Jeffrey Levesque

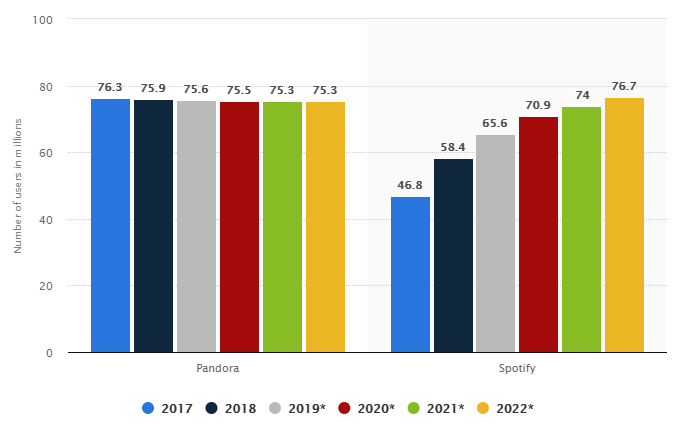
jlevesqu@syr.edu | Bernoulli and Multinomial Naïve Bayes in Sci-kit Learn

IST-736: HW6

PRofessor gates

**Introduction**

In October 23, 2001, Apple launched the revolutionary iPod. Customers were able to selectively purchase up to 1000-2000 songs on a single device. Furthermore, the ability to upload songs from CD's to the ipod, further justified this innovation eclipsing the CD player. While the iPod features were absorbed into the release of the iPhone, streaming radio became available on the same device. Individuals were able listen to music through multiple platforms, including streaming services. Services such as Pandora and Spotify are few or many popular to the general public.



**Figure 1:** Number of Spotify's and Pandora's active users from 2017 to 2022 (in millions)[[1]](#footnote-1).

Music has been a foundation bedrock in every culture, providing society a sense of identity and connection. Moreover, concepts of natural language processing (NLP) can determine which factors of music, define culture for a period time. For example, historical transformation of songs can be analyzed by lyrical content, then correlated to major sociopolitical events. Moreover, genre classification may serve as building blocks for historical analysis.

Companies such as iTunes Radio, Pandora, Spotify, and others may find trends of genre useful, for targeting audience. Though, user historical preferences would likely predict best, overall cultural trend could serve as an added factor, or a normalized facet, pervading in a more general way.

**Analysis**

Classification tasks utilized both unigram and ngram methodologies. For the unigram approach, the Multinomial Naïve Bayes (MNB), Bernoulli Naïve Bayes (BNB), and Support Vector Machine (SVM) were utilized. For the SVM, the linear kernel was selected due to high dimensional features, and for performance. Moreover, the unigram approach also utilized parts of speech (POS). Specifically, each word was suffixed by its corresponding (POS). Unlike the former methodologies, this approach did not use the Port stemmer.

Next, the ngram\_range=(1,3) was utilized for both the CountVectorizer[[2]](#footnote-2), and TfidfVectorizer[[3]](#footnote-3). This allows a range of ngram words to be utilized during the vectorize step. Like the unigram case, the MBN, BNB, and SVM models were used for the base case. However, the ngram was not utilized with the POS. Additionally, the kfold validation was utilized for each of the above implementations.

**Data Preparation**

Data was acquired from the 380,000+ lyrics from MetroLyrics Kaggle competition. Since the unzipped file was over 100MB github constraint, the csv file was split into 4 csv files. Each file contains increments of 95,000 entries, while the last lyrics4.csv retained the remainder. Furthermore, each csv contained the following columns:

* Index
* Song
* Year
* Artist
* Genre
* Lyrics

The unique values, or genres for each song include:

* Country
* Electronic
* Folk
* Hip-Hop
* Indie
* Jazz
* Metal
* Not Available
* Other
* Pop
* R&B
* Rock

Only the Genre (y), and Lyrics (X) columns were utilized. However, since computation was done on a local machine, to remain performant, a subset of lyrics1.csv was taken. Specifically, an equal sample distribution of Genre was taken:

df = df.groupby('genre').apply(lambda x: x.sample(500))

**Results**

The BNB ngram=(1,3) was less performant against the unigram, and POS methodologies:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 2:** BNB confusion matrix on genre prediction. | **Figure 3:** BNB with ngram\_range=(1,3) confusion matrix on genre prediction. | **Figure 4:** BNB with POS suffix confusion matrix on genre prediction. |

The unigram MNB with POS outperforms the ngram\_range=(1,3) and standard unigram variants:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 5:** MNB confusion matrix on genre prediction. | **Figure 6:** MNB with ngram\_range=(1,3) confusion matrix on genre prediction. | **Figure 7:** MNB with POS suffix confusion matrix on genre prediction. |

The SVM followed by the ngram\_range=(1,3) appear to outperform the POS variant.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 8:** SVM confusion matrix on genre prediction. | **Figure 9:** SVM with ngram\_range=(1,3) confusion matrix on genre prediction. | **Figure 10:** SVM with POS suffix confusion matrix on genre prediction. |

The train distributions indicate that each method was relatively balanced.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 11:** BNB train distribution. | **Figure 12:** BNB with ngram\_range=(1,3) train distribution. | **Figure 13:** BNB with POS suffix train distribution. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 14:** MNB train distribution. | **Figure 15:** MNB with ngram\_range=(1,3) train distribution. | **Figure 16:** MNB with POS suffix train distribution. |

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 17:** SVM train distribution. | **Figure 18:** SVM with ngram\_range=(1,3) train distribution. | **Figure 19:** SVM with POS suffix train distribution. |

The unigram BNB shows the best performance and least variability, followed by the BNB POS, and the BNB with ngram\_range=(1,3) with the greatest variability.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 20:** BNB kfold validation. | **Figure 21:** BNB with ngram\_range=(1,3) kfold validation. | **Figure 22:** BNB with POS suffix kfold validation. |

The unigram MNB shows the best performance, followed by the ngram\_range=(1,3) then MNB POS with the lowest score and highest variability.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 23:** MNB kfold validation. | **Figure 24:** MNB with ngram\_range=(1,3) kfold validation. | **Figure 25:** MNB with POS suffix kfold validation. |

Overall, the unigram SVM and SVM with ngram\_range=(1,3) seem to have the best performance, followed by the SVM POS having the highest score variability.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Figure 26:** SVM kfold validation. | **Figure 27:** SVM with ngram\_range=(1,3) kfold validation. | **Figure 28:** SVM with POS suffix kfold validation. |

The results portrayed by both the confusion matrices and kfold validation indicate the computed models are not very accurate. Models were predicted on a balanced 500 sample from a reduced subsample. Using an entire subsample, or the entire provided dataset, could produce substantial improvements. However, the SVM variants generally perform the best at nearly 30% accuracy. Since there are 12 dimensions, a long run random selection would likely have an 8% accuracy. Therefore, the SVM outperforms chance with the largest margin, followed by MNB, and BNB at last. Further exploration, and better ngram\_range selection would likely return improved prediction results.

**Conclusions**

Music has been around much longer than any established language, dating as far back pre-human societies. While music may be a cornerstone of cultural identity through the years, advancement in natural language processing with machine learning, can provide tools to further investigate. Specifically, studies can be formulated to identify whether patterns, and musical genres can both foretell, or simply foreshadow socio-political timelines. In a similar fashion, musical streaming providers, may further investigate whether users from geographic areas, are more predisposed and inclined to genres of music. This may paint an additional story regarding the sociology and transformation of current culture to that of the future.

While music generally serves as a form of entertainment, it is highly descriptive. The complexity of notes, and extent of vocabulary, may indicate societies cognitive focus. Furthermore, linguistic researchers have argued that a broader vocabulary may be an indication of higher intelligence. Though, it may not be clear that a similar argument can be made with musical lyrics, music preference and the associated genres can serve as a societal barometer.

1. <https://www.statista.com/statistics/293749/spotify-pandora-number-active-users/> [↑](#footnote-ref-1)
2. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html> [↑](#footnote-ref-2)
3. <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html> [↑](#footnote-ref-3)