Music Genre Classification and Mood Tagging

Darshan Navadiya, Sanket B Shah, Harin Khakhi, Nilay Gandhi

***Abstract*—A music genre is a classification system that places different instructional materials into neat categories majorly based on the rhythmic structure, harmonic content and instrumentation. In this paper, an automated system is created to classify a piece of music into some predefined genres using GTZAN dataset. Various features are extracted from the musical piece and are tested for their significance in the task of genre classification using Librosa library. Different classifiers, namely Logistic Regression, K Nearest Neighbors, Support Vector Classifiers are used. Results are presented to compare the accuracies of the different classifiers for the given GTZAN dataset.**

***Index Terms*—Musical Genre Classification, Feature Extraction, GTZAN dataset, Support Vector Machines, Dimensionality Reduction, Principal Component Analysis, Logistic Regression, K Nearest Neighbors.**

I. Introduction

Musical genre is a category in which different musical pieces with same characteristics are placed into. This provides a neat categorical way of managing the vast number of musical pieces that the humans have created over its entire history. This task was performed by hand by experts which resulted in inconsistencies due to similarity and background of the musical pieces. To overcome this, automated systems were created to avoid inconsistencies. Various features can be extracted from the audio file and similarities can be checked to classify them into genres. In this paper, this problem of automated classification of musical pieces into their respective genres is addressed. GTZAN dataset is used for classifying the input audio file into 10 genres: Blues, Classical, Country, Disco, Hiphop, Jazz, Metal, Pop, Reggae, Rock. To achieve this, the information inside the audio piece is extracted and different classifiers are applied onto the extracted features.

II. Literature Survey

A work [1] by Tshepo Nkambule from University of the Witwatersrand described a way to classify genres using the GTZAN dataset. An article [2] described a similar approach for the same.

III. Implementation

A. Dataset

The dataset used for training/validation/testing is the GTZAN Dataset [3]. This dataset contains 30 second and 3 second audios clips for 10 different genres. Since the problem boils down to classification of genres, this dataset seemed to be the ideal choice for the same.

B. Preprocessing

The dataset was first scaled by using the StandardScaler() function. The string values like names were encocded into integer values for computation. Also, the dataset consisted of audio samples that cannot be fed directly into a learning architecture. Hence relevant numerical values were needed to be extracted for the same. The python library $Librosa$ [4] was used for this purpose. This website article [5] gave good insights about the library and helped in understanding the process of audio feature extraction

1) Feature Cleaning: $Librosa$ helped in extracting a great deal of features, however, some features couldn't be extracted from a real audio sample. Hence, if we kept those features, it would get impossible to detect genre of a real song. That's why we dropped certain features like $perceptr\_mean, perceptr\_var$. Some features proved to be irrelevant in detecting genre of a song, so we dropped features like $length, filename$. The final dataset that was considered had total 55 features. Features included, were technical properties of an audio like $ZeroCrossingRate, SpectralCentroid, SpectralRolloff, Mel-FrequencyCepstralCoefficients(MFCCs)$, etc.

Further, we decided to experiment with the dataset and see whether the dataset can be further compressed into lesser feature without compromising the Model Performance. We applied Principal Component Analysis on the Dataset using $sklearn$ [6]. To retain 95% of the variance, it was observed that 28 features were sufficient. However, the reduced dataset proved to affect the model performance, that's why as of now, we have decided to not go with the current result of PCA, however that decision might be revised in future.

Finally, the dataset was split into Train and Test Datasets. The split was such that 75% of the data was used as Training Data and 25% was used as Test Data.

C. Model(s) Used

We considered various models so as to compare performance between them with respect to this specific problem and dataset. The library $sklearn$ was used for the implementation

1) Logistic Regression
2) K Nearest Neighbours
3) Support Vector Machine

D. Training

1) Logistic Regression: This algorithm was used, and alongside the same regularisation was also applied. We checked for $L1$ and $L2$ regularisation, after realising that $L1$ yields better performance, we went ahead with it

2) K Nearest Neighbours: The neighbours taken in this algorithm were $n = 10$. The algorithm was then fed the whole training dataset.

3) Support Vector Machine: Their were different hyperparameters to be considered for this like, C value and kernel type. Grid search was done for the same and picking up the best ones, the model was finalised.

E. Validation

1) Logistic Regression:

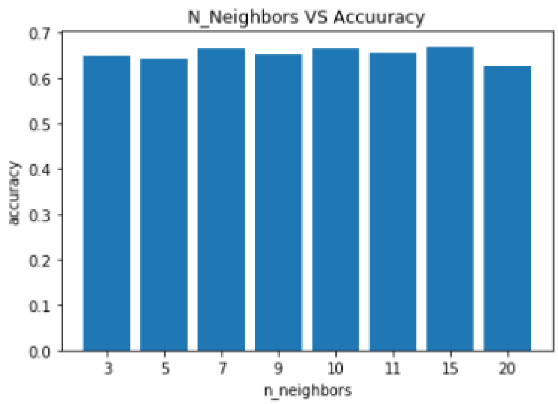

Fig. 1. Checking for various values of N (K Nearest Neighbour)

2) K Nearest Neighbours: The model was validated for different values of N (Neighbours) Fig. 1 describes the results from the validation. It is apparent from the figure that 3 neighbours are enough for this dataset.

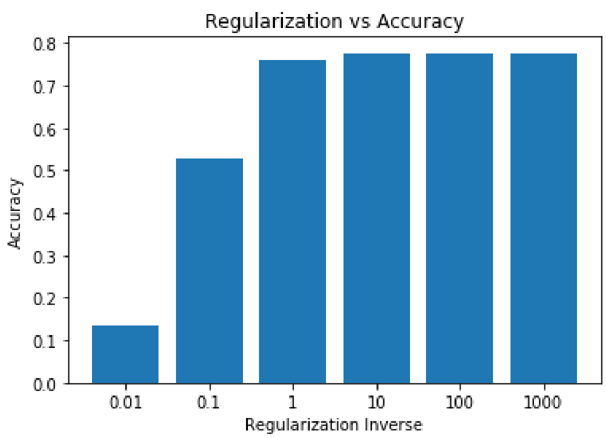

Fig. 2. Checking for various values of C (Support Vector Machine)

3) Support Vector Machine: The model was validated for different values of C (Regularisation Inverse) Fig. 2 describes the results from the validation.

IV. Results

To precisely find out which model is the best in detecting what Genres, we calculated the Confusion Matrix for all of them

1) Logistic Regression:

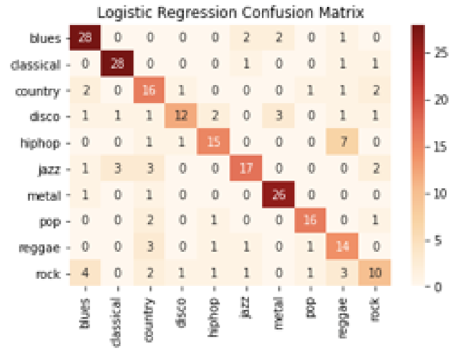

Fig. 3. Confusion matrix for Logistic Regression

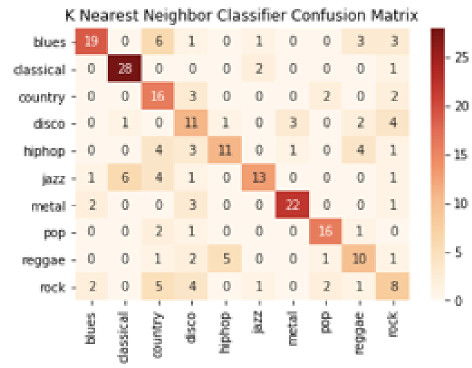

Fig. 4. Confusion matrix for K Nearest Neighbours

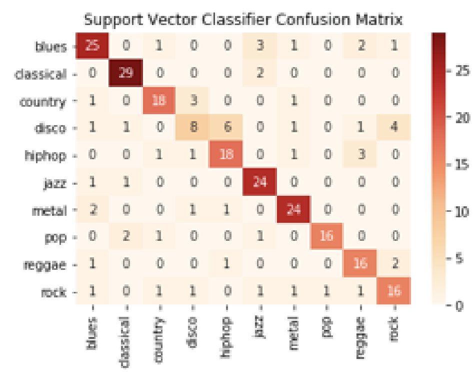

Fig. 5. Confusion matrix for Support Vector Machine.

V. Conclusion

	Logistic Regression	KNN Classification	Support Vector Classifier
precision	0.725	0.64	0.77
recall	0.737	0.62	0.78
f1-score	0.728	0.62	0.77

Fig. 3. Direct Comparison between Algorithms

Finally, we checked for Precision, Recall and F1 Score for these algorithms to make direct comparisons. Fig. 3 describes the same.

From the figure, it becomes obvious that SVM is the best choice for the problem and dataset. Hence, for genre classification, our team shall go forward with this model

Future goal of our team shall be to work on the second module of this project i.e. Mood Tagging. Mood Tagging shall be done by identifying and analysing lyrics of a particular song. Word Embeddings shall be used to correctly map words with their meanings. Just as SVM has been finalised for Genre Classification, we shall finalise an algorithm for Mood Tagging. When Algorithms shall be finalised for both the modules, the goal of our team shall be to eliminate the use of $sklearn$. We shall program the algorithms ourselves and further train, test and validate.

REFERENCES

[1] T. Nkambule and R. Ajoodha, "Classification of music by genre using probabilistic models and deep learning models."
[2] "Python project - music genre classification," Mar 2021. [Online]. Available: https://data-flair.training/blogs/python-project-music-genre-classification/
[3] A. Olteanu, "Gtzan dataset - music genre classification," Mar 2020. [Online]. Available: https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification
[4] "librosa¶." [Online]. Available: https://librosa.org/doc/latest/index.html
[5] P. Pandey, "Music genre classification with python," Dec 2018. [Online]. Available: https://towardsdatascience.com/music-genre-classification-with-python-c714d032f0d8
[6] "learn: machine learning in python - scikit-learn 0.16.1 documentation." [Online]. Available: https://scikit-learn.org/