Lyrics-Based Song Genre Classification

Johannes Scherer
Computer Science @ {NTNU, Universität Ulm} / TDT4310, 2022
johasche@stud.ntnu.no
https://github.com/josch14/song-genre-classification-with-lyrics

Abstract

Musical genres are essential for organizing songs into musical collections and providing well-functioning music recommendation and retrieval. In order to support these methods, songs need to be tagged with their appropriate genre(s). Annotation of genres by humans is time-consuming and costly, while reliable automatic song genre classification is difficult, especially because the boundaries between musical genres are not clearly defined. Thus, song genre classification remains a challenging topic. To this end, we target this task by only using song lyrics. For this, we implement both, traditional and machine learning text classification methods. Furthermore, we investigate how the classification performance of all methods depends on the number of considered genres in the dataset. Our experiments show that the classification performance of text classification methods degrades for increasing number of considered genres. The best results were consistently achieved using the Bernoulli Naive Bayes classifier.

1 Introduction

With the already existing and constantly rising amount of multimedia data available in digital format, it is important to provide tools to efficiently browse, search, and find desired data (Haase (2004)). For example, to find songs, music recommendation and retrieval systems typically rely on automatic music analysis such as artist and song similarity, or automatic genre classification (Fell and Sporleder (2014)). In general, manual annotation of such data by humans usually provides higher quality data, but is time-consuming and costly. Despite the challenges it provides, well-functioning text classification can provide huge upsides, which justifies the importance of this task.

When played on a local device or via a streaming platform, songs, or music more generally, are mainly audio data. In addition to that, songs typically come with metadata containing further information such as artist, album, and release date. The song lyrics, however, usually do not come with the rest of the data, which explains the popularity of websites such as `Genius` that provide song lyrics. So when predicting the musical genres of songs using their lyrics, we have to keep in mind that we are not working with the raw data of the medium, but rather with a different representation of it. More generally, for song genre classification different types of features can be used: the audio signal of the song itself (consisting of parameters like frequency, decibel, bandwidth etc., Chillara et al. (2019)), the corresponding lyrics, additional information such as artist and release date, or any combination of such. Interestingly, Fell and Sporleder (2014) argue that it is worthwhile to look at lyrical properties when classifying music, and observed that classifiers that incorporate textual features usually outperform audio-only classifiers.

Although songs do not necessarily belong to a single musical genre, we approach song genre classification as a multi-class, single-label text classification task. For text classification, numerous methods have been proposed in literature, which can be divided into traditional methods (meaning statistics-based models), and machine learning methods (Li et al. (2020)). In this work, we target the task of song genre classification with both, traditional methods text classification methods (Naive Bayes (NB) and Support Vector Machines (SVMs)), and machine learning methods. More precisely, we use a bidirectional LSTM (Long Short-Term Memory) model. For LSTMs, Sari et al. (2019) showed that GloVe (Global Vectors for word representation) embeddings can improve text classification performance. We use GloVe to examine whether this is also the case for song genre classification. Furthermore, we are interested in which

classification performance can be achieved for datasets that contain songs from different numbers of genres. Therefore, we train and evaluate all methods on a total of 11 datasets containing the lyrics of at least 2, and up to 12 musical genres.

2 Related Work

Works in song genre classification extract a wide variety of features from both, the audio signal and additional data, and forward these features to different classification models. Goel et al. (2014) predict potentially multiple genres of a song by obtaining a set of features like beats/tempo, energy, loudness, speechiness, valence, danceability, acousticness, discrete wavelet transform etc., and feed those into a parallel multi-layer perceptron network. McKay et al. (2010) investigated the utility of combining four different features types, extracted from (1) lyrical, (2) audio, (3) symbolic, and (4) cultural sources for song genre classification. First, they found that when using the feature types separately from each other, then worst accuracies were achieved when only using the lyrical features. Furthermore, they found that cultural features consisting of information extracted from both, web searches and mined listener tags, were particularly effective. Finally, they found that, with some exceptions, combining feature types does improve classification performance. Fell and Sporleder (2014) made a similar observation, stating that classifiers which incorporate textual features usually outperform audio-only classifiers.

There are several works that approach song genre classification as a text classification task using only song lyrics. Traditional text classification approaches have utilized n-gram models and algorithms such as Naive Bayes and SVM, while the use of deep learning methods such as Recurrent Neural Networks or Convolutional Neural Networks has produced superior results (Tsaptsinos (2017)). For the classification of songs into four musical genres, Kumar et al. (2018) apply two word embeddings, namely Word2Vec and Word2Vec with TF-IDF (Term Frequency–Inverse Document Frequency), to map lyrics words into vectors consisting of real numbers, and apply multiple classification methods such as SVM, Random Forest, XGBoost (eXtreme Gradient Boosting) and Deep Neural Networks to predict genres. They found that, for all methods, using Word2Vec with TF-IDF outperforms simple Word2Vec. Finally, Tsaptsinos (2017) argues that, since the words of a song combine to lines, multiple lines combine to segments (e.g., chorus), and segments form a complete song, lyrics exhibit a hierarchical layer structure. Because of that, he adapts a Hierarchical Attention Network (HAN) to exploit these layers and learn the importance of words, lines, and segments. By testing their model over a 117-genre dataset and a reduced 20-genre dataset, he showed that HAN outperforms both non-neural models and simpler neural models, whilst also classifying over a higher number of genres than previous research. However, he argues that in order to produce a state-of-the-art classifier, it is evident that the classifier must take into account more than just the lyrical content of the song.

3 Lyrics-Based Song Genre Classification

We approach song genre classification as a multi-class, single-label text classification task, using only song lyrics as input data. We use a total of six different models, three traditional methods ((1) to (3)) and three machine learning methods ((4) to (6)). This enables us later to discuss whether the more elaborate training of machine learning methods is justified by achieving better classification performances. In the following, we present the different methods and show which preprocessing steps were used for each method.

Naive Bayes (NB) is the simplest and most broadly used technique for constructing classifiers (Li et al. (2020)). Based on applying Bayes' theorem, Naive Bayes algorithms work under the assumption that when the target class is given, the textual features are independent of each other. By utilizing this assumption, for a textual input, Naive Bayes algorithms predict class membership probabilities based on the words occurrences in the text.

(1) Bernoulli Naive Bayes: We use the Bernoulli NB classifier, which uses binary term occurrence features, i.e., whether a word occurs in the song lyrics or not. Here, we preprocess each song lyrics by putting it in lower case, removing symbols, and resolving contractions (e.g., *they're* to *they are*).

(2) Multinomial Naive Bayes: We use the Multinomial NB classifier, which works on the concept of term frequency, i.e., how often a word occurs in a text (Singh et al. (2019)). For term frequency extraction, we make use of TF-IDF. TF-IDF rescales the frequency of words by how often they appear in all lyrics so that the scores for highly frequent words across all lyrics are penalized (Kumar et al. (2018)). We preprocess each song lyrics by putting it in lower case, removing symbols, resolving contractions, and removing stop words[1].

Joachims (1999) first used Support Vector Machines (SVMs) for text classification. SVM-based approaches, which are binary classifiers by default, turn text classification tasks into multiple binary classification tasks (Colas and Brazdil (2006)). In short, SVM constructs an optimal hyperplane in the feature space by maximizing the distance between the hyperplane and the two categories of the training set(s), which results in the lowest error rate of classification.
(3) SVM: We use a Support Vector Classification (SVC) method, i.e., a SVM for text classification. We use a linear kernel. For SVM, we preprocess each song lyrics by putting it in lower case, removing symbols, and resolving contractions.

Recurrent Neural Networks (RNNs) are widely used in Natural Language Processing (Pouyanfar et al. (2018)). Unlike traditional neural networks, RNNs use the sequential information of texts to understand the context in which words occur. RNNs suffer under the vanishing gradient problem, which the Long Short-Term Memory (LSTM) explicitly aims to solve (Li et al. (2020)). Furthermore, bidirectional LSTMs (BiLSTM), consist of two LSTMs to further increase the available contextual information: one taking the input in a forward direction, and the other in a backwards direction. GloVe (Global Vectors for word representation), as introduced by Pennington et al. (2014), is an unsupervised learning algorithm for obtaining vector representations for words. For LSTMs, Sari et al. (2019) showed that GloVe embeddings can improve text classification performance. For all following presented machine learnings approaches, we preprocess each song lyrics by putting it in lower case, removing symbols, resolving contractions, and removing stop words. In our experiments, using these preprocessing steps resulted in the best performances for each method. Furthermore, for each method, we vectorize the processed text inputs by turning each text into a sequence of integers.
(4) Averaged GloVe + Output Layer: For the first machine learning classification method, we use GloVe word embeddings in combination with a dense output layer. For an input text, we first compute the average GloVe vectors of all words in the text. We use this vector as the feature vector of the given song lyrics, and add a single output layer with softmax activation. Finally, we implement dropout between the feature vector and the output layer. For this model, we use 50-dimensional GloVe[2] word vectors and keep their weights trainable.
(5) LSTM: We use a bidirectional LSTM with 64 units, add a hidden dense layer with 64 neurons and ReLU activation, and finally an output layer with softmax activation. We implement dropout between the three layers.
(6) LSTM + GloVe: Finally, instead of feeding the input sequence directly to the LSTM, we use each word's corresponding GloVe vector. Other than that, we use exactly the same architecture as the LSTM. For this model, we use 100-dimensional GloVe word vectors and keep their weights trainable.

4 Experimental Apparatus

We train each of the proposed text classification methods separately on multiple datasets, which differ in the number of considered genres (see Section 4.1), and evaluate and compare each method by collecting multiple metrics (see Section 4.3). We implemented all models in Python, using Scikit-learn[3] for the

[1]Note that we remove stop words for Multinomial NB, where we use TF-IDF that usually penalizes stop words, but do not remove stop words for Bernoulli NB, which works with word occurrences. Although this might not sound intuitive, we achieved the results for these methods when doing so.

[2]GloVe is available at: `https://nlp.stanford.edu/projects/glove/`.

[3]Scikit-learn is available at: `https://github.com/scikit-learn/scikit-learn`.

Index	Genre	# Songs	%	Cumulative # Songs	Average Tokens/Song
1	Pop	2,508	8.43	2,508	245.8
2	Rock	4,617	15.52	7,125	174.2
3	Rap	2,004	6.74	9,129	450.6
4	Country	4,762	16.01	13,891	179.3
5	Reggae	1,230	4.13	15,121	209.6
6	Heavy Metal	4,385	14.74	19,506	150.2
7	Blues	1,055	3.55	20,561	152.7
8	Indie	4,244	14.27	24,805	154.5
9	Hip Hop	1,105	3.71	25,910	438.9
10	Jazz	1,348	4.53	27,258	119.5
11	Folk	1,066	3.58	28,324	171.5
12	Gospel/Religioso	1,423	4.78	29,747	143.5
		29,747	100%	-	199.8

Table 1: Song genres, their frequency of occurrence and the average number of tokens they contain in our dataset(s). For example, the dataset consisting of songs from 5 genres includes the genres Pop, Rock, Rap, Country and Reggae, and consists of 15,121 songs.

traditional classification methods, and TensorFlow[4] for the machine learning models.

4.1 Datasets

We build new dataset(s) based on a Kaggle[5] dataset which contains song lyrics of 79 genres and 4,168 artists, scraped from `Vagalume`. In this dataset, each artist belongs to potentially multiple genres, and all of an artist's songs belong to the very same genre(s). Based on this dataset, we filter out all non-English songs using 1) the corresponding attribute that is provided by the dataset and 2) using a spaCy language parser[6] in combination with a language detection pipeline[7]. Furthermore, song lyrics typically have lines that contain information about the artist of a song (e.g., which artist sings the following passage), and whether the following passage is a chorus or an intro/outro. We remove such information to make sure that the classification methods focus on actually spoken words. To make sure that we have a valid dataset for our targeted multi-class, single-label classification problem, we remove the artists/songs that belong to more than one musical genre. We decided to include a total of 12 genres in our dataset(s), which consist a total of 29,747 songs (see Table 1).

Based on this processed dataset, we are now able to create smaller datasets that contain the lyrics of only some of the 12 genres. Following the genre order that is provided in Table 1, we create a total of 11 datasets containing the lyrics of at least two, and up to 12 genres. For example, the dataset that contains the lyrics of five genres consists of 15,121 songs of the genres Pop, Rock, Rap, Country, and Reggae. We split each dataset up into train/test splits of 0.7/0.3. Here, we make sure that, e.g., for the Pop genre all datasets contain the same song lyrics in the train and the test split, and that the genre distribution is the same for the train and test split.

4.2 Procedure and Hyperparameters

We train and evaluate all presented text classification methods for each of the 11 datasets, i.e., for iteratively increasing number of genres (and number of songs). This makes hyperparameter optimization

[4]TensorFlow is available at: `https://github.com/tensorflow/tensorflow`.

[5]Kaggle dataset *"Song lyrics from 79 musical genres"* is available at: `https://www.kaggle.com/datasets/neisse/scrapped-lyrics-from-6-genres`.

[6]The spaCy model that we use is available at: `https://github.com/explosion/spacy-models/releases/tag/en_core_web_sm-3.3.0`.

[7]The language detection pipeline is available at: `https://github.com/Abhijit-2592/spacy-langdetect`.

difficult, as each of the methods potentially has different optimal parameters for each dataset size. Therefore, for each method, we performed grid search for the dataset that contains the song lyrics of 6 genres, and use the optimal hyperparameters for the training on all other datasets. Tests lead to the conclusion that the methods' specified optimal hyperparameters are also most likely the optimal hyperparameters for the other datasets.

For Naive Bayes (Bernoulli and Multinomial) we found the optimal value for the Laplace/Lidstone smoothing parameter α to be 0.05. For SVM, adjusting the regularization parameter C did not lead to any improvement, therefore we left it at 1.00.

For all, Averaged GloVe + Output Layer, LSTM, and LSTM + Glove, we only keep the 10,000 most common tokens for each dataset and, after applying each method's specific preprocessing (see Section 3), truncate the lyrics of all songs to 256 word tokens. We tested different batch sizes, dropout rates, and learning rates. However, using a batch size of 32, dropout rate of 0.2, and a learning rate of 1e-3 was optimal for all three methods. For each method, we use Adam optimizer and cross-entropy loss, and use class weights to help the models to learn on the imbalanced data of our datasets (see Table 1). We train each method with early stopping, i.e., as long as the validation accuracy does not improve for 10 epochs. For the remaining parameters that concern the model architecture itself, please refer to Section 3.

4.3 Measures and Metrics

Precision, recall, and F1 score are vital metrics for the evaluation of (single-label, multi-class) classification methods on unbalanced datasets (Li et al. (2020)). We report weighted (with weights equal to the genre probability) and macro-averages of precision, recall, and F1 score across all genres. This includes the calculation of accuracy, as this is equal to weighted recall.

5 Results

Table 2 shows the macro- and weighted averages of precision, recall, and F1 score performances of all methods on the whole dataset (consisting of song lyrics from 12 musical genres). We assume macro-average F1 score to be the metric that best measures the overall classification performance for our unbalanced dataset as it treats all classes equally. The most noteworthy observation is that the best classification performance (56.70 macro-average F1 score) is achieved when using Bernoulli NB. This does not only apply to macro-average F1 score, but for all metrics except macro-average precision. The best macro-average precision performance of 63.77 is achieved using SVM. Unexpectedly, all machine learning methods perform worse than all three traditional methods. Furthermore, using LSTM in combination with GloVe Embeddings could not significantly improve performance compared to using the LSTM only. In general, the LSTM models were outperformed by all other methods, also by the simpler machine learning model Averaged GloVe + Output Layer.

	Macro-Average			Weighted Average		
Method	Prec	Rec	F1	Prec	Acc/Rec	F1
Naive Bayes (Bernoulli)	58.44	**58.03**	**56.70**	**59.57**	**58.24**	**57.94**
Naive Bayes (Multinomial)	56.51	56.35	55.72	57.16	56.80	56.43
SVM	**63.77**	50.06	53.29	58.53	55.80	55.34
Averaged GloVe + Output Layer	53.27	56.51	54.14	56.37	55.62	55.42
LSTM	44.86	45.62	45.06	47.49	47.51	47.33
LSTM + GloVe	46.23	45.92	45.94	48.07	47.85	47.87

Table 2: Experimental results for lyrics-based song genre classification on the whole (test) dataset consisting of song lyrics from 12 musical genres. Reported are macro- and weighted averages of precision, recall, and F1 score across all genres on the test set, achieved by different text classification methods.

Figure 1 shows the achieved macro-average F1 score performances of all methods for all our datasets, i.e., for an iteratively increasing number of genres (and number of songs). We can state that the observa-

tions we just made also apply for all other datasets, i.e., Bernoulli NB is consistently the best performing classifier, and in general the machine learning methods are outperformed by the traditional classification methods. As expected, the classification performance of all methods degrades with increasing number of considered genres in the dataset. Only when adding the Rap genre to the smallest dataset, i.e., when increasing the number of considered genres from two to three, for most methods the classification performance improves.

For extended experimental results, refer to Section A of the Appendices. Here, similar observations can be made with Tables 3, 4, and 5, and Figures 2-6. Furthermore, Tables 6-11, and Figures 7-12 present detailed classification performances on each musical genre when training and evaluating the methods on the whole dataset. We can observe that in general all methods achieve highest F1 score performances for the genres Heavy Metal, Rap, and Gospel/Religioso. In contrast to that, Folk is a genre all methods struggle to predict correctly.

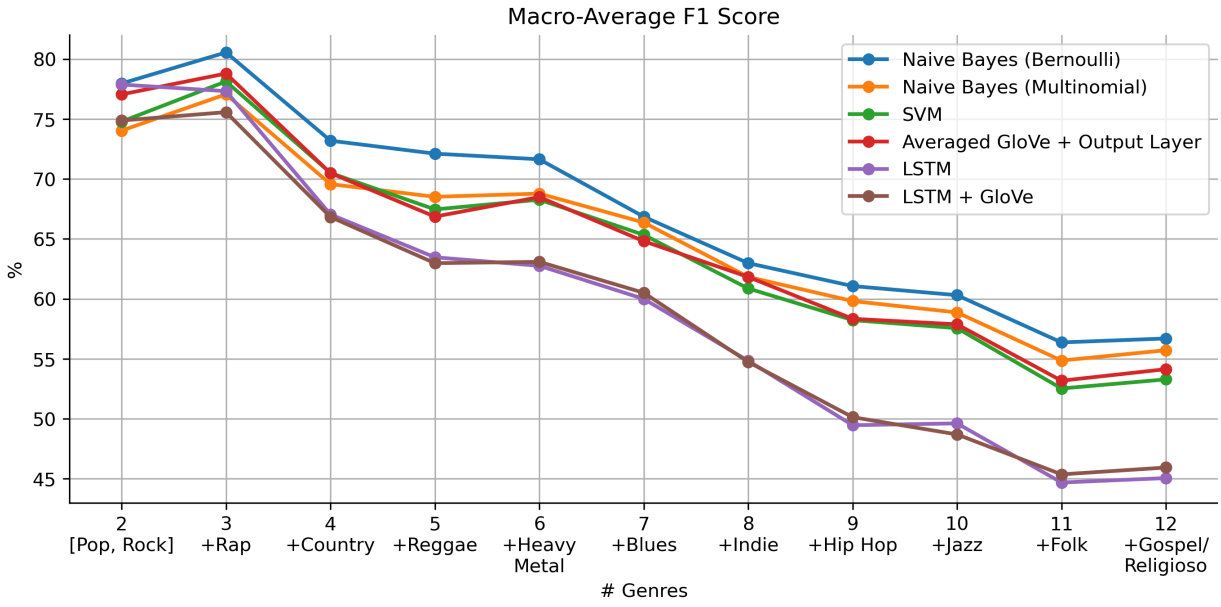

Figure 1: Achieved macro-average F1 score performances of different text classification methods for all our datasets (which differ in the number of considered musical genres).

6 Discussion

With the results presented in Section 5, we now discuss the methods' performances, in particular 1) how the traditional classification methods perform compared to the machine learning methods, 2) how GloVe embeddings affect the models' performances, 3) how the classification performance degrades for increasing number of included genres in the dataset, and 4) how the different genres related to each other.

6.1 Key Results

We made the key observation that, for our used datasets and methods, the machine learning methods are outperformed by the traditional classification methods. The only machine learning model that achieved similar results compared to the traditional methods was the Averaged GloVe + Output Layer. It is also interesting that Bernoulli NB outperforms Multinomial NB and SVM by a margin, which indicates that the frequency of words does not play a key role for song genre classification. Bernoulli NB achieves these performances while requiring less time for training as compared to the other traditional methods, and of course much less training time as compared to the machine learning methods. As we think that the utilized dataset offers enough training data ($\sim$22,500 in the train set when using the whole dataset), we can only guess why the machine learning methods, especially the LSTM models, perform so much worse than the traditional methods. Since all other methods perform better than the LSTM models, this indicates that for song genre classification the context of a word is not as important as expected.

We can state that using GloVe vectors did not really impact the performance of the LSTM model. As we can see in Figure 1, it depends on the number of considered genres which LSTM model performs better. This can also be explained by a degree of randomness in the training of machine learning models. At least the Averaged GloVe + Output Layer achieved similar results compared to the traditional methods, although it takes less time for training compared to the LSTM models. Potentially, this simple yet effective approach can outperform the traditional methods with some further adjustments.

As expected, with Figure 1 (and also Figures 2 to 6), we can observe well how the classification performance of all methods degrades when considering more and more genres in the dataset. This includes one exception, and that is when we also consider songs from the Rap genre, additionally to Pop and Rock songs. We can see, e.g., in Table 6 and Figure 7, that all methods are able to predict Rap songs quite reliably (Bernoulli NB: 77.86 macro-average F1 score). As opposed to that, we can observe that in general all methods struggle to predict Rock songs (Bernoulli NB: 39.19 macro-average F1 score).

With the confusion matrices that are depicted in Figures 7-12, we can determine genres that get mixed up with each other more often. For example, all methods find it difficult to distinguish between songs from the genres Rap and Hip Hop, even though Rap is a genre that is predicted reliably by all methods, and Hip Hop songs are still predicted better as compared to other genres. This can be explained with Hip Hop and Rap being similar musical genres[8] and Rap being a form of music that grew out of Hip-Hop culture[9], thus making it hard to distinguish between Rap and Hip Hop songs when only looking at song lyrics.

6.2 Threats to Validity

There are several concerns related to the dataset that we use to train and evaluate the song genre classification methods, and our approach in general.

We approached song genre classification as a multi-class, single-label text classification task, although songs do not necessarily belong to a single musical genre. We tried to work around this problem by, based on the Kaggle dataset that we utilized (see Section 4.1), considering only songs of artists that belong to one musical genre. Whether all artists in our used dataset can actually be assigned to only one genre remains at least debatable, if not unlikely. We must be similarly critical of the fact that in our used dataset(s) all of an artist's songs belong to the very same musical genre, which we also think is not true in all cases.

Lastly, the Kaggle dataset on which our dataset(s) are based, contains song lyrics of 79 genres and 4,168 artists. Since we only use song lyrics of 12 genres, and many artists are not considered since they belong to more than one genre, we end up having the song lyrics of 632 different artists in our whole dataset. On average, this means that our dataset includes the songs of $\sim$52 different artists per genre. We cannot rule out the possibility that the text classification methods used in this work learn and use lyrical characteristics of different artists rather than lyrical characteristics of the corresponding genre itself, which ideally is not the case.

6.3 Future Work

Our work enables several paths for future work. One issue of our work is that our implemented machine learning methods could not outperform the traditional methods, even though related work shows that this should be achievable (Tsaptsinos (2017); Kumar et al. (2018)). Therefore, it would be interesting to see how other deep learning models approaches perform on this dataset for different numbers of considered genres. Furthermore, in this work we did not make use of any sophisticated feature extraction, e.g., by utilizing characteristics such as rhyme feature, repetitive song structures, and slangs, which potentially can be used to improve classification performances (see Fell and Sporleder (2014)). Finally, we did not consider word lemmatization nor word stemming for lyrics preprocessing. It would be interesting to see how these techniques affect the models performances.

[8] *What is the Difference Between Rap and Hip Hop?* by Carol Francois (accessed: 13.05.2022): `https://www.musicalexpert.org/what-is-the-difference-between-rap-and-hip-hop.htm`.

[9] *Rap vs. Hip Hop* (accessed: 13.05.2022): `https://www.nextlevel-usa.org/blog/rap-vs-hip-hop`.

7 Conclusion

In this work, we targeted the task of song genre classification using song lyrics. For this, we presented six different classification models, three traditional methods and three machine learning methods. We trained and evaluated each method on a total of 11 datasets containing the lyrics of at least two, and up to 12 musical genres. Experiments have shown that the traditional methods, most importantly Bernoulli Naive Bayes, outperformed the machine learning methods, regardless of the number of considered musical genres. Utilizing GloVe word vectors for the bidirectional LSTM did not result in improved classification performance. Finally, we observed that the classification performance of all methods degrades with increasing number of considered genres in the dataset.

References

Snigdha Chillara, AS Kavitha, Shwetha A Neginhal, Shreya Haldia, and KS Vidyullatha. Music genre classification using machine learning algorithms: a comparison. *Int. Res. J. Eng. Technol.(IRJET)*, 6 (05):851–858, 2019.

Fabrice Colas and Pavel Brazdil. Comparison of svm and some older classification algorithms in text classification tasks. In Max Bramer, editor, *Artificial Intelligence in Theory and Practice*, pages 169–178, Boston, MA, 2006. Springer US. ISBN 978-0-387-34747-9.

Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 620–631, Dublin, Ireland, August 2014. Dublin City University and Association for Computational Linguistics. URL https://aclanthology.org/C14-1059.

Anshuman Goel, Mohd. Sheezan, Sarfaraz Masood, and Aadam Saleem. Genre classification of songs using neural network. In *2014 International Conference on Computer and Communication Technology (ICCCT)*, pages 285–289, 2014. doi: 10.1109/ICCCT.2014.7001506.

Kenneth Haase. Context for semantic metadata. In *Proceedings of the 12th Annual ACM International Conference on Multimedia*, MULTIMEDIA '04, page 204–211, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138938. doi: 10.1145/1027527.1027574. URL https://doi.org/10.1145/1027527.1027574.

Thorsten Joachims. Making large-scale support vector machine learning practical. *Advances in kernel methods: support vector learning*, page 169, 1999.

Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. Genre classification using word embeddings and deep learning. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2142–2146, 2018. doi: 10.1109/ICACCI.2018.8554816.

Qian Li, Hao Peng, Jianxin Li, Congying Xia, Renyu Yang, Lichao Sun, Philip S. Yu, and Lifang He. A survey on text classification: From shallow to deep learning. *CoRR*, abs/2008.00364, 2020. URL https://arxiv.org/abs/2008.00364.

Cory McKay, John Burgoyne, Jason Hockman, Jordan Smith, Gabriel Vigliensoni, and Ichiro Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. pages 213–218, 01 2010.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

Samira Pouyanfar, Saad Sadiq, Yilin Yan, Haiman Tian, Yudong Tao, Maria Presa Reyes, Mei-Ling Shyu, Shu-Ching Chen, and Sundaraja S Iyengar. A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys (CSUR)*, 51(5):1–36, 2018.

Winda Kurnia Sari, Dian Palupi Rini, and Reza Firsandaya Malik. Text classification using long short-term memory with glove. *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, 5(2): 85–100, 2019.

Gurinder Singh, Bhawna Kumar, Loveleen Gaur, and Akriti Tyagi. Comparison between multinomial and bernoulli naïve bayes for text classification. In *2019 International Conference on Automation, Computational and Technology Management (ICACTM)*, pages 593–596. IEEE, 2019.

Alexandros Tsaptsinos. Lyrics-based music genre classification using a hierarchical attention network. *CoRR*, abs/1707.04678, 2017. URL `http://arxiv.org/abs/1707.04678`.

Appendices

A Extended Results

This section contains extended experiment results. In particular, we provide the metric performances of all classification methods on the datasets that consist of song lyrics from 3, 6, and 9 musical genres (see Tables 3, 4, and 5). Furthermore, we provide plots that show the achieved macro-average precision and recall performances (see Figures 2 and 3), and achieved weighted average precision, recall, and F1 score performances (see Figures 4, 5, and 6) on all datasets. Finally, for all methods, Tables 6-11, and Figures 7-12 present detailed classification performances on each musical genre when training and evaluating the methods on the whole dataset (that contains the song lyrics of all 12 musical genres).

	Macro-Average			Weighted Average		
Method	Prec	Rec	F1	Prec	Acc/Rec	F1
Naive Bayes (Bernoulli)	**80.77**	**80.45**	**80.55**	**80.75**	**80.26**	**80.46**
Naive Bayes (Multinomial)	76.08	78.63	77.06	77.30	76.47	76.57
SVM	80.20	76.72	78.13	78.83	78.80	78.49
Averaged GloVe + Output Layer	78.75	78.97	78.79	78.96	79.31	79.08
LSTM	77.3	77.42	77.32	77.38	76.90	77.12
LSTM + GloVe	75.73	75.42	75.57	75.63	75.67	75.64

Table 3: Experimental results for lyrics-based song genre classification on the dataset that consists of song lyrics from 3 musical genres (Pop, Rock, Rap).

	Macro-Average			Weighted Average		
Method	Prec	Rec	F1	Prec	Acc/Rec	F1
Naive Bayes (Bernoulli)	**73.34**	**70.92**	**71.64**	**71.53**	**70.57**	**70.79**
Naive Bayes (Multinomial)	69.89	69.19	68.78	69.08	68.40	68.23
SVM	73.05	65.83	68.27	69.71	68.44	68.56
Averaged GloVe + Output Layer	69.16	68.26	68.49	68.09	68.44	68.09
LSTM	63.21	62.66	62.76	63.15	63.25	63.06
LSTM + GloVe	63.42	62.85	63.09	63.38	63.19	63.25

Table 4: Experimental results for lyrics-based song genre classification on the dataset that consists of song lyrics from 6 musical genres (Pop, Rock, Rap, Country, Reggae, Heavy Metal).

	Macro-Average			Weighted Average		
Method	Prec	Rec	F1	Prec	Acc/Rec	F1
Naive Bayes (Bernoulli)	**63.38**	**61.67**	**61.07**	**62.98**	**61.12**	**61.21**
Naive Bayes (Multinomial)	60.98	60.41	59.82	60.69	60.09	59.79
SVM	65.63	55.08	58.23	60.96	59.09	59.11
Averaged GloVe + Output Layer	57.95	59.48	58.34	59.12	59.41	58.95
LSTM	50.04	49.06	49.47	51.21	50.86	50.98
LSTM + GloVe	50.13	50.88	50.14	51.85	52.29	51.84

Table 5: Experimental results for lyrics-based song genre classification on the dataset that consists of song lyrics from 9 musical genres (Pop, Rock, Rap, Country, Reggae, Heavy Metal, Blues, Indie, Hip Hop).

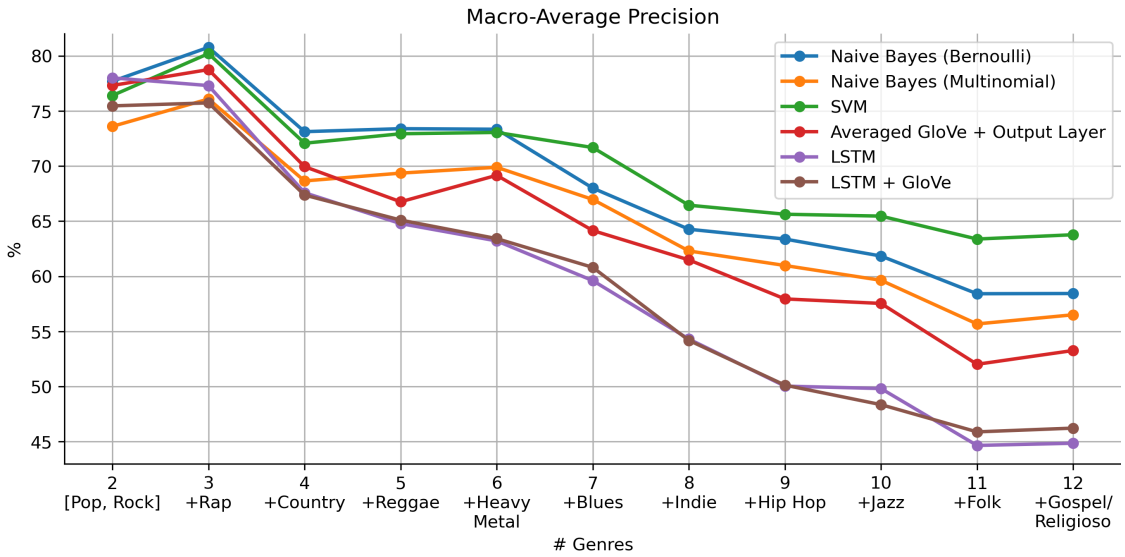

Figure 2: Macro-average Precision performances of text classification methods for all our datasets.

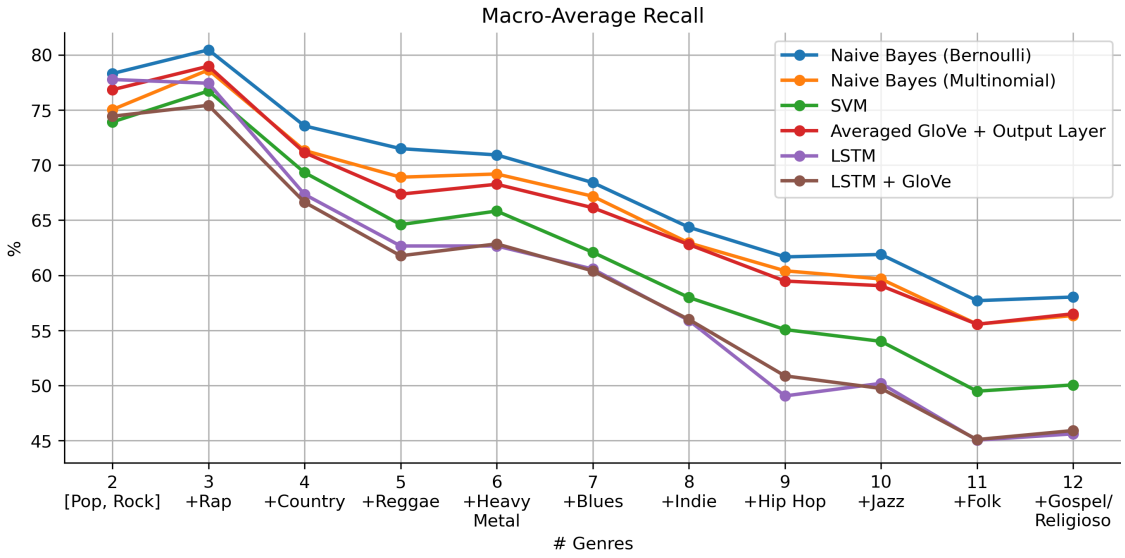

Figure 3: Macro-average Recall performances of text classification methods for all our datasets.

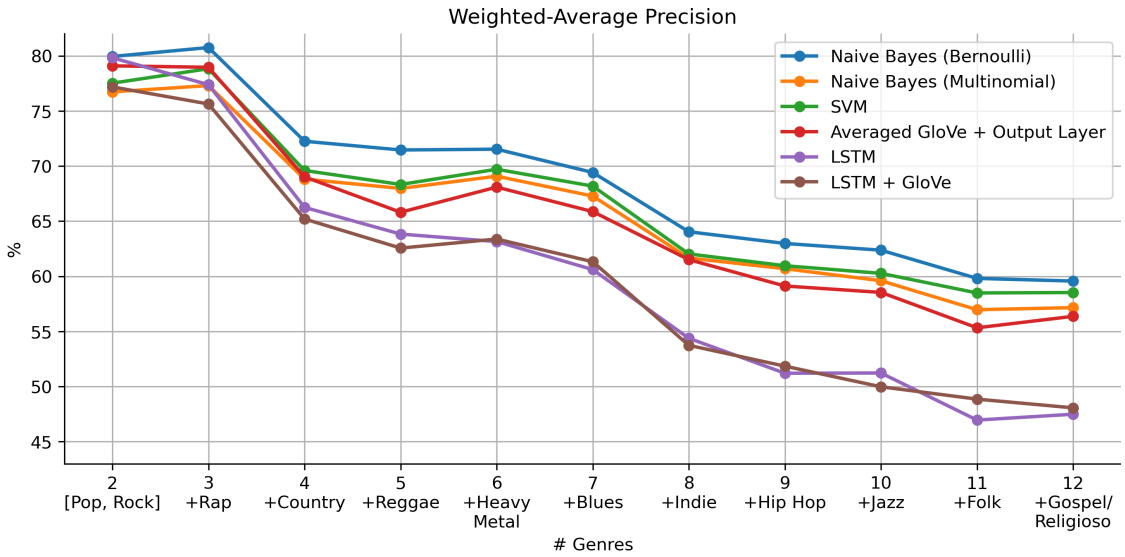

Figure 4: Weighted-average Precision performances of text classification methods for all our datasets.

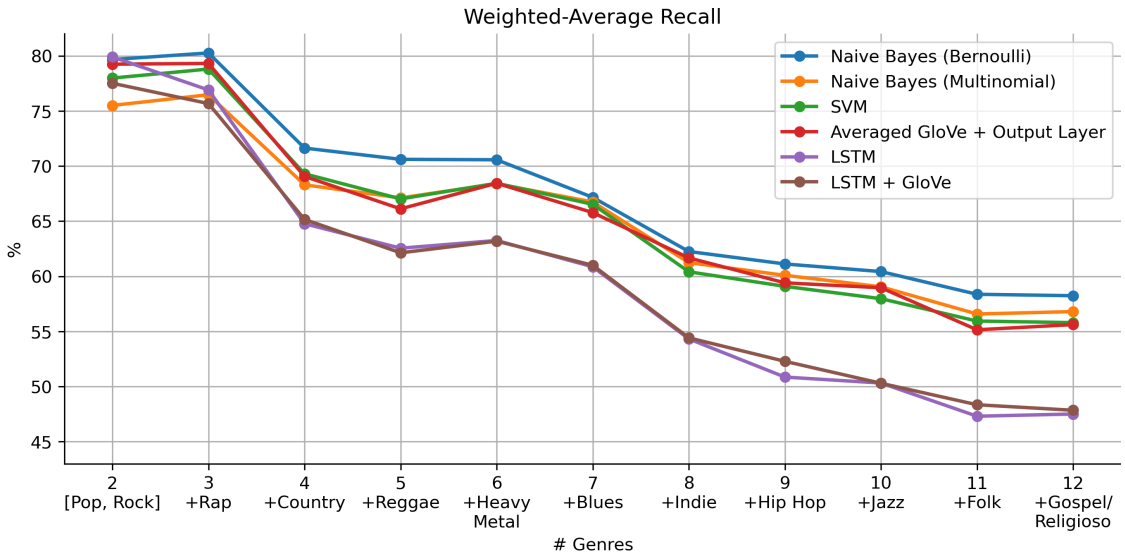

Figure 5: Weighted-average Recall performances of text classification methods for all our datasets.

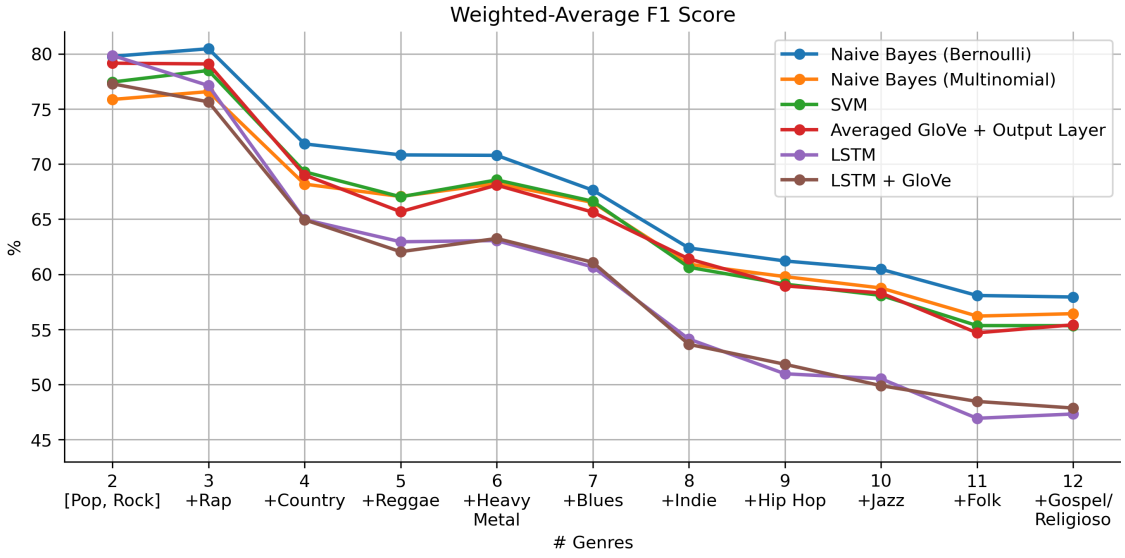

Figure 6: Weighted-average F1 score performances of text classification methods for all our datasets.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	81.28	75.53	78.30
Rap	602	74.06	82.06	77.86
Reggae	369	85.00	55.28	67.00
Gospel/Religioso	427	60.62	68.85	64.47
Country	1,429	64.90	57.31	60.87
Hip Hop	332	68.89	46.69	55.66
Indie	1,274	53.26	57.77	55.42
Jazz	405	43.76	71.85	54.39
Pop	753	43.67	62.82	51.53
Blues	317	40.28	64.67	49.64
Rock	1,386	45.89	34.20	39.19
Folk	320	39.74	19.38	26.05
macro-average	8,930	58.44	58.03	56.70
weighted average	8,930	59.57	58.24	57.94

Table 6: Classification performance per genre on the whole dataset (test set) when using ***Bernoulli Naive Bayes***. Genres are sorted according to the F1 score achieved for them.

Figure 7: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***Bernoulli Naive Bayes***.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	78.13	79.26	78.69
Rap	602	65.36	85.88	74.23
Gospel/Religioso	427	65.45	67.45	66.44
Reggae	369	83.20	55.01	66.23
Country	1,429	60.04	58.57	59.30
Jazz	405	56.90	59.01	57.94
Blues	317	48.74	61.20	54.27
Hip Hop	332	60.00	46.99	52.70
Indie	1,274	55.16	48.67	51.71
Pop	753	39.43	59.23	47.35
Rock	1,386	41.65	33.12	36.90
Folk	320	24.05	21.88	22.91
macro-average	8,930	56.51	56.35	55.72
weighted average	8,930	57.16	56.80	56.43

Table 7: Classification performance per genre on the whole dataset (test set) when using ***Multinomial Naive Bayes***. Genres are sorted according to the F1 score achieved for them.

Figure 8: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***Multinomial Naive Bayes***.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	77.38	79.03	78.20
Rap	602	78.15	77.24	77.69
Gospel/Religioso	427	73.12	59.25	65.46
Jazz	405	67.75	51.36	58.43
Reggae	369	76.32	47.15	58.29
Country	1,429	54.50	62.28	58.13
Hip Hop	332	74.48	43.07	54.58
Blues	317	64.13	45.11	52.96
Indie	1,274	49.24	50.78	50.00
Pop	753	39.95	46.75	43.08
Rock	1,386	34.77	44.16	38.91
Folk	320	34.58	11.56	17.33
macro-average	8,930	60.36	51.48	54.42
weighted average	8,930	57.19	55.59	55.59

Table 8: Classification performance per genre on the whole dataset (test set) when using ***SVM***. Genres are sorted according to the F1 score achieved for them.

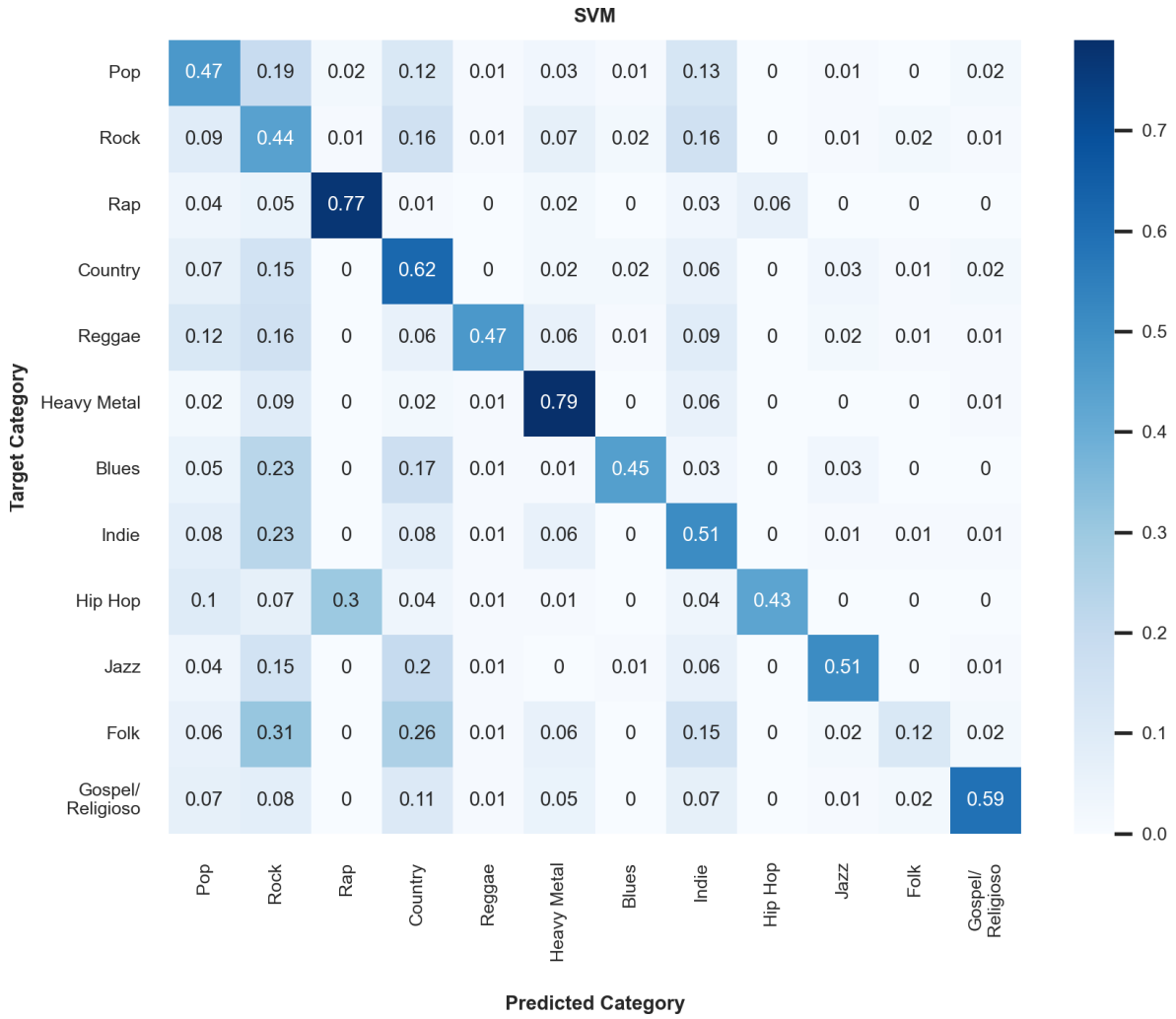

Figure 9: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***SVM***.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	77.04	79.56	78.28
Rap	602	72.20	77.24	74.64
Gospel/Religioso	427	59.31	68.62	63.63
Reggae	369	66.77	56.64	61.29
Country	1,429	63.60	51.22	56.74
Jazz	405	45.60	70.37	55.34
Indie	1,274	52.38	54.47	53.41
Blues	317	40.52	64.04	49.63
Hip Hop	332	50.79	48.49	49.61
Pop	753	48.42	48.74	48.58
Rock	1,386	43.48	30.30	35.71
Folk	320	19.08	28.44	22.84
macro-average	8,930	53.27	56.51	54.14
weighted average	8,930	56.37	55.62	55.42

Table 9: Classification performance per genre on the whole dataset (test set) when using ***Averaged GloVe + Output Layer***. Genres are sorted according to the F1 score achieved for them.

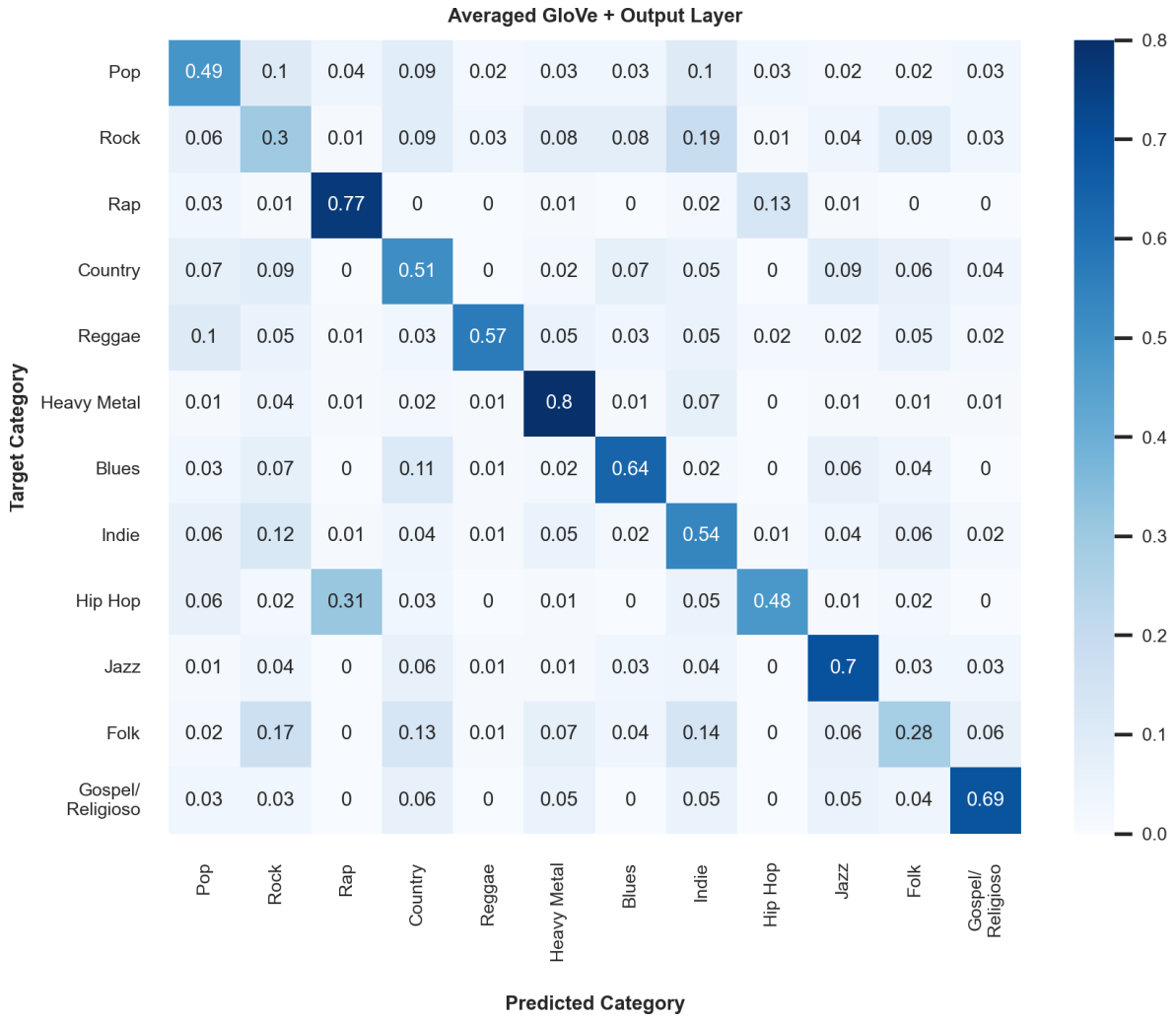

Figure 10: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***Averaged GloVe + Output Layer***.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	71.75	75.08	73.38
Rap	602	59.42	68.11	63.47
Gospel/Religioso	427	60.31	55.50	57.80
Reggae	369	56.37	47.97	51.83
Country	1,429	54.02	47.94	50.80
Jazz	405	46.41	52.59	49.31
Indie	1,274	39.56	45.53	42.34
Blues	317	36.36	44.16	39.89
Pop	753	39.86	37.58	38.69
Hip Hop	332	30.54	34.04	32.19
Rock	1,386	33.13	27.42	30.00
Folk	320	10.54	11.56	11.03
macro-average	8,930	44.86	45.62	45.06
weighted average	8,930	47.49	47.51	47.33

Table 10: Classification performance per genre on the whole dataset (test set) when using ***LSTM***. Genres are sorted according to the F1 score achieved for them.

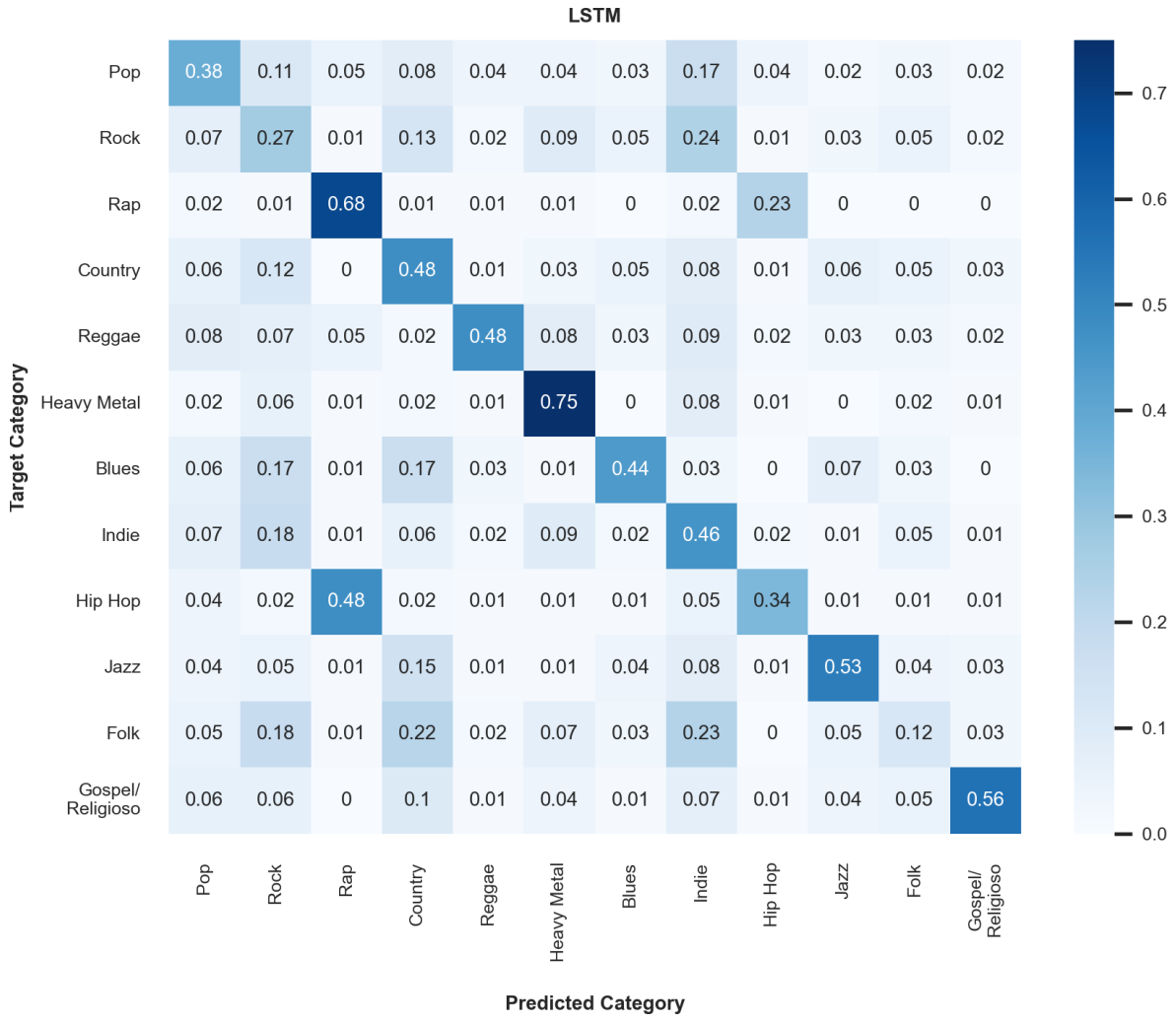

Figure 11: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***LSTM***.

Name	# Songs	Prec	Rec	F1
Heavy Metal	1,316	72.69	69.98	71.31
Gospel/Religioso	427	62.94	58.08	60.41
Rap	602	58.68	61.79	60.19
Reggae	369	60.94	49.05	54.35
Country	1,429	49.49	54.16	51.72
Jazz	405	41.97	51.60	46.29
Blues	317	45.45	45.74	45.60
Indie	1,274	41.86	39.56	40.68
Pop	753	40.60	37.58	39.03
Hip Hop	332	33.15	37.05	34.99
Rock	1,386	34.60	34.20	34.40
Folk	320	12.38	12.19	12.28
macro-average	8,930	46.23	45.92	45.94
weighted average	8,930	48.07	47.85	47.87

Table 11: Classification performance per genre on the whole dataset (test set) when using ***LSTM + GloVe***. Genres are sorted according to the F1 score achieved for them.

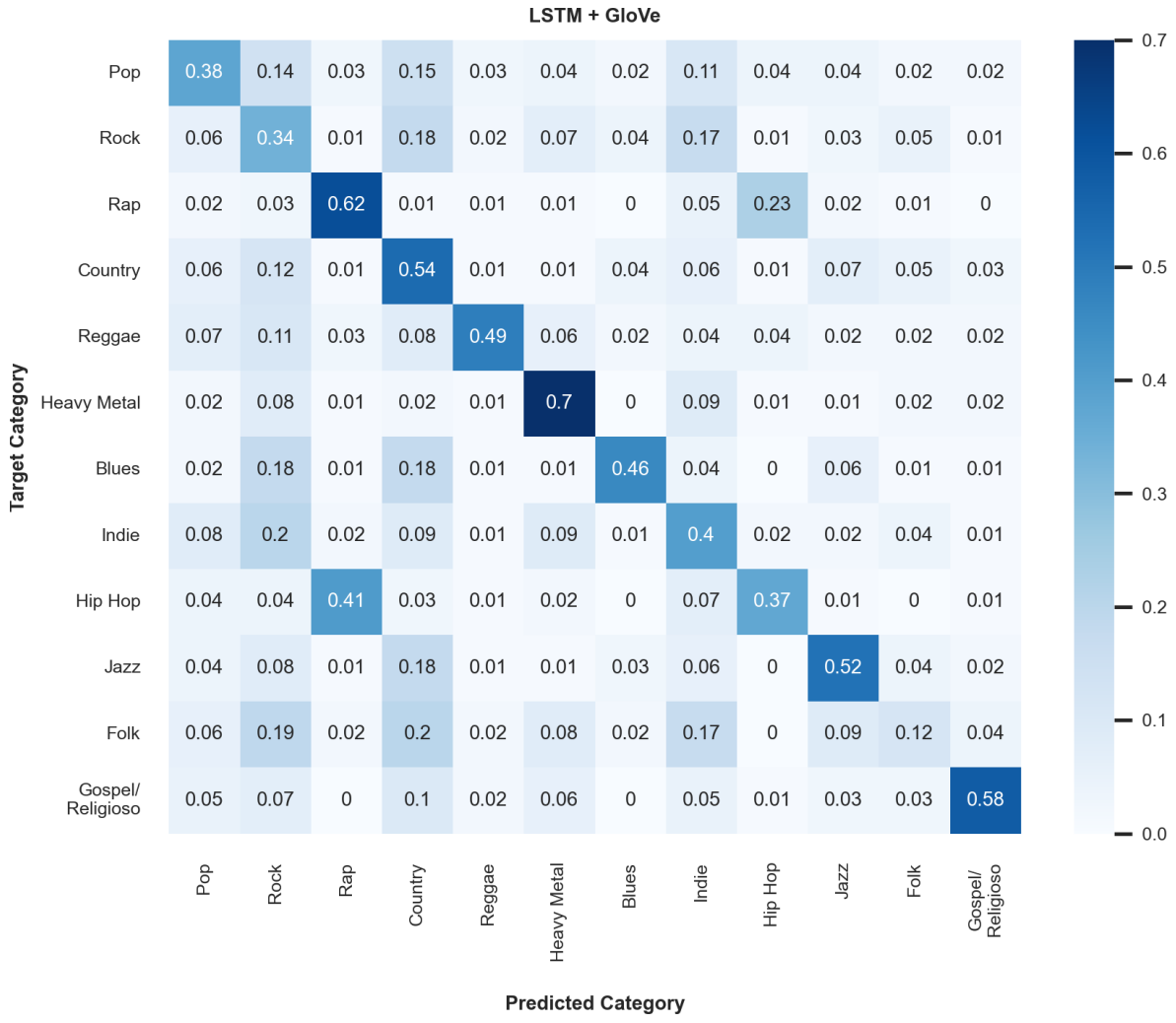

Figure 12: Confusion matrix for lyrics-based song genre classification on the whole dataset (test set) when using ***LSTM + GloVe***.