Music Genre Classification

Members

The project has been taken as a group effort of Sujal Suri (2022514) and Divyasha Priyadarshini (2022180) over the course of Statistical Machine Learning - Winter 2024.

Problem

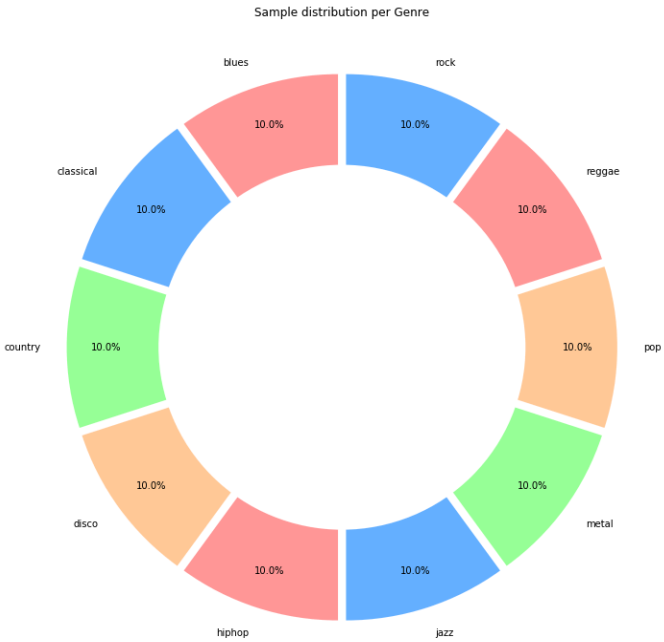
Classification of audio/music data into categories or predefined genres. This is a supervised learning problem, where the model learns from labelled examples in the dataset. This can be applied in several modern day applications. Spotify's Music recommendation system, curated playlists "for you", Amazon music's "radio" based on a recently played song are all based on this classification problem. Genre-based music recommendation and content based music retrieval are also growing applications of this problem.

Literature Review

G. Tzanetakis and P. Cook worked on the GTZAN dataset to extract features representing timbral texture, rhythmic content and pitch content in 2002. They achieved an accuracy of 0.61. Dong and Mingwen achieved an average accuracy of 0.70 with CNNs that parallel the human accuracy for this work (70 percent as well). Changsheng Xu and team applied SVM over the musical genre classification problem to achieve over 0.90 accuracy for a 4 genre problem.

Dataset Detail

GTZAN: GTZAN is like the MNIST of the music data world. It is a popular publicly available dataset that comprises 1000, 30 second long audio tracks belonging to 10 different genres. The dataset is balanced and homogeneous in nature meaning that there are 100 tracks for each genre. The specification of all audio tracks is as follows: 22050 Hz Mono 16-bit in .wav format. Total size of the dataset is 1.23 GBs.



Experimental Settings

Development Environment: Google Colab and Jupyter Notebook

Programming language: Python

Libraries/Frameworks:

```
import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import librosa.display

Import sklearn

import xgboost as xgb

import tensorflow.keras as keras
```

Methodology

The GTZAN dataset consists of 1000 samples of about 30 seconds each. We started by training the model on just the mel-spectral. We trained multiple models like RandomForest , Neural Network, with mediocre accuracy reaching a maximum of 55 - 60 with random forest after applying PCA.

We then realised The dataset is very small in terms of the number of samples, but the size of each sample is too large to be processed accurately. We **augmented** the dataset by dividing each sample into 3 second audio snippets. We considered converting all the smaller audio into mel-sepstral and then applying pca similarly but due to memory restraints, we failed to compute the necessary parameters.

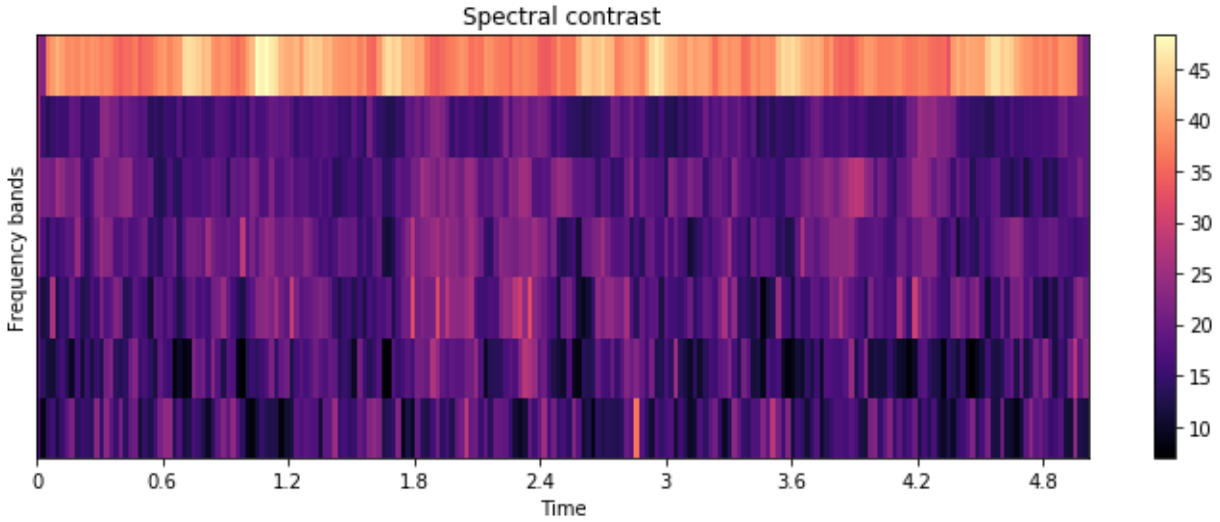
The total number of such samples was now 10,000, approximately 1000 in each class. We then considered a vector for each datapoint with features such as the length, **chroma_shift**, **rms**, **spectral centroid**, **spectral bandwidth**, **roll off, zero crossing rate**, **mfcc coefficient**. We consider the mean and variance of each of the above plots, totaling to a data point vector of length 60.

We applied **PCA** on the dataset, which **did not** lead to any significant **improvements**. Finally we trained the model on Naive-Bayes, **Random Forest Classifier**, Random forest with k-fold validation, **XGBoost**, XGBoost with k-fold validation, **K nearest neighbours** algorithm, and **Neural Networks**.

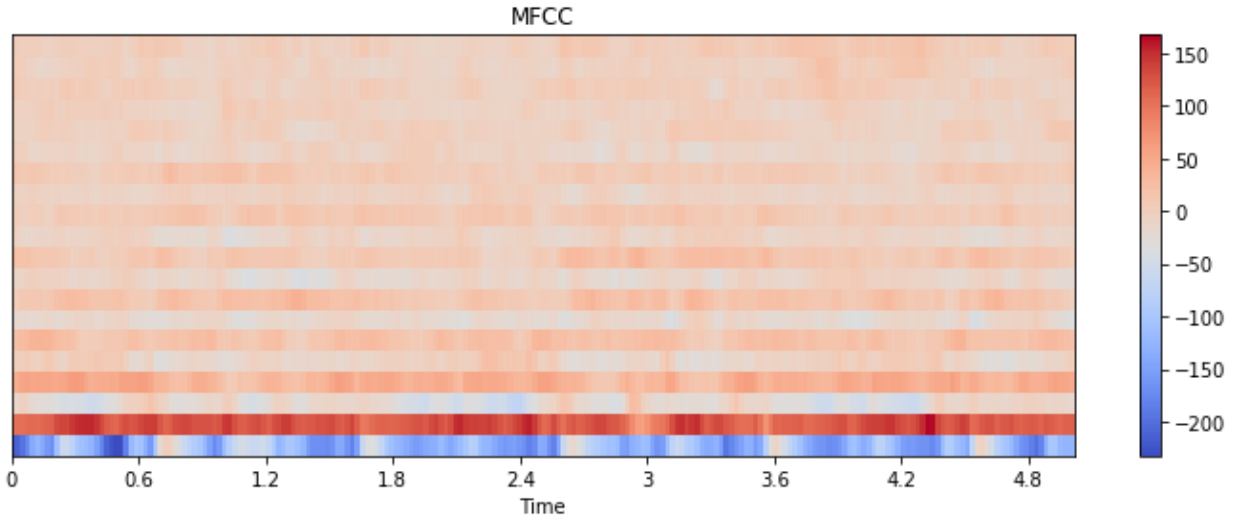
Features

Visualisation of some of the features we considered the data-point

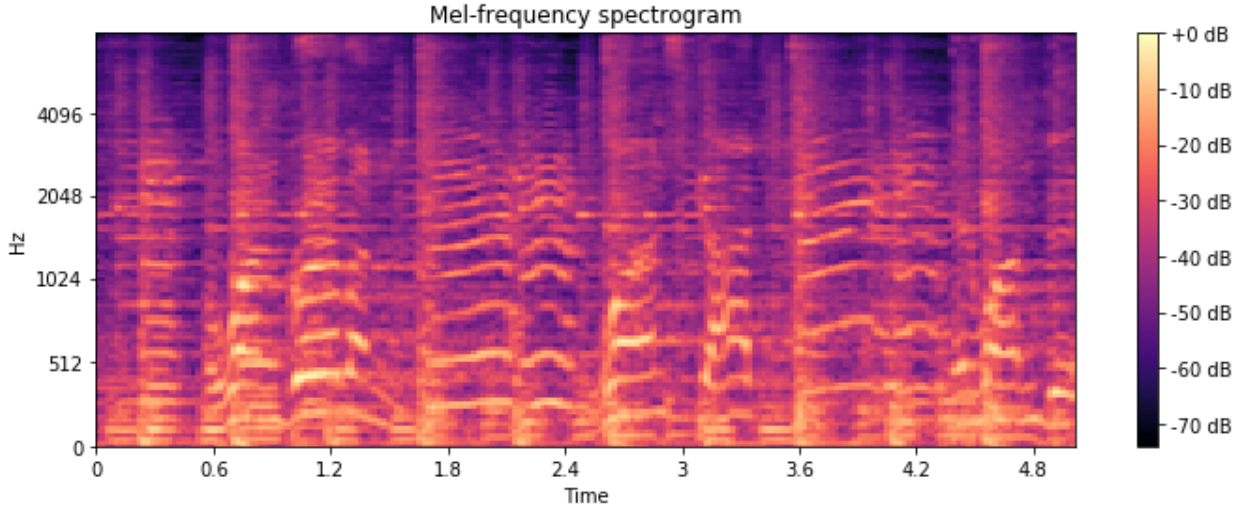
- Spectral Contrast



- MFCC coefficients



- Mel-frequency spectrogram



Model Selection

We considered many models, performing 5 fold cross validation on some in hopes of getting a better result:

1. RandomForest with cross validation (84.04%)
2. Neural Network (86.75%)
3. XGBoost with cross validation (87.78%)
4. Naive-Bayes (51.51%)
5. KNN Classifier (89.32%)

Analysis of Results

On training the models we realised that the dataset was too large in size and **trimming** the audio was the preferred way to go. Also considering a variety of qualities of the sound is better instead of working on a single quality. The results we got for the dataset with sample size 3 seconds were significantly better than 30 seconds.

Though we considered multiple qualities, **PCA** didn't seem to be of much use. In fact, PCA leads to a loss of crucial discriminative information for complex audio data, and accuracies improved insignificantly on its usage.

Naive Bayes is a simple probabilistic classifier, and fails to capture the complexities of audio data, hence returning an accuracy or merely 51.51 percent.

On the other hand, Random Forest being an ensemble learning method is more suitable for such data and performs with an accuracy of 84.04%. We also gathered that XGBoost performs immensely well in classification tasks, and showed an accuracy of 87.78 percent. Neural networks were also able to capture the complexities of audio data with an accuracy of 86.75%.

The classifier that performed best in our observation was the K Nearest Neighbours classifier, with an accuracy of 89.32%. However, Convoluted Neural Networks may have performed better for a larger amount of epochs. However, that will take a considerable amount of time for execution.

Contribution

Both of us contributed equally to the project throughout its duration and gave meaningful insights regarding the possible explanations of the intermediate results we achieved

References

Tzanetakis, George Cook, Perry. (2002). Musical Genre Classification of Audio Signals. IEEE Transactions on Speech and Audio Processing. 10. 293 - 302. 10.1109/TSA.2002.800560.

Dong and Mingwen, “Convolutional Neural Network Achieves Human- level Accuracy in Music Genre Classification,” arXiv.org, 27-Feb-2018

Machine learning on Audio Files - Blog by Suhas Maddali
Geeks for geeks machine learning