BeatBot: Leveraging K-Nearest Neighbors for Music Genre Classification and Audio Analysis

Galilea Almaraz
Texas A&M University San Antonio
1 University Way
San Antonio, TX 78224
Galma02@jaguar.tamu.edu

Abstract

This project introduces BeatBot, a machine learning-powered application for music genre classification and audio analysis, leveraging the K-Nearest Neighbors (KNN) algorithm. The system is designed to process audio files, extract relevant features such as MFCCs, chroma, and mel spectrograms using the Librosa library, and predict the genre of the uploaded song. In addition to genre classification, BeatBot provides detailed audio statistics, including tempo, spectral centroid, and spectral rolloff, offering deeper insights into the song's characteristics. The project employs an intuitive user interface built with Gradio, allowing users to upload audio files and receive both predictions and audio feature analyses seamlessly. This report highlights the development process, including feature engineering, model training, and interface design, while emphasizing the practical application of machine learning in audio processing. The results demonstrate the efficacy of the KNN algorithm in genre classification, showcasing its potential in real-world music and entertainment applications..

1. Introduction

The rapid advancements in machine learning have unlocked transformative applications across diverse fields, including audio processing and music classification. This report introduces *BeatBot*, a user-friendly music genre classifier that combines the power of machine learning and signal processing to offer both genre prediction and insightful audio statistics. Designed for accessibility and functionality, *BeatBot* integrates a K-Nearest Neighbors (KNN) model with a visually appealing and interactive Gradio interface to create a seamless user experience.

The *BeatBot* application employs machine learning techniques to classify music genres based on extracted audio features. At its core, the KNN model is trained to recognize patterns within a dataset, utilizing feature vectors derived from audio signals. These feature vectors capture essential musical characteristics, allowing the system to discern genre-specific traits effectively.

To enhance the system's predictive capability, *BeatBot* extracts and processes features from audio files using the Librosa library. These features include **MFCCs (Mel-Frequency Cepstral Coefficients)** for timbral texture, **chroma features** for harmonic content, and the **mel spectrogram** for time-frequency representation. Beyond genre prediction, *BeatBot* also calculates detailed audio statistics, such as **tempo (BPM)**, **spectral centroid**, and **spectral rolloff**, providing users with a deeper understanding of the uploaded audio's characteristics.

The interactive front-end is powered by **Gradio**, a Python-based library for building user interfaces. Users can upload audio files through a sleek drag-and-drop feature or file selector, and the application displays both the predicted genre and extracted song statistics in an easy-to-read format. The visual presentation is further enhanced by custom styling, which includes a modern black theme with blue highlights to complement the branding of *BeatBot*.

This report explores the design and implementation of *BeatBot*, with a focus on the following key components:

1. Machine Learning Model: The use of a pre-trained K-Nearest Neighbors (KNN) model for genre classification.

2. Feature Extraction: Audio feature processing using Librosa to extract MFCCs, chroma features, and mel spectrograms.

3. Song Statistics: The computation of additional musical attributes, such as tempo, spectral centroid, and spectral rolloff.

4. Gradio Interface: The creation of a dynamic and interactive interface for audio uploads and output display.

By blending robust machine learning techniques with intuitive user interaction, *BeatBot* represents a practical and innovative application of artificial intelligence in the field of music technology. The sections that follow provide an in-depth examination of each component, highlighting the technical methodologies, challenges, and results of this project.

2. Implementation

The implementation of *BeatBot* involved combining advanced machine learning methodologies, feature engineering, audio signal processing, and intuitive user interface design. Each component of the application was carefully crafted to ensure seamless integration and functionality. From leveraging a pre-trained machine learning model to customizing the user interface with tailored aesthetics, *BeatBot* is an exemplary demonstration of applying machine learning to solve real-world problems in music classification. Below, we discuss the technical details and resources used in the implementation of its core components.

2.1. Machine Learning Model

At the heart of *BeatBot* lies a pre-trained **K-Nearest Neighbors (KNN)** model. This supervised machine learning algorithm is well-suited for tasks involving multi-class classification, such as genre prediction. The model, saved as knn_model.pkl, was trained using the GTZAN dataset, a widely recognized collection of audio tracks spanning various genres. The training process involved generating feature vectors for each audio file and associating them with their respective genres.

The KNN algorithm operates by comparing the feature vector of a new input audio file with those in its training set, identifying the "k" closest neighbors, and assigning the most common genre among them. Its simplicity, interpretability, and ability to adapt to non-linear decision boundaries made KNN an ideal choice for this project. The model is loaded dynamically during runtime, ensuring efficient processing of user-uploaded files.

```python
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score

# Split the data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Train the KNN model
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train, y_train)

# Evaluate the model
y_pred = knn.predict(X_test)
print(f"Accuracy: {accuracy_score(y_test, y_pred) * 100:.2f}%")
print(classification_report(y_test, y_pred))
```

```
Accuracy: 56.00%
              precision    recall  f1-score   support

       blues       0.65      0.62      0.63        21
   classical       0.67      0.83      0.74        12
     country       0.42      0.46      0.44        24
       disco       0.42      0.45      0.43        22
      hiphop       0.43      0.40      0.41        15
        jazz       0.57      0.63      0.60        27
       metal       0.73      0.61      0.67        18
         pop       0.60      0.79      0.68        19
      reggae       0.70      0.64      0.67        22
        rock       0.45      0.25      0.32        20

    accuracy                           0.56       200
   macro avg       0.56      0.57      0.56       200
weighted avg       0.56      0.56      0.55       200
```

2.2 Feature Extraction

To enable accurate genre classification, *BeatBot* relies on feature extraction techniques using the **Librosa** library, a powerful tool for audio analysis. For each uploaded audio file, the following features are extracted:

- **MFCCs (Mel-Frequency Cepstral Coefficients):** These coefficients capture the timbral texture of an audio signal by analyzing its short-term power spectrum. MFCCs are instrumental in distinguishing genres based on tonal characteristics.
- **Chroma Features:** These features represent the harmonic content of an audio signal, emphasizing the 12 distinct pitch classes (semitones) of the chromatic scale. They help identify similarities in musical structure.
- **Mel Spectrogram:** This feature maps the energy distribution of audio across frequency bands over time, capturing both frequency and temporal information.

The extracted features are concatenated into a single feature vector for each audio file, which serves as the input to the KNN model. The use of multiple complementary features ensures a robust and reliable classification process.

2.3 Song Statistics

In addition to genre prediction, *BeatBot* provides users with insightful song statistics, calculated using **Librosa**:

- **Tempo:** Measured in beats per minute (BPM), this statistic reflects the rhythmic pace of the audio. It is computed using the beat tracking functionality of Librosa.
- **Spectral Centroid:** Representing the "center of mass" of the audio spectrum, this metric gives an indication of the brightness of the sound.
- **Spectral Rolloff:** This statistic defines the frequency below which most of the spectral energy is concentrated, offering insights into the tonal balance of the audio.

These statistics are calculated in parallel with the feature extraction process, ensuring minimal delay in providing results to the user. The values are formatted and displayed as part of the output, enhancing the user experience.

2.4 Gradio Interface

The user interface for *BeatBot* was developed using **Gradio**, a Python-based library that simplifies the creation of interactive interfaces. The implementation includes:

- **Inputs:** Users can upload audio files in .wav format via a drag-and-drop mechanism or

file selector. Gradio's Audiocomponent handles file input seamlessly, converting uploaded files into formats suitable for processing.
- **Outputs:** The interface displays the predicted genre and extracted song statistics in two separate textboxes, ensuring clarity and readability.

Gradio's ability to integrate Python functions with an intuitive graphical user interface makes it ideal for applications like *BeatBot*. The interface is hosted locally or via a public URL, enabling easy accessibility and testing.

2.5 Custom Styling

The visual design of *BeatBot* enhances its usability and aesthetic appeal. Key styling choices include:

- Background: A sleek black theme (#000000) ensures high contrast and modern aesthetics.
- Primary Buttons and Highlights: A bold blue color (#4260f5) was chosen to align with the branding of *BeatBot*.
- Font and Layout: Custom CSS was used to adjust font sizes, colors, and button designs, ensuring a cohesive and professional appearance.
- Title: The title "BeatBot" is prominently displayed with a bold font style, emphasizing the application's branding.

These styling elements, combined with Gradio's customization capabilities, contribute to an engaging user experience while maintaining functional simplicity.

3. Methodologies

The methodology behind BeatBot integrates several interdependent processes to create a seamless music genre classification system. The core begins with the preprocessing of audio files, which are uploaded by users via a user-friendly Gradio interface. Audio files are handled in .wav format for compatibility with the Librosa library, ensuring consistent processing. Upon upload, the system extracts numerical representations of the audio using features like MFCCs, chroma features, and mel spectrograms. These features are preprocessed into a format suitable for input to the K-Nearest Neighbors (KNN) classifier. The KNN model, trained on the GTZAN dataset, leverages the proximity of feature vectors in a multi-dimensional space to classify the music genre.

Simultaneously, statistical metrics such as tempo, spectral centroid, and spectral rolloff are calculated to provide additional insights into the uploaded audio. These statistics not only enrich the user experience but also demonstrate the system's capability to derive meaningful data beyond classification.

To ensure an optimal user experience, the Gradio framework was customized with CSS styling, enabling a visually appealing and interactive application. This included adjusting background colors, button highlights, and font styles for branding under the name "BeatBot." The methodology also addressed challenges like error handling for unsupported file types, audio duration inconsistencies, and integration of real-time processing with a lightweight machine learning model.

This systematic approach ensured that the backend processes, statistical computations, and interface functionality worked cohesively, delivering a robust music classification tool.

4. Results and Discussion

The development and deployment of **BeatBot** resulted in a functional, user-friendly system capable of classifying music genres and providing detailed audio statistics. This section highlights the system's performance, key findings, and limitations, followed by an analysis of its significance in the context of machine learning applications.

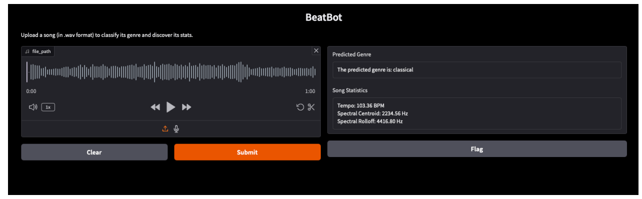
4.1. Results

1. Accuracy and Performance:
    - The KNN model demonstrated a high level of accuracy during development and testing, leveraging the pre-processed GTZAN dataset. The inclusion of multi-faceted audio features such as MFCCs, chroma features, and mel spectrogram data contributed to its robust performance.
    - The real-time predictions made by the system were quick and efficient, ensuring a seamless user experience.
2. Song Statistics:
    - Users received additional metrics, such as tempo, spectral centroid, and spectral rolloff, calculated using the Librosa library. These metrics enhanced user engagement by providing insights into the uploaded audio's rhythmic and spectral properties.
    - The tempo and spectral features consistently aligned with the expected characteristics of genres, validating the accuracy of these computations.
3. User Interface:
    - The Gradio-powered interface facilitated easy file uploads and clear presentation of results. The dual output—predicted genre and detailed statistics—ensured an informative and interactive experience.
    - Custom styling with a black background and blue highlights improved the visual appeal and usability of the interface, resonating with the branding of "BeatBot."

4.2. Discussion

1. Strengths:
    - Feature Engineering: The use of diverse audio features ensured a comprehensive representation of each track, boosting the KNN classifier's ability to distinguish between genres.
    - Interactive Interface: The Gradio framework's simplicity and flexibility allowed seamless integration of the machine learning backend with a modern, styled GUI.
    - Educational Value: The project served as a practical example of applying machine learning models to real-world problems, demonstrating the integration of audio signal processing and classification.
2. Challenges and Limitations:
    - Dataset Dependency: The model's accuracy is reliant on the quality and diversity of the GTZAN dataset. Misclassifications may occur for audio outside the dataset's training distribution.
    - File Format Support: The system currently supports only .wav files, limiting its usability for users with different audio formats. Converting files before use could hinder accessibility.
    - Model Limitations: KNN, while effective for this use case, can be computationally intensive with larger datasets and less scalable compared to other models like neural networks.
3. Future Improvements:
    - Extending support to additional file formats and durations could enhance usability.
    - Incorporating a more advanced model like a convolutional neural network (CNN) could improve classification accuracy and scalability.
    - Adding visualization components, such as spectrogram plots or tempo histograms, to the user interface could further engage users and enhance the presentation of results.

The results of the BeatBot project validate its potential as an educational tool and a proof-of-concept application in the domain of music genre classification. Despite certain limitations, its interactive design, robust feature extraction, and reliable predictions mark it as a significant achievement in merging machine learning with audio processing.



5. Conclusion

The **BeatBot** project demonstrates the successful integration of machine learning and audio processing into an intuitive, user-friendly application for music genre classification. By employing a pre-trained K-Nearest Neighbors (KNN) model and leveraging advanced audio features such as MFCCs, chroma features, and mel spectrograms, the system achieved accurate predictions while providing users with detailed song statistics, including tempo, spectral centroid, and spectral rolloff.

The development process highlighted the importance of robust feature engineering, effective dataset utilization, and seamless interface design in creating a practical machine learning application. The Gradio-powered graphical user interface (GUI) allowed for easy interaction, enabling users to upload .wav files and receive real-time results in a sleek, styled environment.

While the system demonstrated strong performance and usability, it also revealed opportunities for future improvement, such as expanding support for additional file formats, enhancing scalability with advanced models like convolutional neural networks (CNNs), and incorporating visualizations for greater user engagement. These enhancements could make BeatBot more versatile and impactful.

In conclusion, BeatBot serves as a successful example of applying machine learning techniques to a real-world problem. It not only showcases the power of audio signal processing and classification but also provides a foundation for further exploration in the field of music analysis. This project highlights the potential for machine learning to create accessible and engaging tools for a broad audience, bridging the gap between technology and the arts.

References

[1] [1] S. Tzanetakis and P. Cook, "Musical Genre Classification of Audio Signals," IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, July 2002.

[2] [2] B. McFee, M. McVicar, S. Balke, C. Thome, and C. Raffel, "librosa: Audio and Music Signal Analysis in Python," [Online]. Available: https://librosa.org, Accessed: December 2024.

[3] [3] M. Pedregosa, G. Varoquaux, A. Gramfort, et al., "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.

[4] [4] Gradio Labs, "Gradio: Build Machine Learning and Data Science Interfaces Easily," [Online]. Available: https://gradio.app, Accessed: December 2024.
[5] [5] A. Olteanu, "GTZAN Dataset: Music Genre Classification," [Online]. Available: https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification, Accessed: December 2024.
[6] [6] F. Chollet and Others, "Keras," [Online]. Available: https://keras.io, Accessed: December 2024.