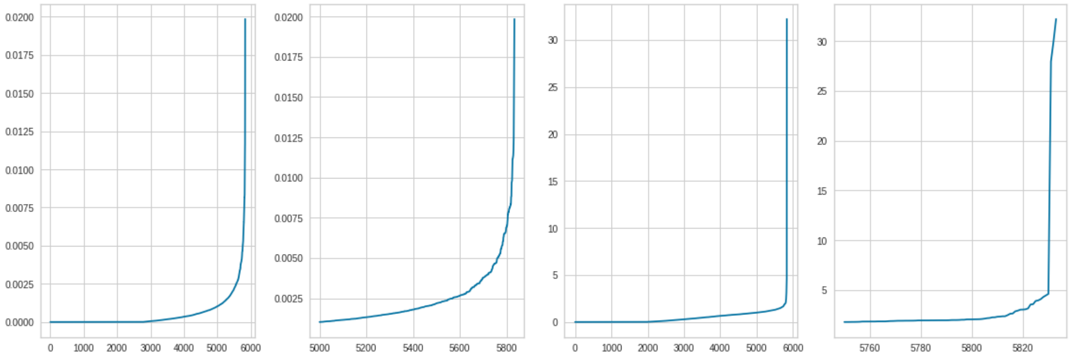
(a) Kmeans PCA (b) Kmens t-SNE

Figure 2: Kmeans with Dimensionality reduction

3.2 DBSCAN

Since the DBSCAN algorithm computes the number of clusters automatically the only parameters that have to be tuned are ϵ and the minimum of number of points in the neighbourhood.

In order to compute ϵ, we compute the distance using the NearestNeighbours and plot the results.See Fig. 3. The value is the one, just before the distance significantly increases. [4]



(a) PCA (b) t-SNE

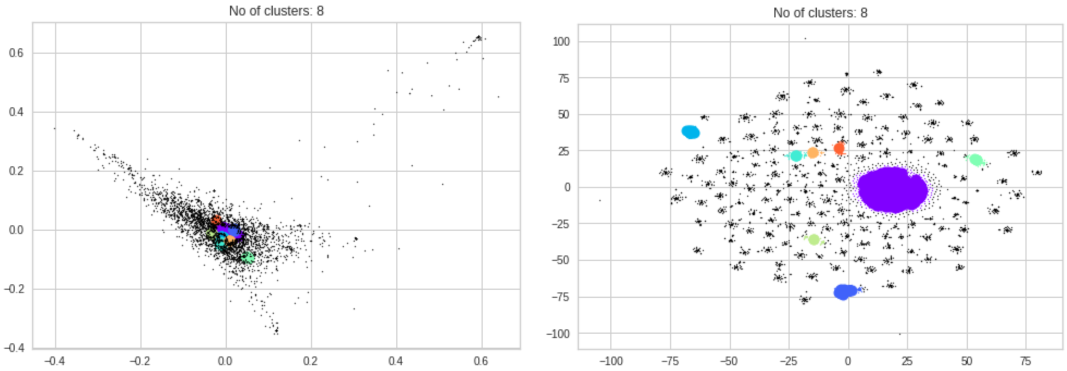
Figure 3: NearestNeighbours Distances, on the right it's the zoom-in on the elbow area

After the value for ϵ was found, then the min_samples value was searched in order to find proper clusters. I believe that this happens because the clusters are not clearly separated, and the algorithm can group up too many points together even if they are from different classes, or create a lot of clusters based on individual samples. See Fig. 4

4 Evaluation and Results

In order to evaluate the models, a simple function that compares the true labels with the predicted labels is not enough as the label assignment is random by the clustering algorithms. Therefore, multiple metrics have been used:

- Completeness Score
- Homogeneity Score



(a) PCA

(b) t-SNE

Figure 4: DBSCAN on PCA and t-SNE data

- Silhouette Score
- Accuracy As stated previously the labels cannot be simply compared, but this problem can be overcome. By permuting the columns of the confusion matrix in order to obtain the maximum value on the main diagonal. This permutation process is quite consuming and it's replaced by the Hungarian algorithm [1] [2]

Method	Accuracy	Completeness	Homogeneity	Silhouette
Random prediction	12.43%			
RandomForest	57.99%			
Kmeans	18%	0.069	0.015	0.094
- t-SNE	20%	0.040	0.040	0.374
- PCA	22.8%	0.080	0.060	0.431
DBSCAN				
- t-SNE(eps=0.015)	19.5%	0.059	0.023	-0.31
- PCA (eps=0.005)	21.8%	0.024	0.011	-0.248

What it can bee seen from the metrics is that most clustering algorithm do a poor job in creating meaningful clusters from the given data. From the Completeness score it's clear that the data from the same cluster is spread across different clusters, and from the Homogeneity score it's clear that a cluster contains data from different labels. The fact that clusters contain different labels it's also given from the Silhouette score, and in the case of DBSCAN having the negative value indicates overlapping. The only class which could be easily identified is the center cluster in the case of t-SNE, which belongs to the label with the most amount of samples.

5 Conclusion

As expected the clustering algorithms have really low accuracy due to the fact that they are not supposed to be used for classification task. Even so, the quality of the clusters is really low, due to the overlapping of the data. One could argue that it's really hard to determine a difference between the data since all the text belong to one overall class "Music." Therefore the words that could make the difference from genre to another are unique words, which cannot be guaranteed to be used in multiple reviews of the same genre.

In conclusion, the aim crating clusters of genres on music reviews is really hard, almost impossible.

References

[1] Stanislas Morbieu, "Accuracy: From classification to clustering evaluation," 2018. [Online]. Available: https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/

[2] M. N. François Role Stanislas Morbieu, "CoClust: A python package for co-clustering." 2019. [Online]. Available: https://smorbieu.gitlab.io/accuracy-from-classification-to-clustering-evaluation/

[3] Nolan Conaway, "18,393 Pitchfork Reviews." https://www.kaggle.com/nolanbconaway/pitchfork-data, 2017.

[4] Cory Maklin, "DBSCAN python example: The optimal value for epsilon (EPS)," 2019. [Online]. Available: https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc