Akshay Mendhakar / Mesian Tilmatine

Automatic Authorship Classification for German Lyrics Using Naïve Bayes

Abstract

Text classification is a prevalent and essential machine-learning task. Machine learning classifiers have developed immensely since their inception. The naïve Bayes classifier is one of the most prominent supervised machine learning classifiers. In this experiment, we highlight the performance of Naïve Bayes for classifying of authors/artists on the German lyrics corpus (“Songkorpus”) and compare the classification results with other classifier algorithms. The corpus of investigation consists of six artists with 970 songs in total. Bayes model evaluation measures revealed a precision of 0.91, recall of 0.94, and F1-measure of 0.9. Furthermore, the classification performance with other classifier algorithms did not reveal any statistically significant difference in performance. The results of the study add to the high volume of reports on the classification accuracy of Naive Bayes for the task of lyrical classification.

Keywords: German Lyrics, Text Classification, Naïve Bayes, Machine Learning

1 Introduction

Text mining methodologies have led rise to multiple applications such as text classification, regression, clustering, and association. In text classification, the desired categories are defined in advance, and records are classified into one or some among them (Kowsari et al., 2019). The popularity of text classification systems has grown drastically in the last two decades (Cichosz, 2014; Fell & Sporleder, 2014; Haggblade, Hong & Kao, 2011, Jiang et al., 2018; Kowsari et al., 2019). The application of text classification can be seen in use cases such as content moderation, sentiment classifier, product review classification, email spam classification etc. (Hu & Downie, 2009; 2010; Homem & Carvalho, 2011; Howard, Silla Jr & Johnson, 2011; Jiang et al, 2018.) The most common classifiers are Decision Tree, Perceptron, Naïve Bayes, Logistic Regression, Support Vector Machine, K-Nearest Neighbor, and Artificial Neural Networks (Khan, Baharudin, Lee & Khan, 2010). In the past decade, research in the field of song classification has received little focus (Mandel & Ellis, 2005). This can be accredited to the lack of standardized lyrical and audio datasets over the internet. Even though researchers can gather data from websites such as *www.azlyrics.com*, *www.songlyrics.com*, *www.lyrics.com*, etc., the need for large standardized datasets remains a significant issue in song classification. Research in the field of song classification can be noted to identify the genre (Mayer, Neumayer & Rauber, 2008), performers (Pettijohn & Sacco Jr, 2009), sentiment of the song (Logan, Kositsky, & Moreno, 2004; Yang & Lee, 2009), progression of a performer's career (Gomaa, 2022), language usage and geographic distribution (Jin & Ryoo, 2014; Pettijohn & Sacco Jr, 2009) etc.

Audio-based classifications focus on features (spectral and rhythmic), tempo, pitch, rhythm, loudness, etc. Classifiers based on textual lyrics focus on text features such as tokes (words, phrases & sentences), word frequencies, morpho-syntactic structures, rhyme patterns, etc. The performance of audio-based classifiers and text-based classifiers for song corpus has been tested empirically. Research reports on the automatic identification of Frederick Chopin's piano pieces were found to have a classification accuracy of 70% (Davis, 2018). In text-based systems, an accuracy ranging from 50-70 % has been reported in sentiment analysis to discover natural genre clusters (Logan, Kositsky, & Moreno, 2004). In contrast, an accuracy of 76% has been reported for lyric-based song sentiment classification using the sentiment vector space model (Yang & Lee, 2009). Similarly, combined audio and text-based classification systems methods have been reported to yield an accuracy ranging from 48.37% to 66.32% (Mayer, Neumayer & Rauber, 2008).

Some researchers have pointed out that the accuracy of song classifiers highly depends on the type of classifiers. In the study by Khan et al. (2010), the accuracy of the classification changes depending on the classifier used, i.e., for Support Vector Machines, it was 67 % to 97 %. In contrast, for Neural Networks, it improved from 76 % to 100 %, depending on genre.

Automatic Authorship classification has a rich research history and developmental trend. The main idea behind authorship attribution is that texts written by different authors can be distinguished by measuring statistical text features (Stamatatos, 2009). This field has developed rapidly with the development of machine learning classification techniques. Depending on the number of target classifications used to classify the dataset, different approaches can be used to perform the classification task. Decision trees and support vector machines are commonly used for binary classification (Elaidi et al., 2018). This constraint makes it difficult to apply these methods to tasks with more than two target classifications. In terms of obtaining a general toolkit, the naive Bayes classifier seems better suited for broader classification goals (Yang, 2018).

This study aims to test whether a naive Bayesian classifier can correctly predict song authors/artists based on lyrics alone. The used corpus of song lyrics ("Songkorpus"; Schneider, 2020) contains multiple linguistically motivated annotation layers (including POS and lemmatizations), but for this study, we only included plain text. The Naive Bayes Classifier was chosen because it seems well-suited for small datasets: our subcorpus comprises 970 text samples divided into six categories. The following article is organized as follows: The next section briefly describes some theories behind naive Bayesian classifiers. Section 3 describes the methods and measures used in our study. Section 4 discusses the results, and section 5 draws conclusions and provides future directions.

2 Theoretical framework of Naïve Bayes

A naive Bayes classifier is a type of probabilistic classification mechanism based on the Bayesian theorem, a posthumous theory by Thomas Bayes (Bayes & Hume, 1763; Tabak, 2004). Derived from the concepts of inferential statistics, Bayes' theorem serves as the basis for multiple machine learning models. The theorem is based on the logical probability of an event occurring concerning other events or features (Lewis, 1998). Equation (1) shows the Bayesian rule with P(A) & P(B) denoting the probability of an event A and event B respectively. Similarly, P(A|B) denotes the probability of A concerning B and vice versa in P(B|A).

P(A|B) = P(B|A) P(A) / P(B) **(1)**

Further, if we try to generalize the equation (1) for a series event represented by x and y, the equation becomes (2).

P(yi | x1, x2, …, xn) = P(x1, x2, …, xn | yi) * P(yi) / P(x1, x2, …, xn) **(2)**

In this estimates the prior P (yi) from the dataset is a conditional factor of the class P(x1, x2, …, xn | yi). This estimation is unviable if the sample size is small. Therefore, the dataset has to include a large number of samples which helps in the estimation of different possible combinations of a given value to predict its possibility. In this situation, where the number of observations in the dataset is growing, the application of the Bayes Theorem becomes difficult. In the case of variables being conditionally independent given the class, the estimation of the variable-value data is represented by the equation (3)

$$\mathrm{P}(\mathbf{x} \mid y) = \prod_{i=1}^{n} \mathrm{P}(x_i \mid y) \tag{3}$$

Here, n represents the number of variables in the sample and x_i is the ith value of the variable x. In situations where there are multiple classes, where we represent the number of classes with *k* and c_i as the ith class equation (3) is represented by equation (4). Thus we represent a classifier that is linear in nature.

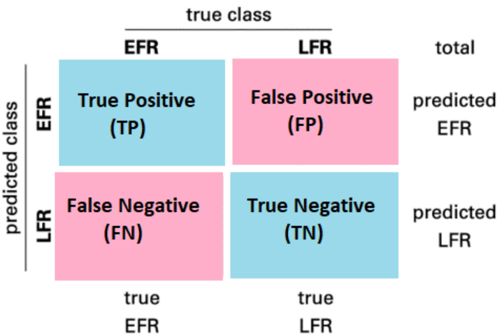
$$\mathrm{P}(\mathbf{x}) = \prod_{i=1}^{k} \mathrm{P}(c_i)\mathrm{P}(\mathbf{x} \mid c_i) \tag{4}$$

When the dataset contains categorical variables, frequency counts play a vital role in the estimation of the probabilities of P(y) and P(x_i | y). It involves methods like the Laplace estimation of the m-estimation method to measure the frequency. Further, this estimation can be compared and updated with new data as the training data is used as a single pass while training only. Therefore, this form of learning is supported by incremental learning. Similarly, when we look at numerical variables, discretization of the data is used. Therefore, the probability estimation is based on density estimation.

Naïve Bayes classifier utilizes the naïve Bayes theorem to solve a wide range of classification problems (Rish, 2001). Its computational efficiency and ease of implementation make it one of the most utilized supervised machine learning classification methods. Applications cover document classification (Ting et al., 2011), spam filtering (Metsis et al., 2006), content moderation (Risch, Ruff & Krestel, 2020), sentiment classification (Narayanan, Arora & Bhatia, 2013) and many more. The widespread acceptability of the naïve Bayes classifier is due to a wide range of factors such as its computational efficiency, low variance, its incremental learning abilities, strong aversion against missing values or high variability in the data.

Computational efficiency in modelling and prediction is an indisputable advantage over some other classification algorithms, which is due to its ability to parallelize data sets. i.e., the training time is linear for the number of training examples and the number of attributes and the classification time is linear for the number of attributes and is not affected by the number of examples studied. For the traits mentioned above, it would be helpful to add two more elements: resistance to over-equipping and the ability to manipulate multiple picks. naive Bayes operates on lower-order probability estimates derived from training data. They can easily be updated as new training data becomes available (Kohavi,1996). The classification results are prone to low variability with a high bias cost. It always uses all attributes for all predictions and is therefore relatively sensitive to noise in classified examples. Because it uses probability, it is also relatively insensitive to noise and missing values in the training data (Gama, Medas & Rodrigues, 2005).

In order to evaluate the performance of the naïve Bayes classifier model, we follow the results obtained from the confusion matrix (Figure 1). True positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) allow to compute Precision (PR), Recall (RE), Accuracy (CA), Error rate (ER) and F1 measures. The formulas are displayed in figure 1.



$$\mathrm{PR} = \frac{\mathrm{TP}}{\mathrm{TP+FP}}$$
$$\mathrm{RE} = \frac{\mathrm{TP}}{\mathrm{TP+FN}}$$
$$\mathrm{CA} = \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$$
$$\mathrm{F}_1 = \frac{\mathrm{2TP}}{\mathrm{2TP+FP+FN}}$$

Figure 1: Classical confusion matrix.

For any machine learning model, the F1 index is one of the most important metrics to determine its performance (Lipton, Elkan & Narayanaswamy, 2014). The value of the F1 index ranges from 0 to 1, where 0 is the worst possible score with poor classification. Another performance metric is the Receiver Operating Characteristics (ROC) curve (figure 2). It graphically determines classification performance of a binary classifier as a function of TP and FP measures.

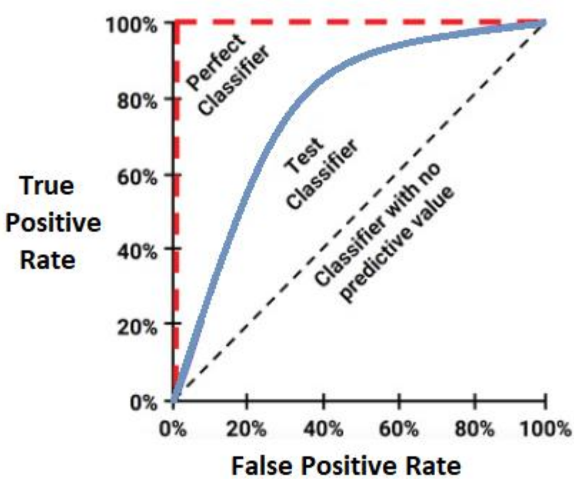


Figure 2: ROC curve.

In figure 2, the dotted diagonal signifies the zero thresholds or no classification. The blue line measures the true performance of the classifier with respect to the red dotted line which symbolizes perfect classification with 100% accuracy (100% true positives and 0% false negatives). In uniformly distributed datasets, measuring the accuracy of the classifier is enough to predict the classifier's performance. Whereas with imbalanced datasets, ROC AUC may be more significant. ROC AUC considers the trade-offs between precision and recall, while accuracy only determines how many predictions are correct. Generally, AUC is preferred over accuracy as it is a much better indicator of model performance.

3 Experiments

We conduct a series of experiments focusing on authorship attribution for song lyrics of Udo Lindenberg, Konstantin Wecker, Stoppok, Ulla Meinecke, Hannes Wader, and Fettes Brot. The results of naïve Bayes and other classification approaches are contrasted.

The first half of the experiment was focused on applying the naïve Bayes classifier to the songkorpus dataset. The publicly available repository provides detailed corpus statistics, as well as visualizations on character, word, verse, song and corpus level. Our first step was data gathering and pre-processing from the songkorpus website. The data came in XML format and was further transformed to plain text. The processed text included semantic and structural information related to the artist and the song verses. All additional information was omitted. The plain textual data was subject to further linguistic analysis, using the Profiling UD tool (Brunato et al., 2020). Profiling UD extracts 130 computational linguistic parameters under raw textual features, morphosyntactic and syntactic parameters. A similar protocol as noted in (ref. Mendhakar, 2022) was used to extract and process the Linguistic features extracted from the tool. The extracted parameters were tabulated into an excel file. By feature reduction, pruning of the number of features was carried out. The resultant dataset consisted of 970 data points of 115 parameters categorized under six different artists. After initial dataset creation, randomization of rows was made. The basic demographics of the dataset created are highlighted in table 1. Table 1 represents the database representation of each artist.

Artist	Songs considered	Representation in the dataset
Fettes Brot	91	9.38%
Udo Lindenberg	316	32.58%
Ulla Meinecke	78	8.04%
Stoppok	77	7.94%
Hannes Wader	168	17.32%
Konstantin Wecker	240	24.74%

Table 1: Description of the dataset considered in the study.

All experiments were conducted on a system with an Intel Corei7 CPU at 2.4GHz, 8 GB of RAM, and 1 TB of secondary storage, running windows 10 and MATLAB 2021b. The dataset was loaded onto the machine learning toolkit of MATLAB software for further processing. The utility of MATLAB's machine learning toolkit was due to the capability of comparing multiple classifiers in one place and also due to its ease of implementation. The classical naïve Bayes classifier was designed with the preset features for classification. The developed dataset was split into a training and testing dataset. Two-thirds of the dataset was used to train the classifier and the rest of the data was used for its testing and validation. The split of the dataset was randomized to eliminate any artist bias. The parameters were tweaked and multiple runs were carried out to find the best possible classification accuracy.

In the second stage of the experiment, the classification accuracy of the naïve Bayes classifier was compared with other commonly used classifiers, such as logistic regression (LR), support vector machines (SVM), naive Bayes (NB), decision tree classifier (DTC), K-nearest neighbor (KNN), and neural networks (NNs). To improve the classification accuracy, hyper-parameter tuning and dimension reduction were employed. Additionally, multiple iterations of the classifier parameters were run by removing correlated features, using log probabilities in calculations, and parallelized calculations were performed. These additional steps were used in order to identify the best possible classification results. The performance of each classifier is compared in the next section.

4 Results and discussion

When measuring performance of the Naive Bayes classifier, various iterations were carried out by implementing the Laplace smoothing, with the classifier's accuracy being the best at the estimator's value of 0.06. Figures 3 & 4 display the confusion matrix and ROC, respectively, of the best naïve Bayes classifier. Table 2 summarizes accuracy, error rate, precision, recall and F-measure across each class of the dataset.

Class	Accuracy(CA)	Precision (PR)	Recall (RE)	F1 Score
1	97.94 %	0.89	0.89	0.89
2	93.40 %	0.92	0.88	0.9
3	97.73 %	0.89	0.82	0.85
4	97.22 %	0.85	0.79	0.82
5	94.95 %	0.83	0.89	0.86
6	95.26 %	0.88	0.93	0.91
Total	96.08 %	0.88	0.87	0.87
Total with 0.06 Laplace	**97.03 %**	**0.91**	**0.94**	**0.90**

Table 2: Evaluation of the Naïve Bayes classifier.

In figure 4, the ROC curve of the classifier shows that the area under the curve (AUC) was 0.89, which is a very good classifier performance.

	Class 1	**Class 2**	**Class 3**	**Class 4**	**Class 5**	**Class 6**	**TPR**	**FNR**
Class 1	81	5	0	3	1	1	89%	11%
Class 2	3	277	3	2	5	12	88%	12%
Class 3	1	4	64	1	2	0	82%	18%
Class 4	1	6	2	61	2	0	79%	21%
Class 5	4	14	5	4	149	3	89%	11%
Class 6	1	10	4	6	9	224	93%	7%

Figure 3: Confusion matrix plot of the Naïve Bayes classifier.

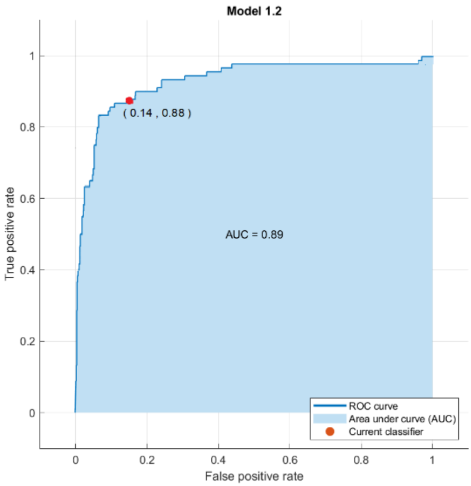


Figure 4: ROC curve of the Naïve Bayes classifier.

Even with the application of Laplace smoothing, the accuracy of the classifier did not change significantly. Therefore, the naïve Bayes classifier build in this experiment has an overall accuracy of 97.03 %

4.1 Comparing the performance of different classifiers

To rank the performance of our classifiers, we apply different classifiers to the dataset. Table 3 show that Naïve Bayes had the best performance and obtained better results for almost all metrics. The average accuracy rate of Naïve Bayes for the test set is 91% and its highest accuracy is 97%. However, the comparison of training and test set accuracies indicates that Naïve Bayes and especially RF suffer from overfitting problems. Decision Trees show the worst performance, deep learning algorithms like neural networks (ANN, CNN and LSTM) take the most execution time, with LSTM being the slowest. One outstanding point is that CNN performed very well and was much faster than ANN and LSTM.

Algorithms	Precision (%)	Recall (%)	F1-Score (%)	Training Set Accuracy (%)	Test Set Accuracy (%) (Avg/Highest)
Naïve Bayes	**91**	**94**	**90**	93	91/97
SVM	73	72	72	70	67/73
RF	80	79	79	100	76/79

ANN	77	78	77	83	82/82
LSTM	87	86	84	88	81/87
CNN	91	92	91	95	91/96

Table 3: Comparison of different classifiers of the study.

5 Conclusion

Creating a meaningful lyrics dataset is a tedious and time-consuming task. For example, guest appearances by other artists or two versions of the same song (e.g. studio version and live version) must be handled with care. By using the precompiled Songkorpus, we empirically tested the accuracy of different authorship classificators on a reliable dataset. The results of our best model seems promising and are in accordance with comparable research reports on naïve bayes classifiers (Rish, 2001; Dai et al., 2007; Labatut & Cherifi, 2012; Nitze, Schulthess & Asche, 2012; Altheneyan & Menai, 2014; Baron, 2016; Shih, Stow, & Tsai, 2019). It can be concluded from our experiments that the Naive Bayes classifier seems to be a good choice for authorship attribution of song lyrics, at least for the investigated singer-songwriter dataset. Since the used dataset is relatively small, it would be a reasonable choice to use our classifiers on bigger datasets in order to make better generalizations.

6 References

Altheneyan, A. S., & Menai, M. E. B. (2014). Naïve Bayes classifiers for authorship attribution of Arabic texts. *Journal of King Saud University-Computer and Information Sciences*, *26*(4), 473-484.
Baron, G. (2016). Comparison of cross-validation and test sets approaches to evaluation of classifiers in authorship attribution domain. In *International Symposium on Computer and Information Sciences* (pp. 81-89). Springer, Cham.
Bayes, T., & Hume, D. (1763). Bayes's Theorem. In *Proceedings ofthe British Academy* (Vol. 113, pp. 91-109).
Brunato, D., Cimino, A., Dell'Orletta, F., Venturi, G., & Montemagni, S. (2020). Profiling-ud: a tool for linguistic profiling of texts. In *Proceedings of The 12th Language Resources and Evaluation Conference* (pp. 7145-7151).
Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, *81*(3), 429-450.
Cichosz, P. (2014). Data mining algorithms: explained using R. John Wiley & Sons.
Cunningham, S. J., Bainbridge, D., & Falconer, A. (2006). " More of an art than a science": Supporting the creation of playlists and mixes.
Dai, W., Xue, G. R., Yang, Q., & Yu, Y. (2007). Transferring naive bayes classifiers for text classification. In *AAAI* (Vol. 7, pp. 540-545).
Davis, A. (2018). Classical Composer Identification on Interval Features for CS230-Spring 2018. Retrieved from http://cs230.stanford.edu/projects_spring_2018/reports/8289789.pdf

Elaidi, H., Elhaddar, Y., Benabbou, Z., & Abbar, H. (2018). An idea of a clustering algorithm using support vector machines based on binary decision tree. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1-5). IEEE.

Fell, M., & Sporleder, C. (2014). Lyrics-based analysis and classification of music. In Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers (pp. 620-631).

Fukunaga, K., & Hayes, R. R. (1989). Estimation of classifier performance. *IEEE transactions on pattern analysis and machine intelligence*, *11*(10), 1087-1101.

Gama, J., Medas, P., & Rodrigues, P. (2005). Learning decision trees from dynamic data streams. In *Proceedings of the 2005 ACM Symposium on Applied computing* (pp. 573-577).

Gomaa, W. (2022). Lyrics Analysis of the Arab Singer Abdel ElHalim Hafez. *Transactions on Asian and Low-Resource Language Information Processing*.

Haggblade, M., Hong, Y., & Kao, K. (2011). Music genre classification. *Department of Computer Science, Stanford University*.

Homem, N., & Carvalho, J. P. (2011). Authorship identification and author fuzzy "fingerprints". In *2011 Annual Meeting of the North American Fuzzy Information Processing Society* (pp. 1-6). IEEE.

Howard, S., Silla Jr, C. N., & Johnson, C. G. (2011). Automatic lyrics-based music genre classification in a multilingual setting. In *Proceedings of the Thirteenth Brazilian Symposium on Computer Music*.

Hu, X., & Downie, J. S. (2010). When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis. In *ISMIR* (pp. 619-624).

Hu, X., Downie, J. S., & Ehmann, A. F. (2009). Lyric text mining in music mood classification. *American music*, *183*(5,049), 2-209.

Idicula-Thomas, S., Kulkarni, A. J., Kulkarni, B. D., Jayaraman, V. K., & Balaji, P. V. (2006). A support vector machine-based method for predicting the propensity of a protein to be soluble or to form inclusion body on overexpression in Escherichia coli. *Bioinformatics*, *22*(3), 278-284.

Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y., & Guan, R. (2018). Text classification based on deep belief network and softmax regression. *Neural Computing and Applications*, *29*(1), 61-70.

Jin, D. Y., & Ryoo, W. (2014). Critical interpretation of hybrid K-pop: The global-local paradigm of English mixing in lyrics. *Popular Music and Society*, *37*(2), (pp. 113-131).

Juba, B., & Le, H. S. (2019). Precision-recall versus accuracy and the role of large data sets. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 33, No. 01, pp. 4039-4048).

Khan, A., Baharudin, B., Lee, L. H., & Khan, K. (2010). A review of machine learning algorithms for text-documents classification. *Journal of advances in information technology*, *1*(1), 4-20.

Kohavi, R. (1996). Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid. In *Kdd* (Vol. 96, pp. 202-207).

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, *10*(4), 150.

Labatut, V., & Cherifi, H. (2012). Accuracy measures for the comparison of classifiers. *arXiv preprint arXiv:1207.3790*.
Lewis, D. D. (1998). Naive (Bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning* (pp. 4-15). Springer, Berlin, Heidelberg.
Lipton, Z. C., Elkan, C., & Narayanaswamy, B. (2014). Thresholding classifiers to maximize F1 score. *arXiv preprint arXiv:1402.1892*.
Logan, B., Kositsky, A., & Moreno, P. (2004). Semantic analysis of song lyrics. In *2004 IEEE International Conference on Multimedia and Expo- ICME* (pp. 827-830).
Mandel, M. I., & Ellis, D. P. (2005). Song-level features and support vector machines for music classification. (pp. 594-599).
Mayer, R., Neumayer, R., & Rauber, A. (2008). Rhyme and Style Features for Musical Genre Classification by Song Lyrics. In *Ismir* (pp. 337-342).
Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes?. In *CEAS* (Vol. 17, pp. 28-69).
Narayanan, V., Arora, I., & Bhatia, A. (2013). Fast and accurate sentiment classification using an enhanced Naive Bayes model. In *International Conference on Intelligent Data Engineering and Automated Learning* (pp. 194-201). Springer, Berlin, Heidelberg.
Nitze, I., Schulthess, U., & Asche, H. (2012). Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification. *Proceedings of the 4th GEOBIA, Rio de Janeiro, Brazil*, *79*, 3540.
Pettijohn, T. F., & Sacco Jr, D. F. (2009). The language of lyrics: An analysis of popular Billboard songs across conditions of social and economic threat. *Journal of language and social psychology*, *28*(3), (pp. 297-311).
Risch, J., Ruff, R., & Krestel, R. (2020). Offensive language detection explained. In *Proceedings of the second workshop on trolling, aggression and cyberbullying* (pp. 137-143).
Rish, I. (2001). An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence* (Vol. 3, No. 22, pp. 41-46).
Salley, K. (2011). On the interaction of alliteration with rhythm and metre in popular music. *Popular Music*, *30*(3), 409-432.
Schneider, R. (2020). A corpus linguistic perspective on contemporary German pop lyrics with the multi-layer annotated "Songkorpus". In *Proceedings of the 12th Language Resources and Evaluation Conference* (pp. 842-848).
Shih, H. C., Stow, D. A., & Tsai, Y. H. (2019). Guidance on and comparison of machine learning classifiers for Landsat-based land cover and land use mapping. *International Journal of Remote Sensing*, *40*(4), 1248-1274.
Stamatatos, E. (2009). A survey of modern authorship attribution methods. *Journal of the American Society for information Science and Technology*, *60*(3), 538-556.
Stamatatos, E., & Widmer, G. (2002). Music performer recognition using an ensemble of simple classifiers. In *ECAI* (pp. 335-339).
Tabak, J. (2004). Probability and Statistics: The Science of Uncertainty. Facts on File. *Inc., NY, USA*, 46-50.

Ting, S. L., Ip, W. H., & Tsang, A. H. (2011). Is Naive Bayes a good classifier for document classification. *International Journal of Software Engineering and Its Applications*, *5*(3), 37-46.

Wickramasinghe, I., & Kalutarage, H. (2021). Naive Bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, *25*(3), 2277-2293.

Yang, D., & Lee, W. S. (2009). Music emotion identification from lyrics. In *2009 11th IEEE International Symposium on Multimedia* (pp. 624-629).

Yang, F. J. (2018). An implementation of naive bayes classifier. In 2018 International conference on computational science and computational intelligence (CSCI) (pp. 301-306). IEEE.

Correspondence

Akshay Mendhakar
University of Warsaw, Faculty of Applied Linguistics, Poland
University of Vienna, Empirical visual aesthetics lab, Austria
a.mendhakar@uw.edu.pl

Mesian Tilmatine
Free University of Berlin, Department for Experimental and Neurocognitive Psychology
Radboud University Nijmegen, Centre for Language Studies, The Netherlands
m.tilmatine@fu-berlin.de

NOTE: The authors are employed under the ELIT network. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860516.