HARMONIC SOUND ANALYSIS : Musical Genre Classification

National School of Applied Science – Al Hociema
Moudni Houda – houda.moudni@etu.uae.ac.ma

ABSTRACT

Many music listeners create playlists based on genre, leaving potential applications such as playlist recommendation and management. Despite previous study on music genre classification with machine learning approaches, there is still room to delve into and build sophisticated models for Music Information Retrieval (MIR) problems. In this work, we apply a variety of machine learning techniques on the GTZAN dataset to classify 10 music genres given input features from music tracks, raising classification accuracy by more than 20% compared to the previously proposed baseline model. Also highlighting the importance of feature selection and parameter tuning in the performance of the model.

Index Terms— Feature Extraction, Genre Classification, Audio Signal Processing, Machine Learning for Audio Analysis

1. INTRODUCTION

Jazz, rock, blues, classical.. These are all music genres that people use extensively in describing music. Whether it is in the music store on the street or an online electronic store such as Apple's iTunes with more than 2 million songs, music genres are one of the most important descriptors of music. This dissertation lies in the research area of Automatic Music Genre Classification which focuses on computational algorithms that (ideally) can classify a song or a shorter sound clip into its corresponding music genre. This is a topic which has seen an increased interest recently as one of the cornerstones of the general area of Music Information Retrieval (MIR). Other examples in MIR are music recommendation systems, automatic playlist generation and artist identification. MIR is thought to become very important in the nearest future (and now!) in the processing, searching and retrieval of digital music. A song can be represented in several ways. For instance, it can be represented in symbolic form as in ordinary sheet music. In this dissertation, a song is instead represented by its digital audio signal as it naturally occurs on computers and on the Internet.

Given the raw audio signal, the next step is to extract the essential information from the signal into a more compact form before further processing. This information could be the tempo or melfrequency content and is called the feature representation of the music. Note that most areas in MIR rely heavily on the feature representation. They have many of the same demands to the features which should be both compact and flexible enough to capture the essential information. Therefore, research in features for music genre classification systems is likely to be directly applicable to many other areas of MIR.

The rest of this paper is organized as follows. In the next section, the proposed feature extraction algorithm is described. The corresponding CS-based classifier is then introduced in Section 3. The experimental settings and results are detailed in Section 4 leading to conclusions in Section 5.

2. RELATED WORK

In MIR research community, there are various studies on establishing effective models for music genre classification. For example, Using MFCCs has become a popular way to approach this problem. I. Karpov[1] implemented the delta and acceleration values of the MFCCs, increasing the amount of information that can be collected from the data. There are other common methods that can be used to classify music, as demonstrated by previous studies in [2], that were not used in this project such as the Octave-Based Spectral Contrast (OSC) or Octave-Based Modulation Spectral Contrast (OMSC).

3. AUDIO ANALYSIS BACKGROUND

For humans, basic aspects in the music such as melody and rhythm are likely to be used in the classification of music and these are also often modelled in the automatic systems. *The feature part* in the automatic system is thought to capture the important aspects of music. The final human classification is top-down cognitive processing such as matching the heard sound with memories of previously heard sounds. The equivalent in the automatic system to such matching with previously heard sounds, is normally the classifier which is capable of learning patterns in the features from a training set of songs.

Finding the right features to represent the music is arguably the single most important part in a music genre classification system as well as in most other music information retrieval (MIR) systems. It requires the understanding of the fundamental of the audio signals and the journey from an analog audio signal to a numeric representation which what is represented as analog-to-digital conversion (ADC).

In the first step, the continuous analog signal, which represents sound waves as voltage fluctuations, undergoes sampling. This process involves capturing snapshots of the analog signal at regular intervals, creating discrete data points in time.

Furthermore, each sample's amplitude, which represents the signal's strength at a particular moment, is quantized into a finite number of digital values. This step assigns numerical values to these amplitude levels, converting the analog amplitude into digital codes.

Lastly, the quantized values are encoded into binary digits (bits) to create a digital representation. Higher bit depths allow for more precise representation by providing a greater range of amplitude values.

This resulting digital representation, often referred to as a waveform or digital audio signal, is a series of discrete numerical values that approximate the original analog signal.

4. AUDIO FEATURES FOR MUSIC CLASSIFICATION

Audio songs, in the context of digital representation, are typically treated as discrete signals. When you listen to a song on a digital device, it's represented as a sequence of discrete samples captured at regular intervals.

As mentioned in the previously, feature extraction is the process of extracting the vital information from a (fixed-size) time frame of the digitized audio signal. These features can be divided into 3 forms:

Time Domaine Features:

relate to how a signal behaves and varies over time. They focus on the signal's characteristics and attributes in the time dimension. These features provide information about how the signal's amplitude or strength changes at different moments or intervals, without delving into its frequency content.

a. **RMS Energy:** RMS stands for "Root Mean Square" and refers to the square root of the energy expended, or the total amount of energy put out divided by the total amount of energy received.
For a signal x(t) over $x[n]$ over N sample, the E_{RMS} is calculated as follow:

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{n=0}^{N-1} x^2[n]}$$

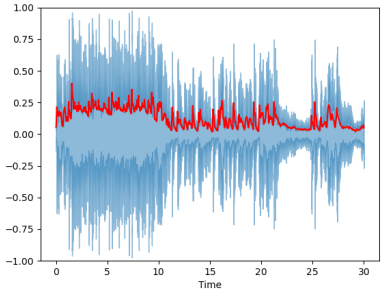

Figure 1: Root means square diagram of a rock music

b. **ZCR**: Zero-Crossing Rate gives us a measure of how often a change of value is seen. In other words, if you had a sequence consisting of either+1 or -1 values, how many times did your signal go from positive to negative? How many times did it go from negative to positive? The ratio of these numbers tells us how often a change of value is seen.
This can be formalized as:

$$\frac{1}{N-1}\sum_{n=1}^{N-1}|sign(x[n]) - sign(x[n-1])|$$

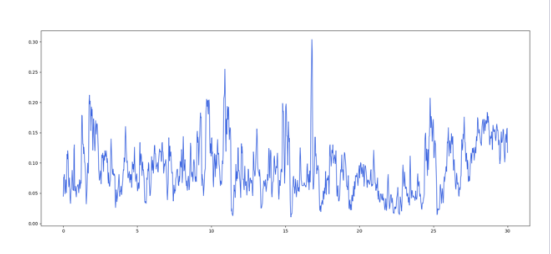

Figure 2: Zero Crossing rate diagram of a rock music

c. **Tempo**: The tempo of a piece of music fluctuates throughout the piece, so calculate the mean tempo. This is done by taking the mean value of the BPM values through several frames in the song.

Frequency Domain Features:

analyze a signal in terms of its frequency components. They provide insights into the distribution and strength of different frequencies present within the signal. Rather than focusing on how the signal changes over time, frequency domain features describe the signal's composition in the frequency dimension.

a. **Spectral Centroid:** The spectral centroid is the point in frequency space where the spectrum reaches its maximum value. In other words, it is the center of mass of the spectrum.

$$C = \frac{\sum_{n=1}^{N-1} f_x\, X(t)}{\sum_{n=1}^{N-1} X(t)}$$

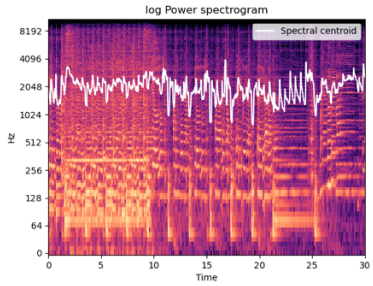

Figure 3: Spectral Centroid of a rock music

Here:

- f represents frequency.
- $X(f)$ is the Fourier Transform of the signal $x(t)$.
- $|X(f)|^2$ denotes the squared magnitude of the Fourier Transform.

b. **Spectral Bandwidth:** The spectral bandwidth is the range of frequencies within a sound wave. For

example, if you listen to a sine wave (a pure tone), you hear a single frequency. But if you play a guitar string, you'll hear many different tones because each note contains multiple frequencies.

$$B = f_{\max} - f_{\min}$$

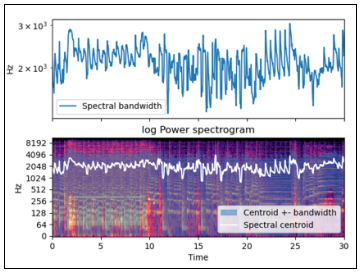

Figure 4: Spectral Bandwidth of a rock song

c. **Spectral Roll-off:** A spectral roll-off is the frequency at which a certain per cent of the total spectral energy lies. For example, if let's say 85% then that means that 15% of the spectrum lies above that point.

$$R = \arg\left\{\sum_{i=0}^{N-1} X[i] \quad \leq Percentage * \sum_{i=0}^{N-1} X[i]\right\}$$

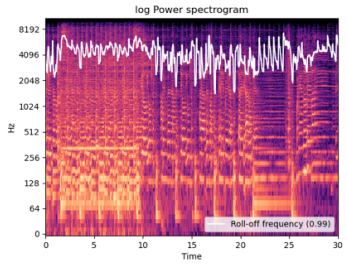

Figure 5: Spectral Roll-Off a rock music

Time-Frequency Domaine Features:

features aim to capture both time and frequency information simultaneously. They represent how the signal's characteristics evolve over time and across various frequencies. These features provide a joint representation, combining the advantages of time domain and frequency domain analysis.

a. **MFCC**: Mel-Frequency Cepstral Coefficients are used to obtain the parameters of speech. Since MFCCs were originally designed for voice recognition, they are used to extract features from sound samples. An example of such a feature extraction technique is Gaussian Mixture Modeling.

$$C_n = \sqrt{\frac{2}{N}} * \sum_{k=1}^{N} \log(E_k) * \cos\left(\frac{\pi n(k-0.5)}{N}\right)$$

Here:

- C_n represents the n-the MFCC coefficient.
- E_k is the log energy output of the k-the filterbank.
- N is the number of Mel filterbanks used.
- The summation is performed over all filterbanks.

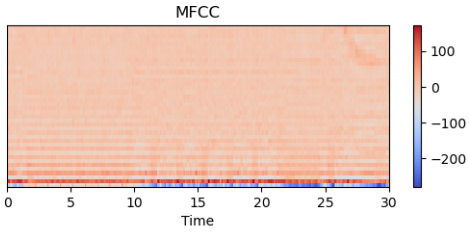

Figure 6: MFCCS diagram of rock song

b. **Chroma**: The chroma value is the sum of the energies of the 12 semitones represented by the pitch, regardless of the octaves. For example, G (G sharp with sharps) is 5/octaves 3 + 4/octaves 7/12.

$$C_i(t) = \sum_{\text{all } f \text{ bins corresponding to pitch class } i} X(f, t)$$

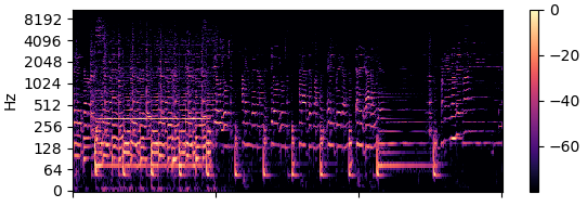

Figure 7: Chroma energy of a rock song

5. METHODOLOGY

In this project I worked on Raw audio dataset of 1000 songs, then segmented each audio of 30 seconds into 10 audios of 3 seconds.

The next step was about extracting the features I explained in the previous paragraph for each song, and store the data into a csv file.

Then the model building, where I attempt to build 3 models (Logistic regression, Random Forest, XGBoost) without parameters, then using grid search, I tried to adjust the hyperparameters to get the best accuracy possible. To finally, deployed the best model.

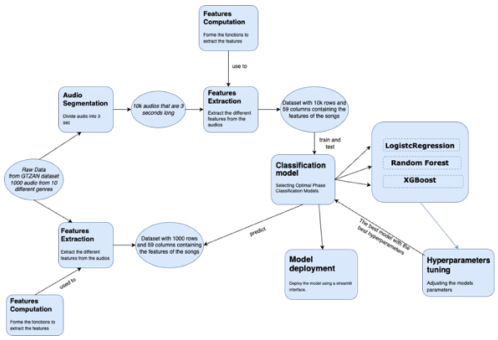

Figure 8: project's schema

6. DATASET AND FEATURES EXTRACTION

a DATASET GTZAN

The GTZAN Dataset – Music Genre Classification is a popular dataset for research on music genre classification. It consists of 1000 audio files, each 30 seconds long, from ten different music genres: *blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae,* and *rock*. The dataset is freely available on Kaggle and is widely used in research to test music genre classification algorithms. The GTZAN dataset contains audio files in WAV format with a sample rate of 22050 Hz and a bit depth of 16 bits. The audio files were sampled from the Million Song Dataset and preprocessed to ensure high quality and the absence of irrelevant noise. The GTZAN Dataset – Music Genre Classification is a balanced dataset, with each music genre containing 100 audio files. This makes the dataset suitable for evaluating the performance of music genre classification algorithms. The dataset has been widely used in research and has become a benchmark dataset for evaluating the performance of different machine learning algorithms.

b FEATUR.ES EXTRACTION

Segmentation into 3-Second Audio Clips

Before feature extraction, the audio dataset from GTZAN was segmented into 3-second audio clips. This segmentation process aimed to break down longer audio files into smaller, more manageable segments. Each 3-second segment represents a concise snippet of the original audio, enabling more granular analysis and feature extraction. As a result of this segmentation, the dataset expanded to encompass a larger number of individual audio clips, totaling around 9,990 files(audio file *jazz.00054.wav* is damaged).

Feature Extraction Process

The feature extraction process leveraged the *Librosa* library, a powerful tool for music and audio analysis in Python. Several key features were extracted from each segmented audio clip:

- Tempo
- RMS
- ZCR
- Harmonic and Percussive Component
- Spectral Centroid
- Spectral Bandwidth
- Spectral Roll-off
- Chroma STFT
- MFCC

Statistical Summary: Mean and Variance

For each of these extracted features, both the mean and variance were calculated. The mean provides a measure of the central tendency or average value of the feature across the 3-second audio segments. Meanwhile, the variance quantifies the degree of dispersion or spread of the feature values around the mean. Collecting both the mean and variance provides a comprehensive understanding of how these audio attributes vary and their typical values across the segmented audio dataset.

Data Storage

Following feature extraction and statistical calculations, the resultant mean and variance values for each feature, along with the associated filename and label information, were compiled into a structured CSV file. This CSV file serves as a structured repository of analyzed audio features, enabling further analysis, modeling, or exploration of the dataset using machine learning algorithms or statistical techniques.

7. DATA PREPROCESSING

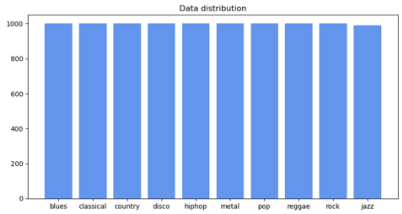

Figure 9: Data Distribution

The dataset exhibits a notably balanced distribution across various music genres, with each genre represented by a substantial count of instances. Specifically, the majority of genres, including blues, classical, country, disco, hip-hop, metal, pop, reggae, and rock, are uniformly distributed, each comprising 1000 instances. However, it is worth noting a slight deviation from this uniformity in the 'jazz' category, which accounts for 990 instances. This difference is due to the damaged audio file *jazz.00054.wav,* that was deleted earlier from the GTZAN Dataset. Despite this minor variance, the dataset largely maintains a balanced distribution, with each genre adequately represented for machine learning analysis.

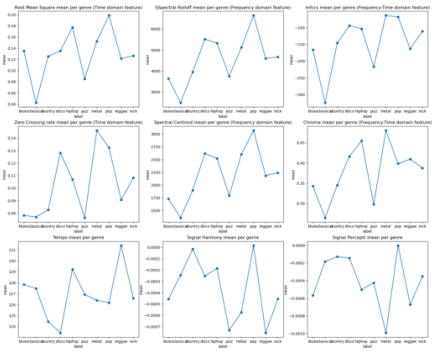

Figure 10: features variance in comparison with the music genres

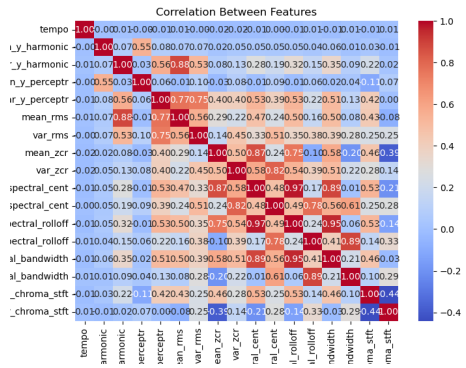

Figure 11: Features correlation

We've conducted a feature correlation analysis, revealing that a substantial portion of the features selected show limited correlation. This outcome aligns with our deliberate and meticulous feature selection process. Interestingly, among the features that do exhibit correlation, we find them to hold significant relevance in providing additional insights into the audio file characteristics. Consequently, considering the importance of these correlated features and their informative value, there's no immediate necessity to employ Principal Component Analysis (PCA) or reduce the dimensionality of the features dataset.

8. APPROACHES AND MODEL SELECTION

a CLASSIFIERS

In the pursuit of optimal classification performance, a variety of classifiers were considered, each offering distinct capabilities. This selection process aimed to leverage diverse algorithmic strengths in handling the intricacies of a multi-class classification task.

- **LOGISTIC REGRESSION MODEL**

The Logistic Regression (LR) model, a fundamental linear classifier, is traditionally tailored for binary classification. In the context of this multi-class problem, LR operates via a one-vs-rest strategy. It trains 10 individual binary classifiers, each discerning a specific class from the rest. During testing, the model evaluates the probabilities from these classifiers, selecting the class with the highest probability as the predicted label.

The logistic function governs the LR model's decision boundary, mathematically expressed as:

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta 0X0 + \beta 1X1 + \cdots + \beta nXn)}}$$

⇒ P(Y=1 | X) : denotes the probability of belonging to class 1 given predictors X .

⇒ $\{\beta 1, \beta 2, \ldots, \beta n\}$: represent the coefficients for the predictors.

⇒ $\{X_1, X_2, \ldots, X_n\}$: signify the predictor variables.

- **RANDOM FOREST MODEL**

Random Forests, operating as an ensemble of decision trees, embody the essence of bagging by creating multiple trees from random subsets of the dataset. Predictions are derived from a multitude of decision trees, each contributing its own assessment. In addition to output classes, probabilities of predictions are also provided, signifying the likelihood of a sample belonging to each category.

- **XGBoost MODEL**

Extreme Gradient Boosting (XGBoost), a scalable and distributed gradient-boosted decision tree (GBDT) algorithm, serves as a pinnacle in machine learning systems for regression, classification, and ranking challenges. Its underlying methodology revolves around assembling numerous weak learners into a formidable strong learner. Each decision tree is trained on a subset of the data, and their collective predictions merge into the final classification.

Each model's distinct approach and underlying mathematics were rigorously considered to ascertain the most suitable candidate for the multifaceted task at hand. Their diverse functionalities, ranging from linear boundaries to ensemble learning strategies, offered a comprehensive spectrum of classification methodologies to explore.

c MODEL'S EVALUATION

i MODELS WITHOUT HYPERPARAMETERS

Model Without Parameters	Accuracy
LOGISTIC REGRESSION	0.703063
RANDOM FOREST	0.805580
XGBoost	**0.829542**

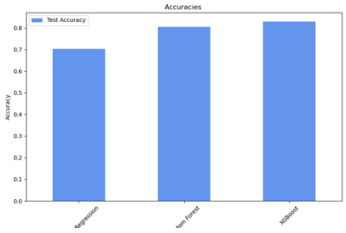

Figure 12: Accuracies before the hyperparameters tuning

The provided accuracies reflect the performance of machine learning models applied to the 3secondes dataset comprising 9,990 rows and 59 columns, encompassing diverse audio features. After analyzing (Assure that the data is equally distributed, and the features interact differently with different genres), standardizing and encoding the dataset, the models building begin without parameters.

Upon evaluation, the models exhibited distinct levels of predictive accuracy on this expansive dataset. Logistic Regression, with an accuracy of 72.6%, while Random Forest notably improved the accuracy to 85.2%, leveraging the power of ensemble learning based on decision trees. The subsequent model showcased even higher accuracy rate, with XGBoost closely trailing at 88.8%.

These models, ranging from traditional to cutting-edge techniques, demonstrated varying degrees of sophistication in handling the intricacies of the audio features. XGBoost, in particular, showcased remarkable performance, indicating its adeptness in capturing intricate patterns within the audio data, likely benefiting from their boosting methodologies. This disparity in accuracy underscores the significance of model selection, where more advanced algorithms like XGBoost exhibited superior performance in comprehending the nuances embedded within the diverse audio feature set, outperforming simpler models like Logistic Regression.

ii TUNING THE HYPERPARAMETERS

- **Grid search**

Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved.

For the hyperparameters tuning section we used grid search to choose the best hyperparameters with best accuracy.

Model	Hyperparameters	Accuracy
LOGISTIC REGRESSION	*C = 1, max_iter=500, penalty = 'l2'*	*0.725742*
RANDOM FOREST	*max_depth=20, n_estimators= 300*	*0.860422*
XGBoost	*random_state=42, learning_rate = 0.2, max_depth = 5, n_estimators = 300*	***0.891950***

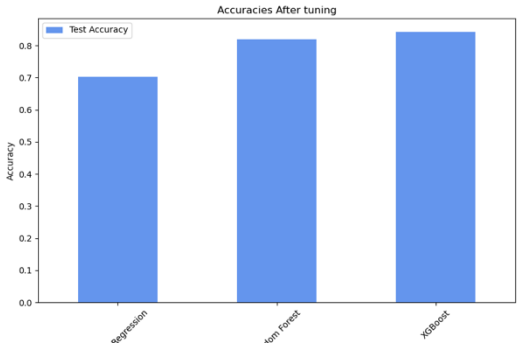

Figure 13: Accuracies after the hyperparameters tuning

After tuning the hyperparameters, the models retained their original rankings, affirming that XGBoost remains the optimal choice for deployment.

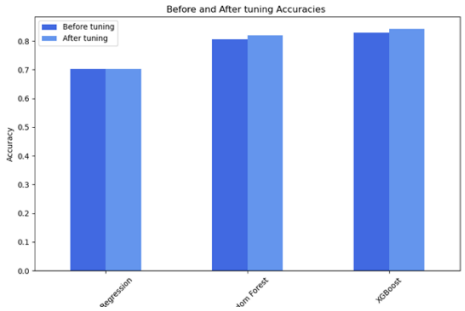

Figure 14: comparison between the accuracies after and before the hyperparameters tuning

- **Precision**: Precision is a metric that measures the accuracy of positive predictions made by a model. It is the ratio of true positive predictions to the total predicted positives (true positives + false positives). A high precision indicates that of all the positive predictions made by the model, how many were actually correct.

- **Recall**: Recall, also known as sensitivity, is a metric that measures the ability of a model to correctly identify all positive instances. It is the ratio of true positive predictions to the total actual positives (true positives + false negatives). High recall signifies the model's capability to capture a high proportion of actual positives.

- **F1-score:** The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances both precision and recall. It's particularly useful when the classes are imbalanced. F1-score reaches its best value at 1 (perfect precision and recall) and worst at 0.

Logistic Reression Classification Report:

	precision	recall	f1-score	support
0	0.68	0.68	0.68	330
1	0.91	0.94	0.92	330
2	0.63	0.59	0.61	330
3	0.58	0.61	0.60	330
4	0.70	0.61	0.65	330
5	0.78	0.83	0.80	327
6	0.78	0.87	0.83	330
7	0.78	0.80	0.79	330
8	0.65	0.63	0.64	330
9	0.51	0.47	0.49	330
accuracy			0.70	3297
macro avg	0.70	0.70	0.70	3297
weighted avg	0.70	0.70	0.70	3297

Figure 15: Logistic Regression report

Random Forest Classification Report:

	precision	recall	f1-score	support
0	0.85	0.83	0.84	330
1	0.85	0.96	0.90	330
2	0.77	0.78	0.77	330
3	0.76	0.78	0.77	330
4	0.88	0.81	0.84	330
5	0.83	0.82	0.83	327
6	0.88	0.94	0.91	330
7	0.79	0.82	0.81	330
8	0.76	0.81	0.78	330
9	0.82	0.65	0.72	330
accuracy			0.82	3297
macro avg	0.82	0.82	0.82	3297
weighted avg	0.82	0.82	0.82	3297

Figure 16: Random Forest Report

XGBoost Classification Report:

	precision	recall	f1-score	support
0	0.89	0.85	0.87	330
1	0.89	0.97	0.93	330
2	0.81	0.80	0.80	330
3	0.77	0.82	0.80	330
4	0.89	0.81	0.85	330
5	0.84	0.86	0.85	327
6	0.91	0.91	0.91	330
7	0.85	0.86	0.86	330
8	0.80	0.80	0.80	330
9	0.76	0.74	0.75	330
accuracy			0.84	3297
macro avg	0.84	0.84	0.84	3297
weighted avg	0.84	0.84	0.84	3297

Figure 17: XGBoost Report

After a thorough comparison of the three reports, it's evident that XGBoost consistently achieved the highest precision, recall, and F1-score across the majority of classes.

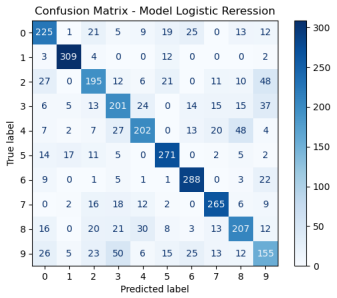

Figure 18: Logistic Regression Confusion matrix

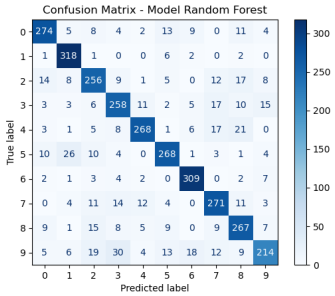

Figure 19: Random Forest Confusion matrix

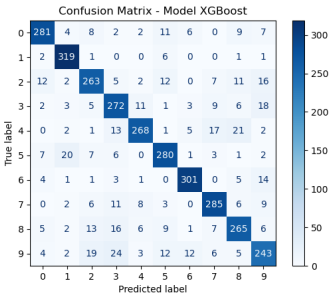

Figure 20: XGBoost Confusion matrix

In analyzing the confusion matrices, it's apparent that XGBoost demonstrates notable accuracy in predicting the correct genres for the majority of songs.

9. MODEL DEPOLYMENT

After selecting the best performing model, I deploy it using the library *pickle* in my *streamlit* interface.

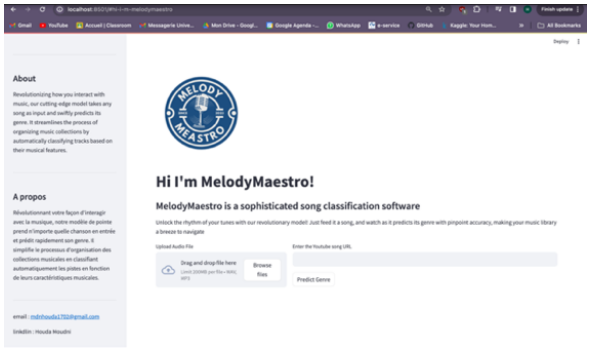

Figure 21: home page

MelodyMaestro, a streamlit application, where I deployed my model, to classify songs into various genres effortlessly. The streamlit interface offers seamless navigation, presenting an intuitive user experience.

Upon entering the application, users encounter an 'About' section on the left, providing insights and details about the software. The main panel boasts a versatile file uploader, allowing users to upload their songs directly. Additionally,

there's a convenient text field enabling users to input YouTube song URLs for instant genre classification.

a YOUTUBE SONG

Figure 22: Predict a song from YouTube

Users provide a song URL, enabling the software to download the song directly from YouTube for analysis. The *features_extraction* function parses the downloaded song, extracting crucial audio features that characterize the song's musical attributes. These features could encompass tempo, spectral characteristics, harmonic attributes, and more. Standardizing the extracted features ensures uniformity in their scales and ranges, enhancing the model's ability to interpret and learn from this data effectively. Leveraging the trained machine learning model, the software predicts the music genre based on the standardized features.

b UPLOADED SONG

Figure 23: Predict a file from your local

This user-friendly functionality empowers individuals to analyze and classify their local music files seamlessly, providing quick and accurate genre predictions based on the song's unique audio features.

10. CONCLUSION AND FUTURE WORK

This research work provides the details of an application which performs Music Genre Classification using Machine Learning techniques. The application uses a XGBoost model to perform the classification. Based on several music features extracted from the GTZAN dataset. This is done by using the libROSA package of python. A piece of software is implemented which performs classification of songs from YouTube platform into their respective genres.

The extension of this work would be to consider bigger data sets and also, with time the style represented by each genre will continue to change. So, the objective for the future will be to stay updated with the change in styles of genres and extending our software to work on these updated styles. This project can also be extended to work as a music recommendation system depending on the mood of the person.

11. ACKNOWLEDGMENT

I would like to express my sincere gratitude to Mr. Aziz Khamjane for his invaluable guidance, support, and mentorship throughout the duration of this project. His expertise, encouragement, and insightful feedback were instrumental in shaping the direction and success of this endeavor.

REFERENCES

[1] Karpov, Igor, and Devika Subramanian. ”Hidden Markov classification for musical genres.” Course Project (2002).

[2] Lee, Chang-Hsing, et al. ”Automatic music genre classification using modulation spectral contrast feature.” Multimedia and Expo, 2007 IEEE International Conference on. IEEE, 2007.

[3] Peter Ahrendt, ‘Music Genre Classification Systems’ , A Computational Approach(2006)

[4] Rajeeva Shreedhara Bhat ,Rohit B. ,Mamatha K. , Music Genre Classification, (2020)