Genre Classification of Songs Based on Lyrical Features Using the KNN and Naïve Bayes Models

Tasneem Abed - 1408535
COMS4030A 2019 Project
Due date: 22nd May 2019

***Abstract*—The organization of music is a scaling problem in the music world as online streaming services have databases that contain millions of songs. One such organization technique may be to label songs by their genre. In this study, multiclass classification algorithms for songs genres are implemented. The data used are lyrical features such as unique words per line as well as a bag-of-words representation of text. Both a Naïve Bayes and K-Nearest-Neighbours algorithm is implemented to perform the classification and compared.**

I. INTRODUCTION

OVER the past decade there has been a rapid increase in access to the Internet and online content globally. This has caused a shift in the music industry from physical formats, such as CD's, towards online streaming and digital formats. The databases of some music streaming services are approaching sizes of millions of songs. This presents the challenge of organizing music content. Strategies that enable access to music content is under intensive research in the field of Music Information Retrieval (MIR) [1]. One of the primary features of a song is the genre it belongs to and, generally, the genre is the same as the genre of the artist who wrote it. Typically, audio clips combined with lyrical features are used in music analysis, however this study focuses on the lyrical content [3]. Mobile applications such as Shazam and SoundHound are able to listen to audio clips and identify the song name, artist and genre amongst others, however, in this project we are interested in the lyrical content of songs and if they contain enough information to predict the genre.

Classification is an important and powerful concept in machine learning. Supervised learning makes use of data that has been labelled to train a model to predict the labels for new data points. This study will implement the K-Nearest Neighbour (KNN) multimodal classification model to predict one of 4 genres of a song based on meta-data of the song's lyrics. A Naïve Bayes model will also be implemented to perform the same classification but this time using the bag-of-words representation. The performances of each of these algorithms will be compared.

II. METHODOLOGY

A. *DATA*

THE data used in this project was scraped from the Internet using LyricsGenius in the Genius API. Genius is a website that claims to have "the world's biggest collection of song lyrics and musical knowledge" and provides an easy to use API that generates a personal key for you. The objects that are returned from the API contains information of the song that was searched for, however is missing the genre. In order to retrieve songs from a particular genre, I compiled a list of 25 artists for each of the Country, Pop, Rap and Rock genres and extracted 30 songs per artist. This gave a dataset with 3000 songs and 750 songs from each genre. Only the song name, artist name and lyrics were extracted. The extraction process took long and had to be redone numerous times as each time an improvement was made that reduced pre-processing efforts. It was noticed that many songs were repeated as there were remixed or live versions of the same song that resulted in duplicate lyrics. There were some songs that did not contain any lyrics as they were just instrumentals. To account for this, songs that had the terms 'Live', 'Remix' or 'Instrumental' were skipped. Most punctuation marks were also removed.

B. *DATA PRE-PROCESSING*

In a similar study that used unsupervised learning techniques to find genre clusters of songs, the lyrics were not directly used as features. Meta-data of the lyrics were extracted and used as features instead. From this study [2] I have borrowed these features to use as my own in the KNN algorithm. These features are:

- Number of words per line
- Unique words per line (UWPL)
- Mean word length (MWL)
- Token ration (TR) - The ratio of unique words and total number of words and
- Total number of words

A sample of the data is shown below in Table 1. It is important to note that since the genre was not a feature that we could extract directly using the API, the genre feature was manually added during pre-processing. Each of the above features were manually calculated.

After feature extraction the ranges of values of each feature needed to be analysed. Most features ranged between 1 and 10 or 0 and 1, but the total number of words had a range of 2172 and thus a large bias would be placed on this feature. Due to this, I normalized the data using rescaling:

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

TABLE I
EXTRACT OF PROCESSED DATA FOR KNN

WPL	UWPL	MWL	TR	Total	genre
0.50000	0.6	0.430919	0.37537	0.151136	Country
0.41667	0.5	0.39847	0.393782	0.175707	Pop
0.50000	0.6	0.499180	0.485772	0.224849	Rap
0.25000	0.3	0.534355	0.515152	0.57951	Rock

After normalization, all the features ranged between 0 and 1.

Pre-processing the data for the Naïve Bayes model involved removing all punctuation marks as well as encoding the genre feature so that Country = 1, Pop = 2, Rap = 3 and Rock = 4.

C. KNN ALGORITHM

The K Nearest Neighbour algorithm is a simple classification algorithm that relies on a distance metric between data points. The algorithms takes in a new data point and measures the distance between it and all other data points. It then selects the K nearest data points to it and assigns the new data point the modal class of those points. Figure 1 shows a plot of 2 of our variables against each other. This makes it easier to visualize KNN. Looking at the 2 green triangles at the top of the plot, if we were running a 1-NN on the topmost data point, the algorithm would assign the green class (Rap) to it as the nearest data point to it is green.

The distance metric used is the Euclidean distance:

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

Had the data not been normalized, the distance calculation would have placed a huge bias on the data of higher values. The value of K is chosen to be an odd number, starting at 1, and is consecutively tested.

D. NAÏVE BAYES ALGORITHM

The Naïve Bayes algorithm takes a probabilistic approach to learning by calculating the conditional probability of a class given the data. It also takes into account the probability of the the data given the class. The algorithm is termed 'Naïve' due the assumption it takes that all features are independent. Conditional independence is explained as follows: From the rules of probability, we know that

$$P(X, Y) = P(X)P(Y|X)$$

so, for a class C,

$$P(X, Y|C) = P(X|C)P(Y|X, C)$$

Now, the assumption that knowing the value of X does not effect the value of Y as long as we know class C i.e. X and Y are conditionally independent. More generally, for $x = (x_1, x_2, ..., x_n)$, $x_1, x_2, ..., x_n$ are conditionally independent given a class c iff $P(x|c) = \prod_{i=1}^{n} P(x_i|c)$. The Naïve Bayes classifier is given by this conditional independence assumption applied to Bayes rule, where Bayes rule is:

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}$$

thus giving

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X) = \sum_{Y'} P(X|Y')P(Y')}$$

where $P(Y)$ is the prior probability of each class, $P(X|Y)$ is the class conditional model which is the likelihood of class Y generating X, and $P(X)$ is the normalization term.

Lets look at the bag-of-words representation. It is a representation used in natural language processing and information retrieval for feature extraction when modelling language. In our case we remove duplicate words from every lyric, then create lists of lyrics for each genre. We then create a dictionary of words for every genre. The dictionary has each word appearing in at least one lyric in its genre and has a probability as its value. The probability is the number of times the word appears in the genre divided by the total number songs in that genre. Using these dictionaries, we can take an unseen set of lyrics and check if each word appears in each dictionary, and if so, use the probability attached to that word to satisfy the Naïve Bayes classifier.

A problem that may arise with Naïve Bayes is the zero-frequency problem. A test lyric may contain a word that never appears in the training data, thus it has a 0 probability and when we multiply this with the rest of the probabilities we get a 0 in the denominator. To account for this, a technique called Laplace Smoothing is implemented. We add a small number to each count in the following way:

$$P(new\ word|class) = \frac{\{x_{word} = 1, y = class\} + k}{\{y = class\} + n_{word}}$$

where n_{word} is the number of values x_{word} can take, which in our case is 4, and $k = 1$. In this way there will never be a 0 in the denominator.

With data that contains many words, we will end up multiplying many small probabilities together in order to obtain the denominator, this could lead to an underflow problem as the value becomes so small it gets rounded off to 0. The solution to this problem is to use logs. Now our multiplication becomes additions and small values become much larger eg: $0.000000001 = -9$.

In the next section I will discuss the implementation of these algorithms and any nuances that were applied.

E. IMPLEMENTATION

Both the KNN and Naïve Bayes algorithm were implemented from scratch. The testing and training data split was consistently a 20% - 80% split respectively.
During testing of the Naïve Bayes model, it was noticed that building the dictionaries of words and probabilities for each

genre took long. This was attributed to the fact that the overall set of unique words had 293470 words, and 4 dictionaries had to iterate through it. 2 algorithms were applied in order to increase performance and accuracy.
Firstly, stemming was applied. Stemming is the process of reducing a word to its root form by removing any suffixes or prefixes. In this way, the number of words would be reduced as words with the same stem will become one. For example, the words played, plays and playing would each be reduced to its stem word, play. There is now one word with its corresponding probability instead of 3 words.
Secondly, stop words were removed. Stop words are common words such as 'at', 'by', 'a' and 'the' that do not carry any sentiment. These words were thus simply removed by ignoring the word if its length was 2 or smaller.
These techniques may have contributed to the increase in accuracy of the model, but only slightly increased the running time. To make running faster I decided to use a smaller dataset. Also, when checking if a word in the dictionary appears in the test lyric, it is faster to check if the lyric word is in the dictionary as this requires less checks. The effects of this on the accuracy will be discussed in the discussion section.

It is important to make sure that the data is balanced in order to avoid any bias. By balanced I mean that there are almost equal amounts of data points per class. During the data collection process, equal amounts of each genre were scraped. Once training and testing splits were done, the sizes of each class was checked to make sure they were evenly split. Every case had an even split and one such case had 155, 152, 148 and 145 datapoints in each genre respectively.

The performances of both algorithms is measured by the accuracy of their predictions and the classification error. The classification error refers to the number of cases incorrectly classified and is evaluated as:

$$\frac{1}{N}\sum_{n=1}^{N}[y^{(n)} \neq h(x^{(n)})]$$

where $h(x^{(n)})$ is the predicted class. The accuracy of predictions is the number of predictions the classifier gets correct divided by the overall size of the testing set. A confusion matrix is used to visualize the results. A confusion matrix is a matrix with dimensions CxC where C is the number of classes. In our case, the matrix is of size 4x4. Each column represents the actual class of the test point and each row is the predicted class of the test point. Our main concern is with the diagonal elements as these show the number of correctly predicted test points.

An interesting question was raised while trying to predict the genre of the song using the extracted features for the KNN algorithm, which of these features has the biggest effect on the accuracy? This was investigated by running the KNN algorithm and removing one feature each time and comparing accuracies.

In the next section, all the results of experiments will be shown and a discussion of these will follow.

III. RESULTS

A. *NAÏVE BAYES RESULTS*

In this subsection, the results of the Naïve Bayes experiments will be presented and a discussion will follow in section IV. The confusion matrices correspond to the test # in Table II.

TABLE II
RESULT FROM NAÏVE BAYES ALGORITHM

Test #	Train size	Test size	Stemming	Stop words	Accuracy
1	2400	600	Yes	Removed	72%
2	2400	600	No	Included	63.17%
3	1680	420	Yes	Removed	66%
4	1680	420	Yes	Included	66%
5	1680	420	No	Included	57.38%
6	1680	420	No	Removed	56%

TABLE III
CONFUSION MATRIX: TEST #1

	Actual label			
Prediction	110	17	0	25
	23	121	37	25
	1	3	103	1
	16	18	1	99

TABLE IV
CONFUSION MATRIX: TEST #2

	Actual label			
Prediction	9	0	0	0
	32	108	33	15
	0	3	119	1
	88	48	1	143

TABLE V
CONFUSION MATRIX: TEST #3

	Actual label			
Prediction	104	49	6	38
	0	31	11	2
	0	3	80	2
	6	21	5	62

TABLE VI
CONFUSION MATRIX: TEST #4

	Actual label			
Prediction	104	49	6	38
	0	31	11	2
	0	3	80	2
	6	21	5	62

TABLE VII
CONFUSION MATRIX: TEST #5

	Actual label			
Prediction	119	48	4	92
	1	44	28	4
	0	2	68	0
	0	0	0	10

TABLE VIII
CONFUSION MATRIX: TEST #6

	Actual label			
Prediction	119	54	5	93
	1	38	28	3
	0	2	67	0
	0	0	0	10

B. *KNN RESULTS*

In this subsection I will present the results obtained from the KNN algorithm experiments. A discussion on these results will take place in the Discussion section, section IV.

TABLE IX
TABLE OF ACCURACY PER K VALUE

K	Accuracy
1	57%
3	59.5%
5	60.5%
7	57.8%
9	57%
11	61.3%

TABLE X
TRAINING AND TESTING ERRORS PER K

K	Training Error	Testing Error
1	0.0000	0.43
3	0.2575	0.405
5	0.2996	0.395
7	0.3225	0.422
9	0.3412	0.430
11	0.337	0.387

TABLE XI
ACCURACY WHEN FEATURE IS REMOVED

K	WPL	UWPL	MWL	TR	Total
1	54.5%	57%	53%	50%	39.5%
3	57%	57%	55%	51.75%	41.5%
5	57%	58.75%	59%	53%	42.25%

The accuracies that appear in Table IX and XI are averages taken over 5 runs.

IV. DISCUSSION OF RESULTS

A. *NB*

The Naïve Bayes algorithm was implemented with the bag-of-words representation. Despite taking some time to build the dictionaries, the accuracy observed when training with the full dataset was good with 72%. However, using a smaller dataset seemed to knock the accuracy down quite a bit, by 6%. The use or absence of the stemming and stop word removal techniques also had substantial effects on the performance. The highest accuracy recorded was 72% which used both techniques. However, using the same size dataset but with neither technique brought the accuracy down by 8.83%. Stop words add redundant computation as they are the kinds of words that are highly likely to appear in every lyric while stemming places more emphasis on a single word rather than distributing it over a few words. Both techniques reduce the number of words all together and this helps to emphasise the uniqueness of each word which may be a testament to its genre. However, the results show that stemming plays a bigger role in improving the accuracy than stop words. The same accuracy was recorded for when stop word removal was both used and not used while stemming was applied. There is a 1.38% difference between 2 tests that both do not use stemming but one includes stop words and the other does not. This shows that stop words could be effective if they have a better definition. Stop words are not necessarily only of length 2 and, in the same sense, not all words of length 2 are stop words. Creating a dictionary of stop words then removing those only may be more robust. Furthermore, if stemming is applied, some words could be reduced down to 2 letters which then gets ignored.

The diagonal elements of the confusion matrices are the correctly classified test cases. The rows and columns go in the order: country, pop, rap and rock. On many instances, country songs have a 0 in either its column or row. This indicates that country songs are not predicted to be that genre or songs are not predicted to be country. In every case, 2-3 pop songs are predicted as rap songs. Although these two genres are quite close, as many pop songs feature rap artists and many rap artists produce pop-like songs, this is quite a low number. This may be the influence of the token ratio feature. Purely rap songs have numerous unique words while pop songs are often repetitive. This would lead to these two genres not having many mispredictions with each other. Table VII and Table VIII both do not apply stemming but Table VIII removes stop words. The results are almost identical. We can see that the removal of words with length 2 or smaller had no effect on the predictions. If every lyric contained the same stop words, it would be as if the stop words were not there in the first place. By this logic, removal of stop words may only increase the time performance as there will be less words, not the accuracy.

B. *KNN*

The accuracies of the KNN implementation are not too high. It steadily increases by an average of 1.75% for the

first 3 K's then drops back down for the next 2 K's then starts to rise again. Looking at Fig. 1, we can see that even for just 2 features, the data is clustered quite close together. This can explain the fluctuation in accuracies with changing K as in some cases, the most common class of the K nearest neighbours can change due to just one neighbour.

The training error is less than the testing error as the same data that was used to train the model is now assessing it. The model adapts to the training data and is likely to do well with what it has already seen. The training error is useful for model selection and in this case, the high training error rates is cause for concern.

The feature importance shows us something interesting. When the Total number of words feature is removed, the accuracy dramatically decreases in comparison to the other features. The accuracy never reached 50% without the Total words. This shows that the impact it has is significant. This is understandable when looking at the variation in the length of lyrics over the genres. Rap music has a larger emphasis on the words and not so much on the musical elements, thus it tends to have a larger amount of words. It is also fast paced so more can fit in a shorter time. Rock music often enjoys the musicality of the genre with its use of organic instruments and can compromise vocal time for more instrumental time. Table XII shows the maximum lyric length occurring in each genre. As we can see, there is a large variation between the genres. The feature with the second largest effect is the Token ratio. This feature compares the unique words to the total number of words in the song. Pop music often has a lot of repeated words which is appealing to commercial radio stations whereas rap music is more of a story being told so the more unique words the better. Rock music also tends to have more unique words as there are less instances of the chorus. These variations give the feature higher predictability. The removal of unique words per line does not seem to affect the accuracy by much. A reason for this could be that a single line of a song may be too granular to look at and that the feature consists of the mean unique words per line so it is likely that the numbers are very similar.

The KNN algorithm was easy to implement and did not take too long to run, however did not produce the greatest results. The best results were obtained using a small K that is greater than 1 , in our case 5. We were able to easily compute the feature importance. Improvements to the accuracy could be made by having more features such a sentiment score for the lyrics or non-lyrical features such as song duration.

TABLE XII
MAXIMUM LYRIC LENGTH PER GENRE

Genre	Max lyric length
Country	1368
Pop	2205
Rap	3469
Rock	608

C. *COMPARISON*

The top accuracies obtained by the KNN and Naïve Bayes algorithms are 61.3% and 72% respectively. The accuracy of KNN mostly stays around the late 50s and early 60s where Naïve Bayes only drops to below 60% when stemming is not applied. Although more tests could be run, it is safe to say that Naiv̈e Bayes did better than KNN. However, it is not just the algorithms that are comparable, but their features as well. Both use different representations of lyrics. Although the bag of words representation creates a high dimensional feature space, it proves to be efficient.

Discovering whether or not lyrical features, in particular the ones used in the KNN implementation, contain enough information to determine the genre of the song is a key question in this project. Although the accuracy may fall short of brilliance, it is high enough to consider these features as informative. In particular, the total number of words and the token ratio provide insights into the length of the lyric, which may be a proxy for the duration of the song, and the uniqueness of the words.

V. CHALLENGES AND IMPROVEMENTS

A Challenge that was met during this study was the slow running of algorithms. With a few tweaks, the efficiency was increased but not by enough to avoid a wait. Another challenge was the implementation of the Naïve Bayes algorithm and making sure it was working correctly. Web scraping also posed a challenge as on numerous attempts, the incorrect data was loaded. For example, Whenever songs by the rock band Queen were searched for, results for Beyonce were given. Web scraping was also a slow process.

To improve this study, techniques to obtain better accuracies should be tried. For the Naïve Bayes implementation, similar techniques to stemming such as lemmatization may be tested. Lemmatization reduces the word to a form that actually exists in the language where stemming may produce words that are not necessarily in the language. A more robust way to remove stop words should be employed, perhaps by using a library that contains known stop words.
Perhaps more features may be extracted from the lyrics, such as a 3 element sentiment score for the lyrics being negative, neutral or positive.

VI. CONCLUSION

THE classification of the genre of songs is a developing technique that is used in the music industry. Music streaming applications have databases of music in the millions and searching these databases needs to be robust. Words are powerful and carry a lot of information which is why the lyrics of songs can be used to help identify genres of songs, or even the artist who wrote it.

This project has implemented 2 machine learning techniques each using different word representations to try and predict the genre of a song solely based on its lyrics. The Naïve

Bayes model has proven to be more accurate than the KNN model with the bag-of-words representation. The KNN model, however, has proven that metadata of lyrics carry information that can be useful in the prediction of genres.

REFERENCES

[1] Michael A Casey, Remco Veltkamp, Masataka Goto, Marc Le-man, Christophe Rhodes, and Malcolm Slaney. Content-based music information retrieval: Current directions and future challenges.Proceedings of theIEEE, 96(4):668696, 2008.

[2] Walsten, D. and Orth, D., Song Genre Classification through Quantitative Analysis of Lyrics. Unpublished manuscript.

[3] Michael Fell and Caroline Sporleder. Lyrics-based analysisand classification of music. InProceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages620631, 2014.

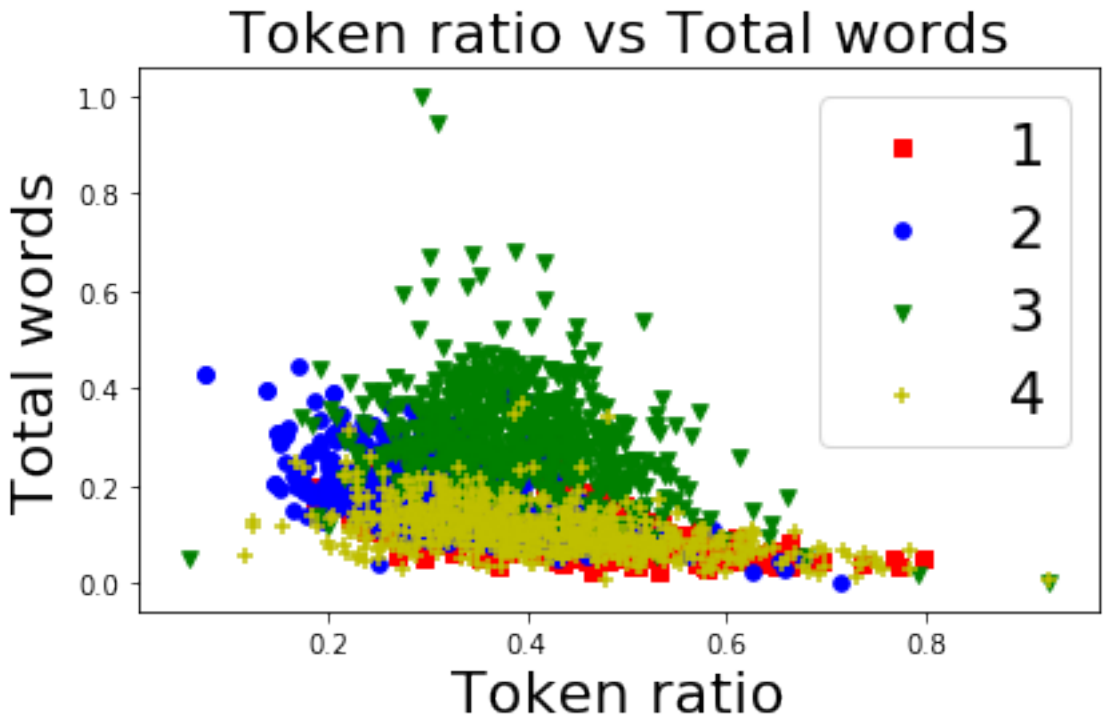

Fig. 1. A plot of the different genre's total words against Token ration