Noorh Alkhraan
21 October 2021

Music genre Classification Report

Abstract:

The goal of this project was to utilize machine learning classification models to predict the genre of a given music features in order to to give more understanding and recognition of music genres. I worked with the data provided by one of the MachineHack Hackathon[1], leveraging its features and 11 classes with a random forest model to solve this multi-classification problem.

Design:

The project aims to investigate and classify genres of music, These classifications have the capacity to significantly improve our knowledge, recognition, and makes it easier for us to find music that suits our tastes, as well as for artists to sell their music in a way that distinguishes them from the competition. The data is acquired by one of the MachineHack Hackathon that demonstrates 11 types of music classes and 16 column features.

Data:

Training dataset: 17,996 rows with 17 columns. Some important features are popularity which explains how much popular a music is, danceability which indicates how much danceable the music is and energy.

The target variable is called 'Class' which includes a numeric encoding of the classes (from 0-10). The aim is to explore these feature to predict the genre class such as Rock, Indie, Alt, Pop and others.

Algorithms:

Feature engineering

[1] https://www.kaggle.com/purumalgi/music-genre-classification

1- Dealing with null values, by either replacing them with the mean, mode or median of the feature.

2- Exploring skewed features that will harm the accuracy of the classification and maintaining them by using log() function.

3- Checking for data imbalance, as it is happening when dealing with multi-class problems.

4- Scaling the data so that all values in the dataset have the same range of values. I used SMOTE to oversample the minority classes, and adding to them to the range of the majority class.

Models

Logistic regression, support vector machines, XGBoost and random forest classifiers were used before settling on random forest which is the model with strongest cross-validation performance.

Model Evaluation and Selection

The entire training dataset was split into 80/20 train and all scores reported below were calculated on the training portion only. The calculation of F1, recall and precision were taken as the average (macro) of all classes.

Final random forest scores: 14 features with 48147 instances

- Accuracy 0.75877
- F1 0 0.7492 macro
- precision 0.7438 macro
- recall 0.7614 macro

Tools:

- Numpy and Pandas for data manipulation
- Scikit-learn for modeling and scaling
- Matplotlib and Seaborn for plotting
- Imblearn for oversampling with SMOTE

Communication:

The slides contains visuals showing some features and found results.