Music Genre Classification
CS7IS2 Project (2020-2021)

Priyanka Arumugam, Salil Vinit Kulkarni, Sneha Konoth, Vivek Kumar

arumugap@tcd.ie, sakulkar@tcd.ie, konoths@tcd.ie, kumarv2@tcd.ie

Abstract. The project explores the AI application in the sector of music genre detection. We surveyed 6 approaches for the music genre detection, and selected 3 approaches: LSTM, XGBoost and Genetic Algorithm based SVM. The working of these algorithms are briefly discussed in this report. We attempt to perform a comparative analysis on these algorithms based on the parameters like the execution time and accuracy. To model the algorithms, we use the GTZAN dataset. A brief idea on preprocessing steps involved in the dataset is discussed here. In conclusion, we found that the Ensemble Learning: XGBoost performed the best.

Keywords: Music Genre Detection, Comparative Analysis, GTZAN, LSTM, XGBoost, Genetic Algorithm, GA-SVM.

1 Introduction

Artificial Intelligence has been in great attraction all over the world from the past couple of years due to its exciting opportunities in a plethora of industries. There are a lot of interesting applications of AI in many fields which are being explored in many companies like Alphabet Inc, Amazon, Salesforce, etc. One of the industries which we choose to work upon and compare the Artificial Intelligence approaches is the Music Industry.
In this report we aim to perform a comparative analysis for three AI approaches viz. LSTM, XGBoost and Genetic Algorithm based SVM for Classification of Music Genre on the GTZAN Dataset.We would be performing this comparative analysis on the basis of parameters such as Accuracy, Precision and Execution time.

Link for the presentation:
https://tcdud-my.sharepoint.com/:p:/g/personal/konoths_tcd_ie/EZBFQReoidVGjjwe0qmWLMMBgfYI1zSkwKo_vfnxvPHi9g

2 Related Work

We reviewed a couple of approaches for music genre classification. The following is a brief description of the approaches we surveyed.

[1]Authors Chun Pui Tang et.al in "*Music Genre classification using a hierarchical Long Short Term Memory (LSTM) model*" describes two strategies which they compared on the GTZAN Dataset. In the first approach, they took only 6 genres into consideration and in the second strategy they espouse a hierarchical divide and conquer approach for all the 10 genres. While comparing the results of these two approaches, they found that the Strategy two i.e hierarchical divide and conquer worked better and gave a higher accuracy than the state-of-the-art i.e the strategy one. They concluded that LSTM has a potential to be used for music genre classification based on their results.

[2]In "*Machine Learning and chord based feature engineering for genre detection in popular Brazilian music*", authors Bruna D. Wundervald and Walmes M. Zeviani, emphasises on the harmonics and the mechanics of different music genres and how it could help in the task of detection of the music genre. Broadly, the authors try to extract the features from the harmonics of the music data which they have collected. For this purpose, they use random forest as a feature selection. After applying this technique on the data, they found out that it gave an accuracy of 62% for the features selected by random forest.

[3]On exploring the application of Machine learning and Artificial intelligence in the music genre detection and its growing interest in the digital music processing, Authors Ahmet Elbir, Hilmi Bilal Çam et.al in "*Music Genre Classification and Recommendation by Using Machine Learning Techniques*", focuses on acoustics features which they extract by using digital signal processing and using that with the help of Machine learning techniques, classifies and detects music genre. They compared two approaches viz. CNN and SVM. On this comparative analysis they found that SVM performs better.

[4]In "*Feature Selection for Musical Genre Classification Using a Genetic Algorithm"*, the authors Abba Suganda Girsang, Andi Setiadi Manalu and Ko-Wei Huang, proposes a model for music genre classification where the feature extraction is done by evolutionary algorithm such as Genetic Algorithm (GA) and classification is performed by machine learning algorithms such as Naive Bayes Classifier (NBC), K-Nearest Neighbor (kNN) or Support Vector Machines (SVM). For short-term and mid-term feature extraction, the approach uses low level time-domain and frequency-domain features. It was observed that the use of GA for feature extraction improved the F-measure score by 15%. The study showed that the SVM-GA has a score of 80.1% when compared to 72% of KNN-GA and 67.3% of NBC-GA. However, feature selection using GA increases computational time.

[5]In "*Classification of Music Genres with eXtreme Gradient Boosting*" by Jesper Muren, the author aims to build and evaluate a classifier based on extreme Gradient Boosting(XGBoost) for music genre classification and to understand the performance of the classifier with different features and on different genres. It was noted that XGBoost gave an accuracy score of 73.43%, under a lacking dataset. Therefore, with better dataset and fine tuning of features, the model accuracy may increase. The work also shows that there exists some correlation between some of the features and genres.

3 Problem Definition and Algorithm

Problem Definition:

In this report, we explore the AI application in the field of digital music processing and music genre classification by comparing 3 approaches: LSTM, Genetic Algorithm with SVM and XGBoost. By conducting this analysis, we aim to find the most appropriate technique which could be used for music genre detection. We strive to compare these approaches on some fixed parameters.

Algorithm:

Neural Network Approach - LSTM (Long Short Term Memory):

RNN Recursive Neural Network suffers from short term memory. Thus, when it has to process a large sequence of data it can get difficult for it to process and this could add up in the learning process of the mode. Additionally, it also suffers from gradient loss problems during back propagation, which means the gradient reduces while updating each step. This could also affect the accuracy as well as the learning process of the model as, if the gradient becomes significantly low , it won't help the layers to learn thus , would have short term memory.

To tackle all these problems of the RNN, LSTM and similar algorithms were implemented. The LSTM internally has a mechanism called gate. These gates help in the flow of the data as well as they can learn which of the data is important and which of the data from the sequence which is passed can be discarded.

The flow of the data in the LSTM is almost the same as in RNN. The LSTM does the processing of the data as it moves forward. The main difference between the LSTM and the RNN is the sort of internal operations that happen. The main operation of the LSTM includes the:

Sigmoid, Tanh, Pointwise multiplication, Pointwise addition ,Concatenation

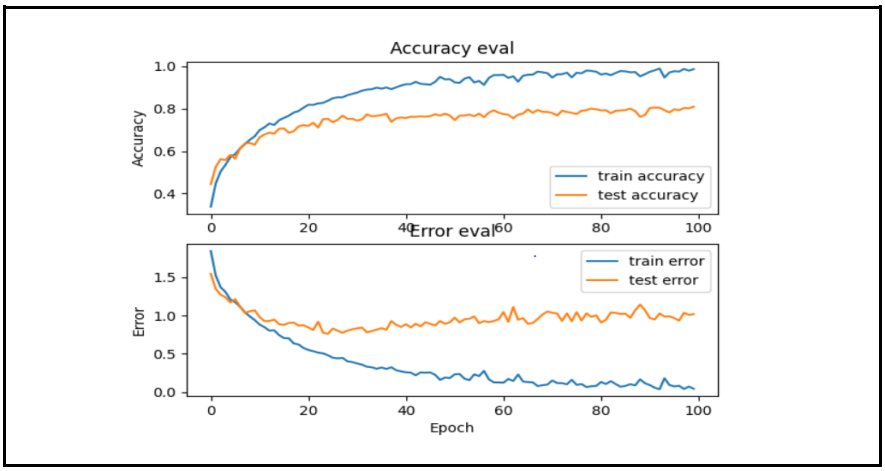
For LSTM-RNN model, we are extracting only the Mel Frequency Cepstral Coefficients from the raw data because it defines the brightness of the sound.We are using Librosa library to convert the audio tracks into MFCC features and later saving them in a JSON file.

Experimental Results

Initially the feature data is split into training set and test set with a proportion of 0.75 and 0.25 respectively. Next, the training set is further divided into training set and validation set with validation set having 20 % of the training set.

A sequential model with a linear stack of layers is used.The first layer is the LSTM layer with 64 units and it returns sequences to ensure that the next layer receives layer of sequences, not just randomly scattered data.Then we have a dense layer with 'relu' activation followed by a dropout layer to avoid overfitting. Finally we have a fully connected dense layer with 'softmax' activation and final neurons equal to the ten different genres.

The model fit over 100 epochs with batch size of 32.The accuracy of the predicted test dataset from the model is 80%. We could actually get a better accuracy for more epochs.



Ensemble Approach (XGBoost) :

eXtreme Gradient Boosting or XGBoost is an ensemble machine learning algorithm based on decision trees. XGBoost is an optimized version of Gradient Boosting algorithm that shows improved scalability, accuracy, computational speed and overall model performance.

Ensemble Learning uses different machine learning models for aggregated decisions to improve the prediction performance. XGBoost employs the Boosting algorithm to perform ensemble learning. The Boosting algorithm combines the different models sequentially so as to minimize the errors and improve the overall performance. In Boosting, each predictor learns from the mistakes of the previous predictor by assigning different weights to the correct and incorrect predictions.

The performance of XGBoost is majorly attributed to two factors, Algorithm enhancements and System optimization. Some of the XGBoost features are:

System Optimization

- Parallel Learning - XGBoost uses all available CPU cores to parallelize the process of tree generation.
- Tree Pruning - Since boosting algorithms are greedy, to avoid overfitting or bias, the stopping criterion used is maximum depth of tree. Computational performance is increased by the depth-first approach.
- Hardware Optimization - The cache awareness by using internal buffers and the out-of-core computing to optimize the available disk space helps in utilizing hardware resources efficiently.

Algorithm Enhancement

- Regularized Learning - To prevent overfitting, both Lasso (L1) and Ridge (L2) regularization is employed.
- Sparsity Awareness - XGBoost allows for missing values by handling sparsity patterns in the data.

- Cross-validation - The number of iterations in a single run is determined by built-in cross-validation method.

The dataset is initially subjected to some data preprocessing such as removing unnecessary columns. In order to build an efficient XGBoost classifier, we need to select the best features. The importance of a feature is determined based on the Permutation Importance. The top 30 features are used to train a XGBoost classifier model.
For hyperparameter tuning, we choose an initial learning rate and determine the optimum number of trees. Based on these values, we tune the parameters. The accuracy obtained for XGBoost is 88.4%.

Genetic Algorithm Approach (GA + SVM):
Extracting the best features is one of the crucial steps in any kind of classification problem when data contains a large number of dependent and independent variables. Feature engineering is VERY important in audio analysis where statistical analysis and parameters like expected value, standard deviation and distances are derived from the features. These data are then normalized to avoid the anomalies and minimize redundancy. There might be some features that would affect the output and prediction of trained models by reducing the accuracy for classification problems. In multivariable data, there would be chances of having strong correlation between various feature vectors. In order to remove such scenarios, we do the optimization or feature reduction where we find the best features for the machine learning model to apply.
Genetic Algorithm (GA) – An optimization technique
GA is one of the advanced algorithms in computer science,not only for AI algorithms. This is more motivated by the human genetic process of passing genes from one generation to another.
Implementation:
1.Initial Population:
Randomly selected the population based on the data. The initial population is generated and encoded in binary {0,1}. As the output of this implementation is to find out if the feature element has been selected or not in the reduced set of features. Hence, the solution is represented in binary form feature labelled 1 are used and label 0 are not.
2.Fitness Function
After being initialized, the parents are selected. That is the best solutions in the current population are selected for mating in order to produce better solutions. To find the best chromosome, we need to find the fitness value associated with selected parents. The fitness function will return a classification accuracy for each solution. We are using the SVC model to get the accuracy.
3.Selection
Once we get the fitness score, the best fitted chromosome would be selected as parents to pass the genes (feature vectors) for the next generation and create a new population.
4.Cross-Over
A new set of chromosomes will be created by combining the parents and cross over it with a new population set.
5.Mutation:

This process alters one or more feature vector value in a chromosome of the new population which helps in generating diversity. The new generated set of population will be used in next generation.
6.Repeat the process from fitness function to mutation again for each generation.

Genetic algorithm results provide best 30 features out of 57 features available in audio file.
Machine Learning Classifier – SVM
This new set of features will be used for training and fitting the model. As we are using SVM classifier as a model to classify the music genre in this project. As we have fit the classifier with best features, it improves the classification accuracy significantly with 84 %.

4 Experimental Results

- Methodology:

We are using GTZAN dataset for the training and testing purpose. Following is a brief description about GTZAN.

GTZAN:

The dataset which we are using for the music genre classification is GTZAN. GTZAN has 1000 records in total. There are 10 genres in this dataset which are : Each of these Blues, Classical, Country, Disco, Hip Hop, Jazz, Metal, Popular, Reggae, and Rock. genres have 100 audio clips of 30 seconds long. Along with the audio it even has a csv file with 60 features.
We would be working on these features and applying the three approaches on it to see which performs the best and the reason behind it.

Evaluation Criteria:

We'll be evaluating the methods on the basis of the accuracy, precision and execution time for the GTZAN Data set.
While Comparing the approaches with the baseline model which was logistic regression we got a result of 74% which is worse as compared to our three approaches.

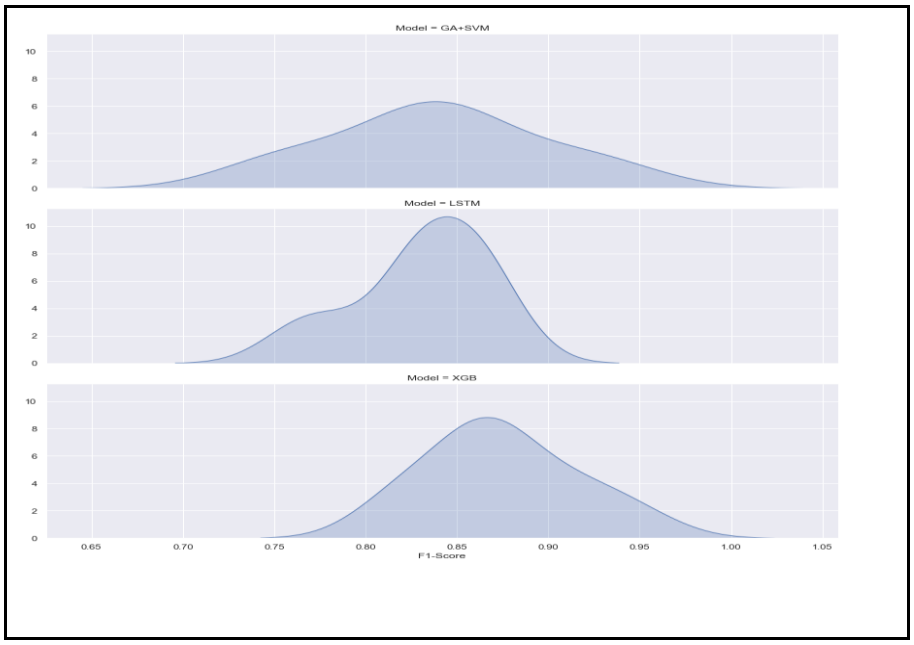
- Results:

In comparison, we arrived at a conclusion that the XGBoost, which was the ensemble technique, was best.

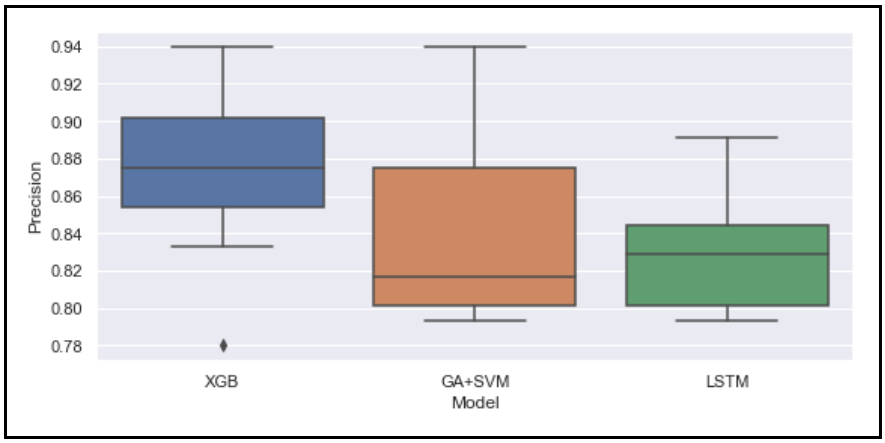
Accuracy:

While looking at the accuracy, we found out that the XGBoost had the highest accuracy of 88% while the lowest was of LSTM which was 80% and GA + SVM had an accuracy of 84%.

The distribution plot below gives us the F1-score, which is the balance between the precision and the recall. The mean of the F1-score distribution gives the accuracy of the model. From the F1-score plot, it is clear that the XGBoost has the highest mean and hence the highest accuracy compared to LSTM and GA+ SVM approaches. The Confidence interval for XGBoost is narrow which shows that there is less error when compared to LSTM and GA+SVM.



Precision:
Precision is a metric which measures the true positive values to all the values which were retrieved. Taking this into consideration, XGBoost still outperformed the other 2 methods.
This can be seen in the box plot below.



The mean precision for XGBoost is around 87% whereas the LSTM and GA + SVM are around 81% and 84% respectively.

Below is the table for comparative analysis.

Sl. No	Approach	Execution Time	Accuracy
1	LSTM	4 minutes	80%
2	GA + SVM	12 minutes	84%
3	XGBoost	<1 minute	88.4%

We also tested these approaches on the basis of the execution time XGBoost took less than 1 minute while LSTM took 4 mins and GA + SVM took around 12 mins. They were compared on google colab with a GPU it provides.

Following tables shows the overall performance of the approaches.

Parameters	**F1-Score**			**Precision**		
Genre	XGB	LSTM	GA+SVM	XGB	LSTM	GA+SVM
blues	0.861	0.837	0.847	0.848	0.807	0.805

metal	0.947	0.863	0.913	0.94	0.841	0.895
reggae	0.819	0.763	0.753	0.78	0.793	0.793
jazz	0.859	0.83	0.83	0.874	0.8	0.8
pop	0.864	0.824	0.824	0.871	0.864	0.845
disco	0.902	0.854	0.854	0.896	0.834	0.82
classical	0.921	0.871	0.931	0.928	0.891	0.94
rock	0.881	0.859	0.859	0.904	0.845	0.885
hiphop	0.867	0.824	0.814	0.876	0.824	0.814
country	0.821	0.779	0.759	0.833	0.798	0.796
Average	88.4 %	82.9 %	84 %	87.5 %	82 %	85.9 %

Overall we found out that XGBoost performs the best amongst all the approaches that we compared.

- Discussion: From the results we understand that the ensemble learning works better with the Music Genre Classification, which was surprising! because initially we anticipated that the LSTM which is a modification of RNN would work better as music follows a specific sequence but it performed the worst in comparison of the other three approaches.

In comparison with the

We also tried to bring in eccentricity in our model by not going for pre-trained Neural network model, in contrast we tried to implement the architecture by ourselves. Along with this we also customized the SVM model by selecting optimized feature by adding Genetic Algorithm to it

5 Conclusions

After surveying the three models for genre classification, based on the accuracy and Precision, we came to a conclusion that XGBoost outperforms the other two techniques. We also got a brief idea about the applications of Artificial Intelligence in the domain of Music Genre Classification.

We also studied that, although GTZAN Dataset is widely used for the music genre classification it suffers from some biases like most of data is mislabeled as well as the audio files used are very similar. Such factors can affect the model's behavior.

References

1. A. Girsang, A. Manalu and K. Huang, "Feature Selection for Musical Genre Classification Using a Genetic Algorithm", *Advances in Science, Technology and Engineering Systems Journal*, vol. 4, no. 2, pp. 162-169, 2019. Available: 10.25046/aj040221.
2. T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *arXiv.org*, 2016. [Online]. Available: https://arxiv.org/abs/1603.02754v1. [Accessed: 30- Apr- 2021].
3. C. Tang, K. Chui, Y. Yu, Z. Zeng and K. Wong, "Music genre classification using a hierarchical long short term memory (LSTM) model", Semanticscholar.org, 2018. [Online]. Available: https://www.semanticscholar.org/paper/Music-genre-classification-using-a-hierarchical-Tang-Chui/5ffbd42b649a720883acf609f22caaa080f3890a.
4. B. Wundervald and W. Zeviani, "Machine learning and chord based feature engineering for genre prediction in popular Brazilian music", *arXiv.org*, 2019. [Online]. Available: https://arxiv.org/abs/1902.03283.
5. O. Diab, A. Mainero and R. Watson. "Musical Genre Tag Classification With Curated and Crowdsourced Datasets." 2012. [Online]. Available: http://cs229.stanford.edu/proj2012/DiabMaineroWatson-MusicalGenreTagClassificationWithCuratedAndCrowdsourcedDatasets.pdf
6. A. Elbir, H. Bilal Çam, M. Emre Iyican, B. Öztürk and N. Aydin, "Music Genre Classification and Recommendation by Using Machine Learning Techniques," 2018 Innovations in Intelligent Systems and Applications Conference (ASYU), 2018, pp. 1-5, doi: 10.1109/ASYU.2018.8554016.
7. J. Muren, "Classification of Music Genres with eXtreme Gradient Boosting", 2019.
8. "Illustrated Guide to LSTM's and GRU's: A step by step explanation", *Medium*. [Online]. Available: https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21.
9. "XGBoost Algorithm: Long May She Reign!", *Medium*. [Online]. Available: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d.
10. "XGBoost: A Deep Dive into Boosting", *Medium*. [Online]. Available: https://medium.com/sfu-cspmp/xgboost-a-deep-dive-into-boosting-f06c9c41349.