**Abstract**

The Music Sorter project is an application that utilizes the K-Nearest Neighbors (KNN) algorithm to classify and organize music tracks based on their audio features. The goal of the project is to provide an efficient and automated system for sorting a large music collection into distinct genres or categories.

The KNN algorithm is a supervised machine learning technique used for classification tasks. In this project, the algorithm is trained on a labeled dataset of music tracks, where each track is represented by a set of audio features such as tempo, pitch, rhythm, and timbre. These features are extracted using signal processing techniques.

When a new music track needs to be sorted, the KNN algorithm compares its audio features to those in the database and identifies the k nearest neighbors based on similarity measures. The majority genre/category among the k nearest neighbors is assigned to the new track.

To evaluate the performance of the Music Sorter, a comprehensive dataset of music tracks with diverse genres is used for training and testing. The accuracy of the classification is measured using metrics such as precision, recall, and F1-score.

The results indicate that the KNN-based Music Sorter achieves a high level of accuracy in classifying music tracks into appropriate genres or categories. It demonstrates the potential of the KNN algorithm in organizing and sorting music collections efficiently, saving time and effort for music enthusiasts and professionals.

The Music Sorter project can be extended to incorporate additional features or integrate with existing music platforms to enhance music recommendation systems and provide personalized music experiences to users.

**Methodology**

Dataset Collection: Gather a diverse dataset of music tracks with labeled genres or categories. Ensure that the dataset is representative of various music genres and contains a sufficient number of tracks for each genre.

1. Feature Extraction: Extract relevant audio features from each music track in the dataset. Commonly used features include tempo, pitch, rhythm, timbre, and spectral characteristics. Signal processing techniques such as Fast Fourier Transform (FFT) and Mel-Frequency Cepstral Coefficients (MFCC) can be employed for feature extraction.
2. Data Preprocessing: Normalize the extracted features to eliminate any biases and scale them to a consistent range. This step helps in removing any variations due to differences in track duration, volume levels, or audio quality.
3. Splitting the Dataset: Divide the dataset into training and testing sets. The training set will be used to train the KNN algorithm, while the testing set will be used to evaluate its performance.
4. Training the KNN Algorithm: Implement the KNN algorithm using a suitable programming language or machine learning framework. Provide the training set to the algorithm and tune the hyperparameters such as the number of neighbors (k) and the distance metric used for similarity calculation. The algorithm will learn the relationships between the audio features and their corresponding genres/categories.
5. Classification and Sorting: When a new music track needs to be sorted, extract its audio features and preprocess them. Then, use the trained KNN algorithm to compare these features with the features in the training set. Determine the k nearest neighbors based on similarity measures (e.g., Euclidean distance or cosine similarity). Assign themajority genre/category among the k nearest neighbors to the new track.
6. Performance Evaluation: Assess the performance of the Music Sorter by comparing the predicted genres/categories with the actual genres/categories of the testing set. Calculate evaluation metrics such as precision, recall, and F1-score to measure the accuracy of the classification.
7. Iterative Improvement: Analyze the results and fine-tune the parameters of the KNN algorithm if necessary. Consider experimenting with different distance metrics or feature selection techniques to improve the performance of the Music Sorter.
8. Documentation and Deployment: Document the implementation details, including the KNN algorithm, preprocessing steps, and evaluation results. Prepare the project for deployment, ensuring that it can handle new tracks for sorting in real-time efficiently.
9. Future Enhancements: Explore opportunities for extending the Music Sorter project. This can include integrating with existing music platforms, incorporating additional features like lyrics analysis or mood detection, or enhancing the recommendation system to provide personalized music suggestions based on user preferences.