MUSIC GENRE CLASSIFICATION

Rohit Kumar, Kethireddy Harshith Reddy, Manujendra Meena
B20AI035,B20AI018,B20CS035
Indian Institute Of Technology Jodhpur
Pattern Recognition and Machine Learning (CSL2050)
Course Project

Abstract - This report expresses our experience with music genre classification of different genres of music from audio files. The data set contains a collection of 10 genres of 100 audio files each with a length of 30 seconds each. We have extracted features through Librosa from the audio files and then tested various models from which LGBM gave the best accuracy.Trained the data set on LGBM and made predictions on the basis of this dataset.

INTRODUCTION

Everyone listens to music and so music classification is of the most used classifier. The aim of this project is to make a classifier that classifies music based on their genres. For this, there are many features of music that have to be taken into consideration like amplitude , frequencies , harmonics and tempo. This classifier can be further extended into a song recommendation system and monitoring the popularity of songs with a classification of different genres they belong to.

Datasets :

GTZAN: Audio files with ".wav" format have been used.

The content of the data set:

- Collection of 10 genres with 100 audio files each.
- Each Audio file is **around** 30 seconds long.

Image Dataset :

- Image representation of each audio file.

Other datasets

- 2 CSV files with all the features of the audio files.

DATA PREPROCESSING

During the preprocessing of the data it was found that there was an audio file that was corrupt named "jazz.00054.wav". We encoded the 'label' column and splitted the data where the train data had 70% of the original data and the test data had the remaining 30%.

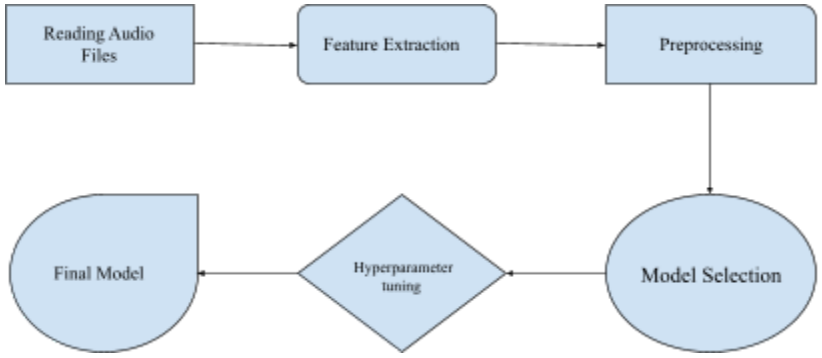
TYPES OF CLASSIFIERS

- **Logistic Regression:** One of the most well-known supervised learning techniques, as well as a popular binary classification technique.
- **SGD Classifier:** This estimator uses stochastic gradient descent (SGD) learning to create regularized linear models: the gradient of the loss is estimated one sample at a time, and the model is updated along the way using a decreasing strength schedule (aka learning rate).

- **SVC(Support Vector classifier):** The fit time scales at least quadratically with the amount of samples, and beyond tens of thousands of data, it may be unworkable.
- **KNNeighbors Classifier:** It's a straightforward supervised machine learning technique that can address both classification and regression issues.
- **GaussianNB:** It's a form of NB algorithm that's unique. When the features contain continuous values, it's employed particularly.
- **Decision Tree Classifier:** It is a type of supervised learning algorithm that can be used for both regression and classification problems.
- **AdaBoost Classifier:** It is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.
- **Random Forest Classifier:** It is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.
- **XGBoost Classifier:** It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.
- **LightGBM classifier:** It is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithm, used for ranking, classification and many other machine learning tasks
- **MLP Classifier:** It iteratively since at each time step the partial derivatives of the loss function with respect to the model parameters are computed to update the parameters.

PROJECT PIPELINE

The following flowchart is the entire pipeline which has been used throughout on the dataset:



BUILDING THE MODEL

The data has been split into 30% test data and 70% train data. The eleven models have been built and compared based on their Cross Validation scores. This was done to test overfitting or underfitting of models.

Below are the Cross Validation Scores (before hyperparameter tuning):

MODEL	CV Score	Runtime
Logistic Regression	0.4481030780243378	2min 35.7 sec
SGD Classifier	0.15232641374373657	3.8sec
Support Vector Classifier(SVC)	0.2761632068718683	25.3sec
KNNeighbours Classifier	0.270722977809592	1.4sec
GaussianNB Classifier	0.4226198997852541	0.1sec
DecisionTree Classifier	0.628489620615605	1.7sec
AdaBoost Classifier	0.466857551896922	9.9sec
Random Forest Classifier	0.8545454545454545	15.6sec
XGBoost Classifier	0.8913385826771654	58.6sec
MLPClassifier	0.20687186828919116	11.3sec
LightGBMClassifier	0.9079455977093772	17.7sec

MODEL SELECTION AND PARAMETER TUNING

Model Selection:

Light Gradient Boost Classifier was chosen as the best model because it has the highest CV score i.e, 90.79 among all of the 11 classifiers we used. This is the CV Score we've obtained without hyperparameter tuning. Random Forest and XGBoost had high CV Scores as well but the runtime of LightGBM was considerably low and had high CV score as well compared to XGBoost while when compared with Random Forest Classifier, it has around 5% higher accuracy and only a difference of 2sec run time which thereby makes **LightGBM** the **best model** for this dataset as it has a better performance than Random Forest Classifier.

Parameter Tuning:

Parameter tuning has been performed on the selected model (Light Gradient Boost) using Randomized Search. The parameters `max_depth`, `learning_rate`, `path_smooth` and `max_bin` when used together have resulted in maximum CV score (`0.938877755511022`).

Final Predicted Model:

We stored the pre-trained models which we've used earlier in a .pkl file and, for the predictions, we split every audio file into 3 seconds and did the prediction on all of the splits and took the mode of the predictions and gave it as the final prediction for the audio file.

CONCLUSIONS & OBSERVATIONS

Light Gradient Boost Classifier, Random Forest Classifier and XGBoost were the best performers among all 11 as the remaining gave accuracies below 70% while these gave an accuracy above 85%. There were audio files which ran for a time less than 30 seconds and a few which were above 30 seconds. To overcome this issue, we trimmed all of the audio files to 30 seconds and then split each of the 30-seconds file to 10 parts each of 3 seconds for training and future predictions.

REFERENCES

1. https://scikit-learn.org/stable/
2. https://librosa.org/doc/main/feature.html
3. https://pythonguides.com/python-tkinter-read-text-file/
4. https://towardsdatascience.com/understanding-boxplots-5e2df7bcbd51
5. https://www.geeksforgeeks.org/how-to-create-a-seaborn-correlation-heatmap-in-python/
6. https://stackoverflow.com/questions/63997969/attributeerror-module-librosa-has-no-attribute-output
7. https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

CONTRIBUTIONS

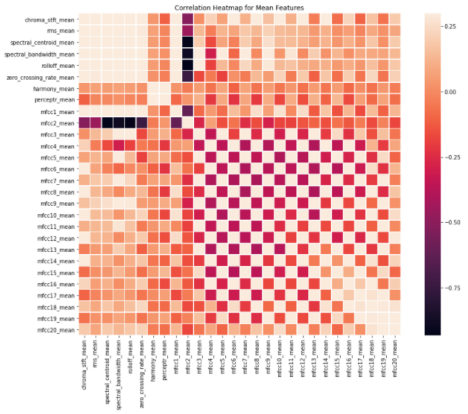
Rohit Kumar (B20AI035): Feature extraction, model selection, App-interface, Report, Imprinting new ideas and finding out mistakes, EDA.

Kethireddy Harshith Reddy (B20AI018): EDA & visualizations, Feature extraction, model selection, report, Imprinting new ideas.

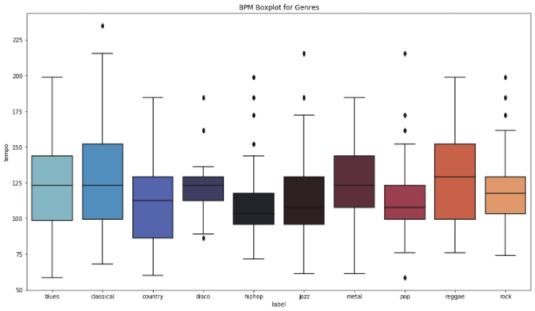
Manujendra Meena (B20CS035): Report, model selection, Visualizations.

PLOTS

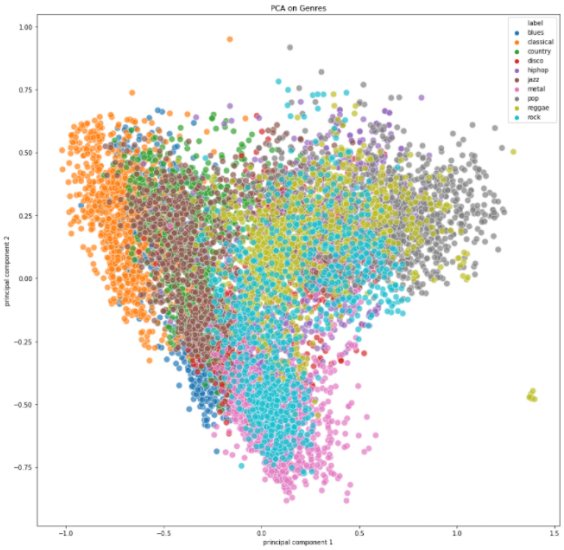
Correlation Heatmap for feature means:



Boxplot:



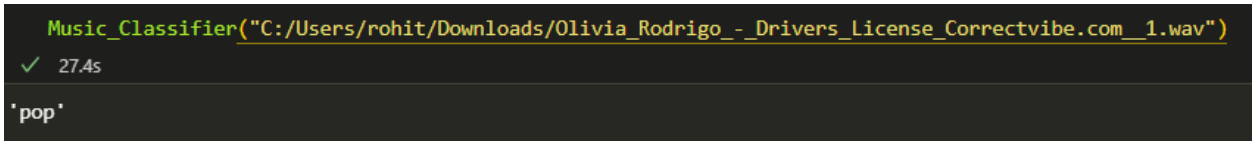
Class-wise visualization:



BONUS WORK

- We tested modern songs on our trained models which were having varying song durations and predicted their genre as an output which was the most accurate among one of the ten genres we were given in the dataset.

The following are a few of the songs on which we've implemented our models to predict their genre which matched:



- We made an interface which takes the .wav audio file of **any song** as an input and classifies the song into one of the 10 genres which we have in the dataset. We used python as a programming language for the same.