Design or your idea

Classifying the Mode of a Sounds .

Problem and Background

Going in to this project, I knew I wanted to work with music audio. After doing some research, I settled on working with raw audio files by extracting features from mp3s. I chose to work with music data because it is a type of audio that can evoke emotion in addition to thought. When I listen to music, I sometimes ask myself: Why does a particular song make me feel happy or sad? The key of a song helps determine the feeling of a song. There are two parts that make up the key, the tonic note (also known as the root note or base note), and the mode (either major or minor). Going in to this project, I posed the question: Can I predict the mode?

Dataset

To collect my data, I downloaded more than 1,000 mp3s in the form of Billboard Top 100 collections. These varied by year, and represent what music was most popular at a given time. This step gave me raw audio files. From there I dove into Digital Signal Processing (DSP) and Music Information Retrieval (MIR). Using the Python-compatible libraries Librosa and Pydub, I sliced all of the mp3s into clips of the first 30 seconds of the song and extracted several audio features. I settled on the first 30 seconds after comparing model performance at 10 seconds, 20 seconds, 25 seconds, and 30 seconds.

The balance between the two classes of my target variable, Mode, were 59% Minor and 41% Major.

Project Design

One of my personal goals for this project was to create .py files in PyCharm that automated the process of extracting data and building and analyzing models. I am happy to say that I accomplished this goal and divided my project code into four files: extract.py, transform.py, model.py, and main.py, in addition to having these steps in Jupyter Notebook with visualizations.

Algorithm

In total, I built, trained, tested, and evaluated 13 classification models. The models include:

1. Logistic Regression Baseline
2. Logistic Regression Optimized with Grid Search
3. K-Nearest Neighbor Baseline
4. K-Nearest Neighbor Optimized with Grid Search
5. Support Vector Classifier Baseline
6. Support Vector Classifier Baseline (Normalized)
7. Support Vector Classifier Optimized with Grid Search (Normalized)
8. Support Vector Classifier Optimized with Random Search (Normalized)
9. Naive Bayes Classifier
10. Decision Tree Baseline
11. Decision Tree Optimized with Grid Search
12. Random Forest Baseline

13. Random Forest Optimized with Random Search

Tools

The main tools that I used in addition to Python, PyCharm, and Jupyter Notebook were Pandas, Librosa, Pydub, and Scikit-Learn. I decided to use Librosa and Pydub for working with raw audio files after doing several hours of research into the different tools available in the fields of DSP and MIR. Librosa has a wide variety of functions that allow for an in-depth analysis of audio files, which allowed me to become familiar with the specific challenges and fascinating parts of working with audio signal processing.