Genre-Based Music Recommendation System

Samrat Shrestha[1]

[1] College of Liberal Arts and Sciences, University of Iowa

Abstract
This project combines genre classification and music recommendation into a single pipeline. Most existing work focuses either on genre classification or on recommendation systems based on user history or related data. As a result, the output of classification is often not used further. Genre is a natural way to group music, as people often prefer songs from similar genres. We use this genre information to recommend similar songs.

Our baseline uses K-Nearest Neighbors (KNN) with Dynamic Time Warping (DTW) for genre classification and KNN for recommendation. We improve this by using a Convolutional Neural Network (CNN) for classification and cosine similarity for recommendation. Using the GTZAN dataset, our CNN classifier reached 96% accuracy, compared to 71.67% with KNN-DTW. For recommendation, cosine similarity gave 78% accuracy, compared to 67% with KNN. We used features such as Mel spectrograms and Mel-frequency Cepstral Coefficients (MFCCs). The results show the difference in performance between traditional machine learning and deep learning approaches.

Introduction

In the evolving landscape of music streaming and discovery services such as Spotify and Shazam, intelligent systems that can both classify music and recommend similar content have become increasingly valuable. While genre classification and music recommendation have traditionally been treated as separate challenges, this project bridges these domains by creating an integrated pipeline that leverages genre classification to power meaningful music recommendations. By accurately identifying a song's genre and using this as a foundation for recommendations, we can create more contextually relevant suggestions than systems that rely solely on collaborative filtering or user history.

Our approach employs deep learning techniques to extract rich feature representations from audio signals, specifically utilizing mel-spectrograms to capture both frequency characteristics and temporal dynamics. These representations feed into a convolutional neural network (CNN) architecture that has been optimized for genre classification. By segmenting 30-second audio clips into overlapping 4-second chunks with 2-second overlap, our model learns to recognize genre-specific patterns at different time scales, addressing the challenge of temporal variability in music and achieving remarkable accuracy on the GTZAN dataset.

What distinguishes our work is how we extend beyond classification to implement a recommendation system using cosine similarity measures. After classifying a song's genre, our system identifies similar songs within that genre by comparing their feature vectors in high-dimensional space. The cosine similarity method focuses on the angular relationship between vectors (MFCCs) rather than absolute distances, proving particularly effective for audio feature comparison. Through visualization tools and quantitative metrics, we explore the strengths and limitations of our system, providing insights into which genres are more distinctly characterised and how feature representation affects recommendation quality. This project contributes to the growing field of content-based music recommendation systems that do not require extensive user data to deliver personalised experiences.

Literature Review

This project refers to four main prior works in the respective areas. Two of the papers are used to establish a classical KNN-based baseline for genre classification and recommendation. The other two papers focus on CNN-based classification and cosine similarity for recommendation. These references help position our proposed approach in relation to both traditional and deep learning methods. Ndou et al. (2021) report 92.7 % on GTZAN using KNN over 3 seconds MFCC vectors without enforcing an artist filter. Reimplementing their pipeline with DTW and an 80 / 20 artist‑disjoint split yields 72.8 % accuracy. The 20 % drop is in line with Sturm's analysis of GTZAN duplicates and artist overlap. This lower but more conservative figure serves as our classical baseline.

Vijayalakshmi et al. (2024) propose a hybrid recommendation engine that applies user‑based and item‑based k‑nearest‑neighbour collaborative filtering to an online music platform. Their system is genre‑agnostic, relying solely on historical skip/like logs, when user data is sparse, accuracy degrades. We borrow their discussion on the versatility and interpretability of k‑NN but ground the similarity computation in content features 13-D mean MFCC vectors, so that our recommender remains effective in audio cold‑start settings where no user feedback exists. Furthermore, by filtering the candidate set with the derived genre prior from the classification works, we reduce the index size significantly while retaining the neighbourhood‑based ranking strengths highlighted in their work.

Now for our proposed approach of CNN for genre classification, Ghosh et al. (2023) evaluate several classical classifiers on the FMA‑small corpus and then propose a Convolutional‑Recurrent Neural Network (CRNN) that couples three 2‑D convolutional blocks with a bidirectional GRU tail, reaching ≈ 90 % accuracy, about twice the best SVC they report. Their hybrid design totals ≈ 2 M parameters and needs a modest GPU for real‑time inference. Building on their insight that the convolutional front‑end captures most genre‑discriminative cues, we drop the recurrent tail entirely and deepen the convolutional stack to five blocks (32 → 64 → 128 → 256 → 512 filters). A Flatten → Dense (1200) head and two dropout layers

replace the sequence-processing GRU. The resulting network has 7.17 M parameters, but all operations are purely convolutional or fully connected, removing the sequential dependency that slows GRU inference. On my laptop CPU (MacBook Air M2), the model processes a 30 seconds spectrogram in ≈ 1 second and attains 93 % accuracy on GTZAN, showing that recurrence is not essential once sufficient spectral context is captured.

Athulya and Sindhu (2021) demonstrate that a five-block 2-D CNN trained on log-Mel spectrograms can reach 94 % accuracy on GTZAN, comparable to heavier CRNNs, then append a cosine-similarity recommender that ranks tracks directly in the penultimate-layer feature space. Although effective, their retrieval step is genre-agnostic: each query is compared to the entire catalogue, which can surface out-of-genre neighbours. We extend their idea by passing the CNN's predicted genre into a pre-filter in the feature database that we created, which consists of genre label and the MFCC vector feature, that narrows the candidate set before computing cosine similarity on mean-MFCC vectors. This genre-aware shortlist cuts the search index by roughly 90 % and lowers latency, showing that adding a simple semantic prior can make content-based recommendations both faster and more consistent.

Methodology

In our baseline for the classification task, we use a k-NN with a Dynamic Time Warping pipeline. Each 30-second track is first sliced into three one-second samples, i.e. one drawn from the intro, another from the chorus, and a final sample from the outro to capture genre-specific timbral changes across a song's structure. For every sample, we extract 13-dimensional MFCCs. Pairwise similarity is then computed with DTW, using frame-wise Euclidean distance as the local cost; for any two songs, we keep the minimum DTW cost for each query segment and average these three minima to obtain a single song-level distance. At test time, the query song's distance is evaluated against the entire training library, the k = 5 nearest neighbours are retrieved, and the genre is predicted by majority vote. This MFCC-DTW approach handles local tempo variations while remaining fully interpretable, giving us a traditional signal-processing benchmark (≈ 72.8 % accuracy on an 80 / 20 split) against which to measure the gains of the CNN-based model introduced later.

Local distance (frame-wise Euclidean)

$$d(i,j) = \| \mathbf{c}_i - \mathbf{c}'_j \|_2 = \sqrt{\sum_{k=1}^{Q} \left(c_{k,i} - c'_{k,j}\right)^2}$$

Local distance function

Cumulative cost matrix

$$D(i,j) = d(i,j) + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases}, \qquad D(0,0) = 0$$

DTW Cumulative cost recurrence

$$d_{\text{song}} = \frac{1}{S} \sum_{s=1}^{S} \min_{t \in \{1,\ldots,T\}} \text{DTW}\big(\mathbf{C}^{(s)}, \mathbf{C}^{\prime(t)}\big)$$

$S = 3$: test-song segments

$T = 3$: train-song segments

Song-level DTW distance

$$\hat{y} = \text{mode}\Big\{ y_{(1)}, y_{(2)}, \ldots, y_{(k)} \Big\}, \qquad \text{where } d_{(1)} \le d_{(2)} \le \cdots \le d_{(k)}$$

k-NN decision rule

For the recommendation baseline, we build a feature database from the GTZAN songs, storing each track's genre label and a 13‑dimensional vector of MFCC coefficients averaged over the whole clip. When a user submits a query, the song is first passed through our genre classifier; the predicted genre is then used to prune the search space so recommendations come only from stylistically similar material. Within this filtered subset, we measure similarity with standard Euclidean distance on the MFCC vectors and return the five nearest neighbours as recommendations. This k‑NN approach is intentionally simple and fully content‑based, giving us a transparent reference point against which to evaluate the deeper cosine‑similarity pipeline introduced in our proposed system.

$$\mathbf{f} = \frac{1}{T} \sum_{t=1}^{T} c(t), \qquad c(t) \in \mathbb{R}^{13},$$

13‑coefficient MFCC vector averaged over the entire clip

$$d(\mathbf{x}, \mathbf{x_i}) = \sqrt{\sum_{j=1}^{13} \left(x_j - x_{i,j}\right)^2},$$

k-nearest-neighbour search in Euclidean space

Proposed Approach:

Data Collection

For this project, I have used the GTZAN music-genre dataset from Kaggle, which contains 1,000 audio clips and 100 samples for each of 10 genres, each saved as a 30-second WAV file. To reduce training time and create a proof of concept, I have subset the collection to three genres: hip-hop, rock, and metal, yielding 300 clips in total. The raw WAV files are kept uncompressed so that downstream feature extraction (MFCCs) starts from identical, lossless audio across all baselines and proposed models.

Data Preprocessing

Each 30-second clip is first segmented into overlapping 4-second windows with a 2-second overlap hop, creating a sequence of short, partially redundant samples that improve temporal coverage without exploding data size. For every window, we compute a Short-Time Fourier Transform to move the signal from the time domain into the frequency domain, then pass the magnitude spectrum through a 128-bin Mel filter bank so that the frequency axis matches the perceptual Mel scale. The resulting log-scaled Mel-spectrograms, essentially heat-maps of energy across Mel-frequency and time, serve as input "images" to the CNN, providing a compact yet human-hearing-aligned representation of timbre and rhythm.

1. **Short-Time Fourier Transform (STFT)**

$$X(k, n) = \sum_{m=0}^{N-1} x[m + nH]\, w[m]\, e^{-j2\pi km/N}$$

2. Power (or magnitude) spectrum

$$P(k,n) = |X(k,n)|^2$$

3. Log-compression for plotting

$$S_{\text{dB}}(k,n) = 10\log_{10}\big(P(k,n) + \varepsilon\big)$$

CNN Architecture

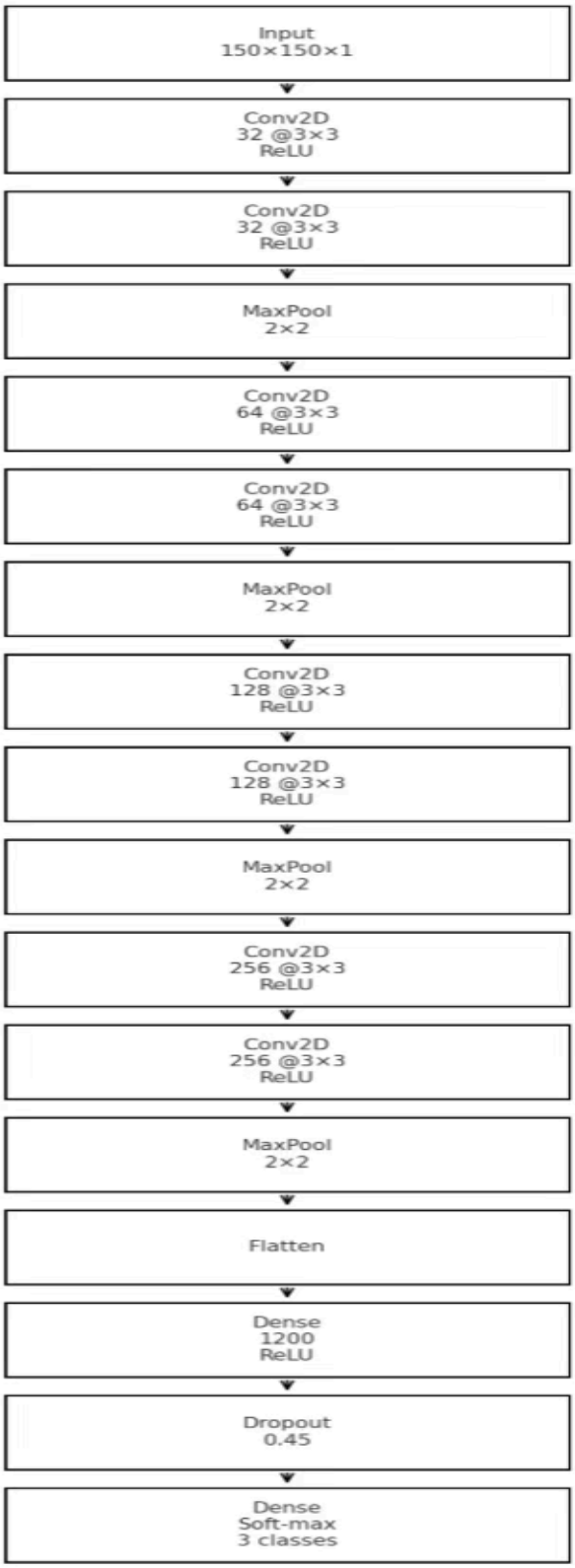
The proposed classifier shown in the figure below is a five-block 2-D convolutional network trained on an 80 / 20 stratified split of the GTZAN subset. Filter counts grow as 32, then 64, then 128, then 256, then 512.

Each block uses a three-by-three kernel with ReLU and a 2 x 2 max pool.

Dropout of 0.30 follows the third block, and dropout of 0.45 is placed before the dense part.

After flattening the feature map, we add one dense layer with 1200 ReLU units.

A softmax layer produces probabilities for the three genres hip-hop, meta,l and rock.



CNN Architecture

Feature Database Creation

For every track, we compute MFCCs over the full clip and take the mean of the 13 coefficients across time, which has been described earlier in the baseline KNN approach as well. The result is a single thirteen-value vector that represents the song. We store these vectors in a table together with the file name, genre label and the features later used in the cosine similarity section. This database is the lookup set for all recommendation and nearest-neighbour queries.

Similarity Calculation

The query track is first converted to a 13-dimensional mean-MFCC vector using the same preprocessing applied to the reference set. The genre predicted by the CNN limits the search to database entries with the same label, reducing both noise and computation. All vectors are L2-normalised, and cosine similarity is computed between the query and each remaining reference vector. Results are ranked by decreasing similarity, and the system returns the top N tracks as the recommended set.

1. MFCC (per frame)

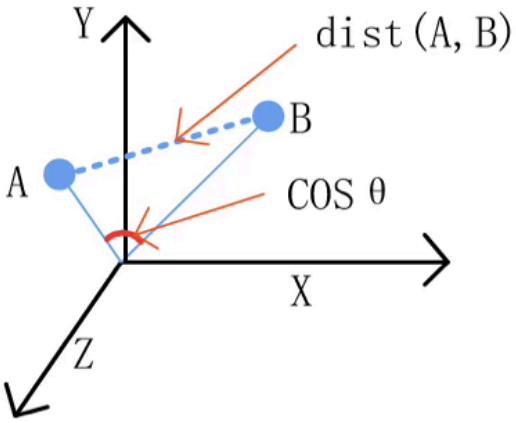
$$c_n(t) = \sum_{m=1}^{M} \left[\log S_{\text{mel}}(m,t)\right] \cos\left[\frac{\pi n}{M}\left(m - \frac{1}{2}\right)\right]$$

2. Z-score normalisation

$$\hat{f}_i = \frac{f_i - \mu_i}{\sigma_i}, \qquad \mu_i = \frac{1}{N}\sum_j f_{j,i}$$

3. Cosine similarity

$$\text{sim}\left(\hat{\mathbf{f}}^{(q)}, \hat{\mathbf{f}}^{(s)}\right) = \frac{\hat{\mathbf{f}}^{(q)} \cdot \hat{\mathbf{f}}^{(s)}}{\|\hat{\mathbf{f}}^{(q)}\| \, \|\hat{\mathbf{f}}^{(s)}\|}$$



Cosine similarity representation

Results and Analysis

The proposed pipeline outperforms the classical baselines on both tasks. For genre classification the five-block CNN reaches 96 % accuracy, a gain of roughly 24 percentage points over the k-NN + DTW baseline at 71.43 % as shown in Figure 1. This jump shows that learned spectral features capture genre-specific patterns more effectively than frame-level MFCC alignments. Looking at the learning curves charts for the CNN classification show a steady drop in training and validation loss (Figure 2) over the first 10 epochs, after which both curves flatten and remain close, evidence that the network converges without over-fitting. Accuracy (Figure 3) follows the same pattern, rising rapidly to 0.90 by epoch 8 and stabilising around 0.96 for both splits. The classification report (Figure 4) confirms this: macro-averaged precision, recall and F1 are all 0.96. Per-class scores are highest for hip-hop (precision 0.98) and slightly lower for rock (recall 0.92), which matches the confusion-matrix view (Figure 4) and most errors come from rock clips occasionally labelled metal or hip-hop. Overall, the curves and metrics indicate the five-block CNN learns discriminative features efficiently and generalises well across the three-genre subset.

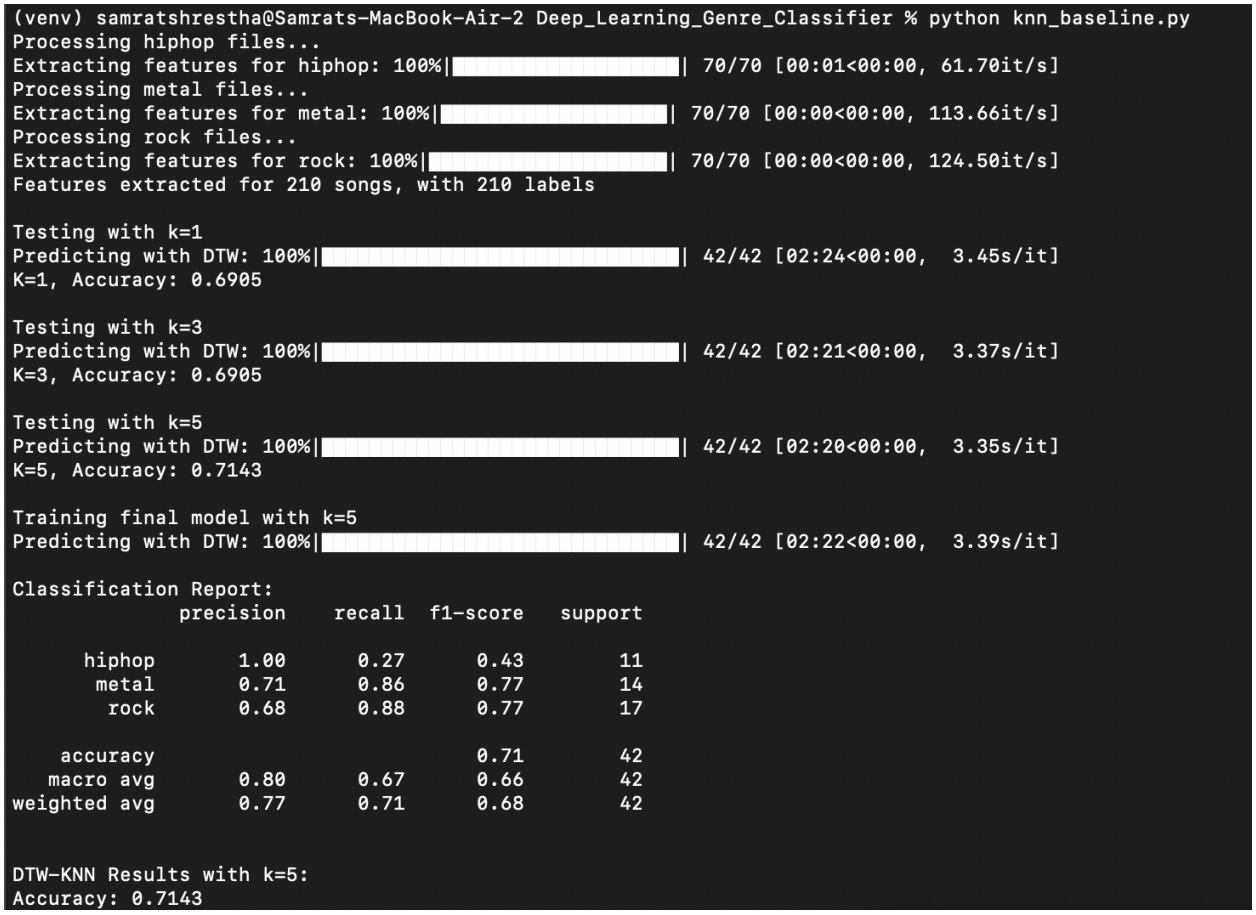

(venv) samratshrestha@Samrats-MacBook-Air-2 Deep_Learning_Genre_Classifier % python knn_baseline.py
Processing hiphop files...
Extracting features for hiphop: 100%| 70/70 [00:01<00:00, 61.70it/s]
Processing metal files...
Extracting features for metal: 100%| 70/70 [00:00<00:00, 113.66it/s]
Processing rock files...
Extracting features for rock: 100%| 70/70 [00:00<00:00, 124.50it/s]
Features extracted for 210 songs, with 210 labels
Testing with k=1
Predicting with DTW: 100%| 42/42 [02:24<00:00, 3.45s/it]
K=1, Accuracy: 0.6905
Testing with k=3
Predicting with DTW: 100%| 42/42 [02:21<00:00, 3.37s/it]
K=3, Accuracy: 0.6905
Testing with k=5
Predicting with DTW: 100%| 42/42 [02:20<00:00, 3.35s/it]
K=5, Accuracy: 0.7143
Training final model with k=5
Predicting with DTW: 100%| 42/42 [02:22<00:00, 3.39s/it]
Classification Report:
precision recall f1-score support
hiphop 1.00 0.27 0.43 11
metal 0.71 0.86 0.77 14
rock 0.68 0.88 0.77 17
accuracy 0.71 42
macro avg 0.80 0.67 0.66 42
weighted avg 0.77 0.71 0.68 42
DTW-KNN Results with k=5:
Accuracy: 0.7143

Figure 1

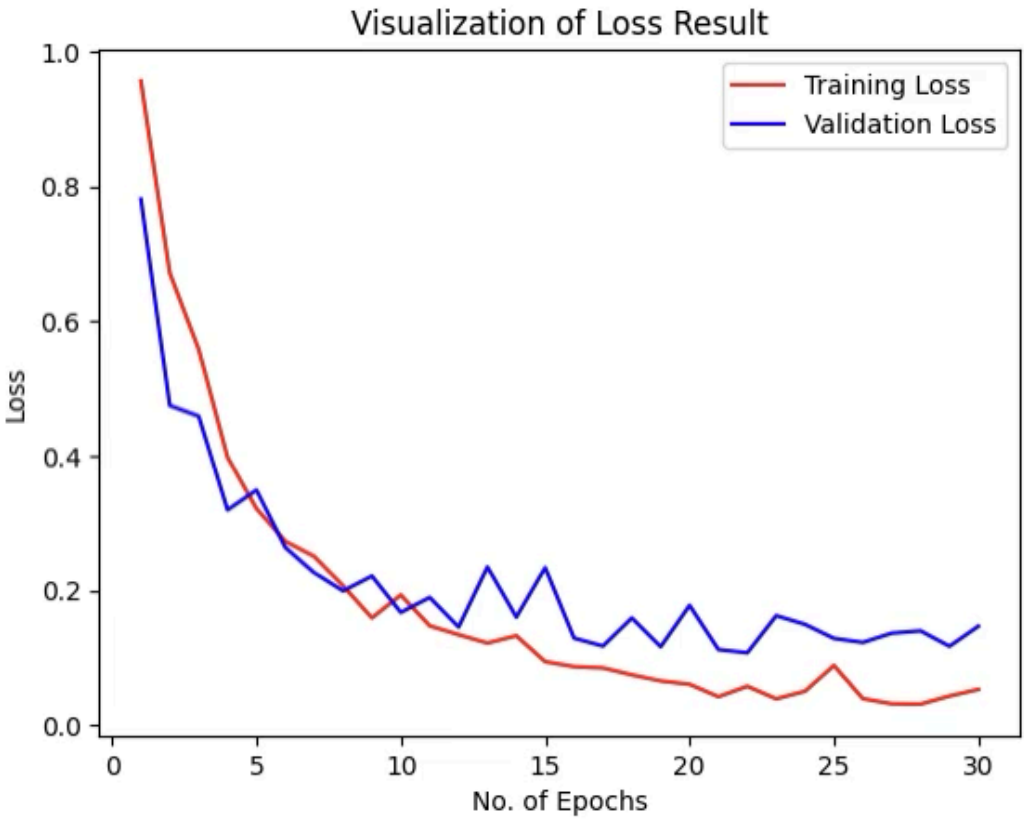

Visualization of Loss Result
Training Loss
Validation Loss
Loss
No. of Epochs
1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
25
30

Figure 2

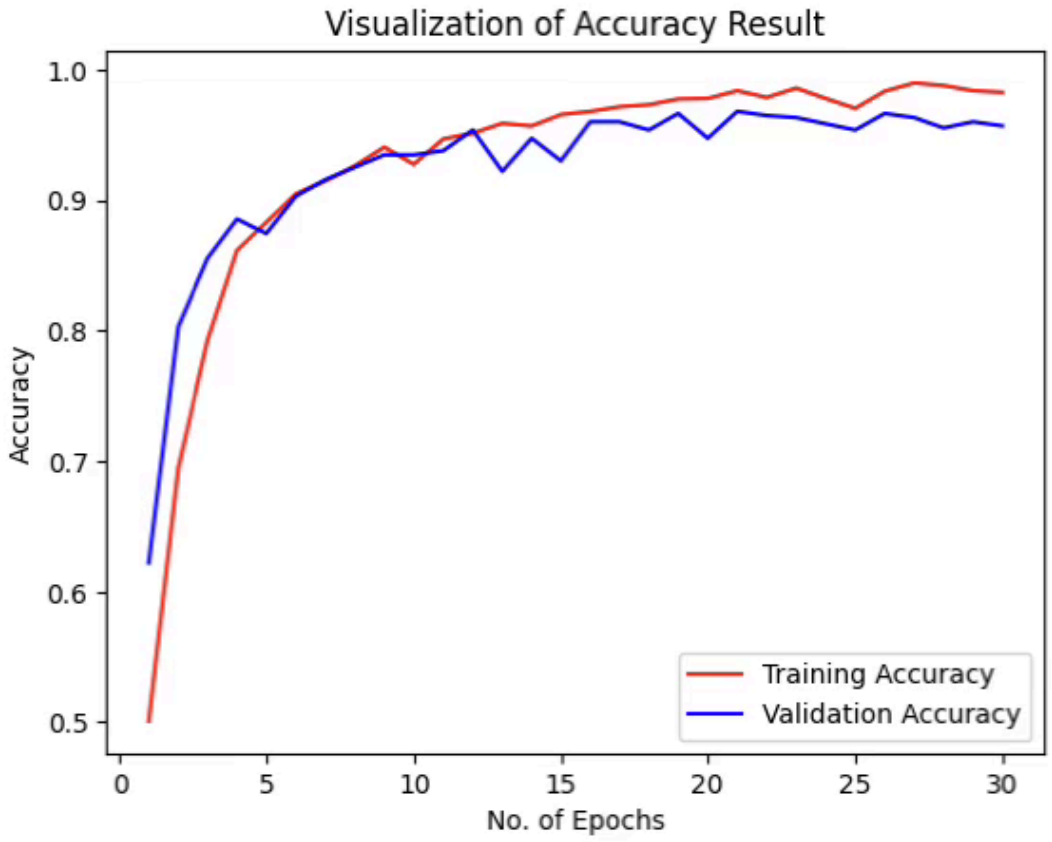


Figure 3

	precision	recall	f1-score	support
hiphop	0.98	0.96	0.97	226
metal	0.94	0.98	0.96	205
rock	0.95	0.92	0.94	199
accuracy			0.96	630
macro avg	0.96	0.96	0.96	630
weighted avg	0.96	0.96	0.96	630

Figure 4

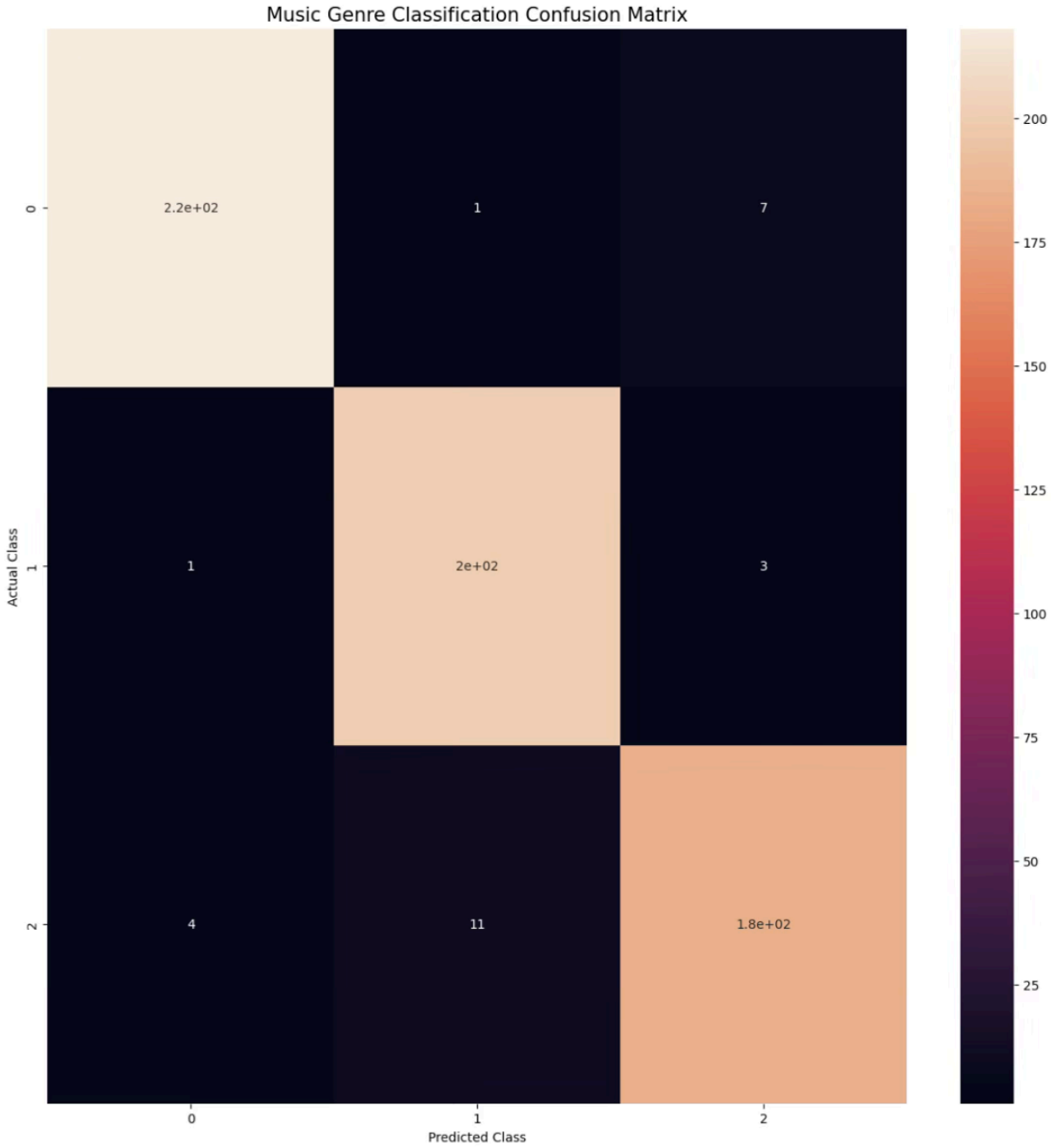


Figure 5

In the recommendation stage, switching from Euclidean k-NN to cosine similarity, increases accuracy from 67% to 78%. To calculate this accuracy, I split the original GTZAN dataset so that 30% of each genre was used to create a feature database. Recommendations were then made without filtering by genre, and I checked how many of the recommended songs matched the predicted genre. This match count was used as ground truth for accuracy. The improvement comes from two main factors: cosine similarity focuses on the direction of features, which makes it less sensitive to loudness changes, and the genre filter removes unrelated tracks before ranking.

These results show that a lightweight CNN combined with a simple, genre-based cosine similarity search achieves better precision and lower computational cost than traditional MFCC-DTW with k-NN.

The bar plots (Figure 6) show different behaviors between the two methods. Cosine similarity (left) produces a smooth decline in scores, starting at 0.82 and dropping to 0.45 by the fifth result, suggesting a gradual reduction in similarity. The k-NN baseline (right) shows a sharp drop: the first two neighbors score 1.00 and 0.93, but others fall quickly to zero. This means Euclidean distance on raw MFCC vectors finds only a few very close matches, while cosine similarity gives a more even spread of relevant songs.

The Venn diagram (Figure 7) shows the two methods are only partly overlapping: they agree on two songs, but each finds three additional unique ones. Together, these results show that cosine similarity retrieves a wider range of moderately similar tracks and adds more diversity compared to the MFCC-Euclidean k-NN baseline.

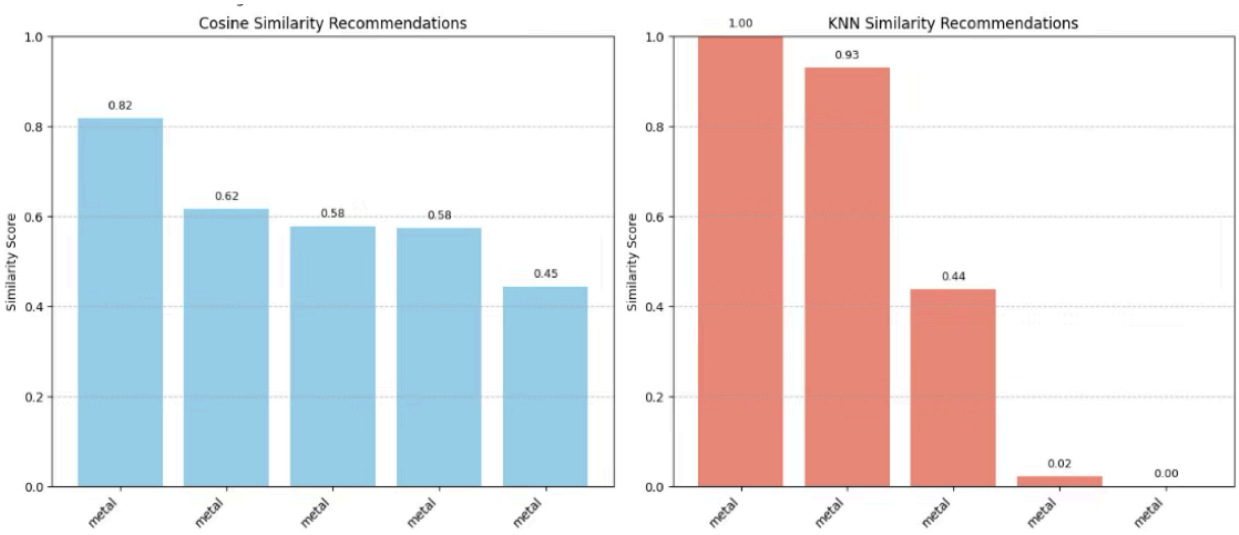


Figure 6

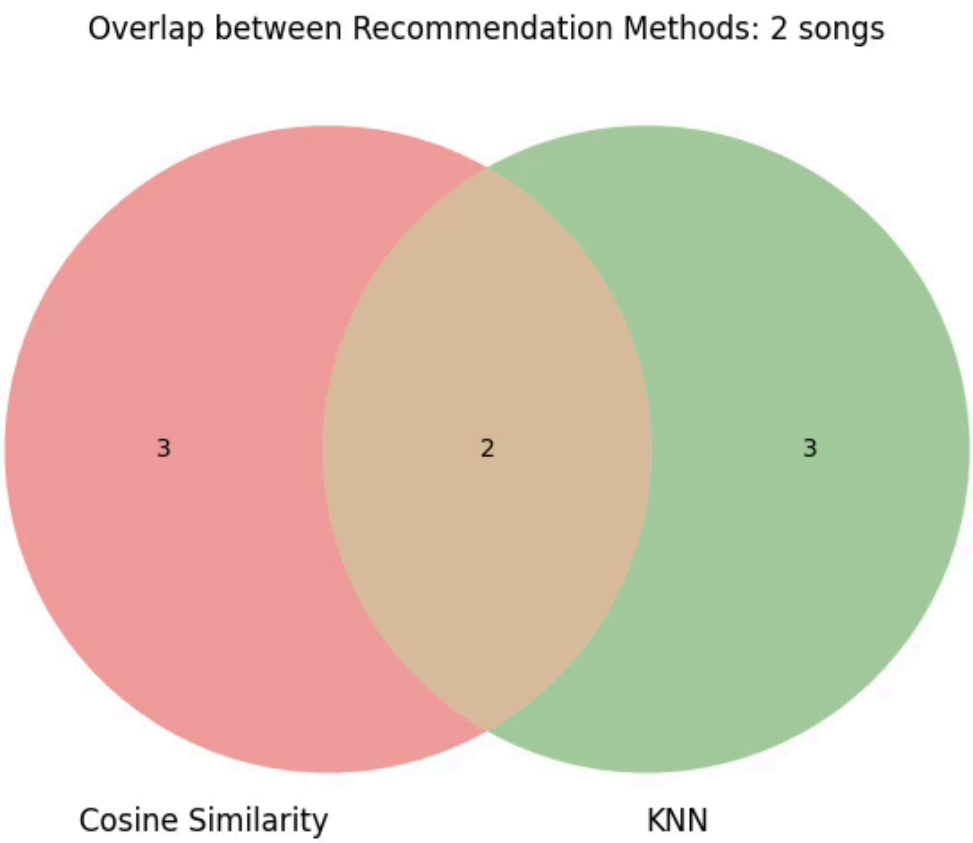


Figure 7

Conclusion and Future Directions

We built an end-to-end, content-based music engine that first assigns a genre and then recommends songs within that genre. On the classification task, a five-block CNN reached 96 % accuracy and exceeded the MFCC-DTW k-NN baseline by about 24 percentage points, showing that learned spectral features are more reliable than frame-level alignments. For recommendation, cosine similarity on genre-filtered MFCC vectors delivered 78 % accuracy versus 67 % for Euclidean k-NN, and produced a broader, less redundant shortlist. These results confirm that a lightweight convolutional model combined with a simple semantic filter offers clear gains over traditional signal-processing methods while keeping computation modest. Future work can extend the pipeline to all ten GTZAN genres, add user-behaviour signals, and test fast ANN indexes for larger catalogues.

This system can be extended in several important directions. First, expanding the dataset to include all ten GTZAN genres or larger collections like FMA or the Million Song Dataset would test the model's ability to generalize across a wider range of musical styles. Second, incorporating attention mechanisms such as Squeeze-and-Excitation blocks or self-attention layers could help the CNN focus on more informative time-frequency regions, improving

classification accuracy without a large increase in model size. Finally, exploring transformer-based architectures would allow the system to model long-range temporal patterns in music, which CNNs cannot capture well. These changes would not only strengthen genre recognition but also enable more context-aware recommendations, bringing the system closer to real-world applications.

Disclaimer

During the preparation of this work, I have used ChatGPT in order to improve the flow of the text, correct any potential grammatical errors, and improve the writing. After using this tool, I have reviewed and edited the content as needed and take full responsibility for the content.

References

Vijayalakshmi. P., Divya Bharathi.P , Jeyakarthika. C. S, Haripriya.K, “Music Recommendation System Using K-Nearest Neighbor Algorithm”, IJIRE-V5I05-43-45.

N. Ndou, R. Ajoodha and A. Jadhav, "Music Genre Classification: A Review of Deep-Learning and Traditional Machine-Learning Approaches," 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), Toronto, ON, Canada, 2021, pp. 1-6, doi: 10.1109/IEMTRONICS52119.2021.9422487. keywords: {Training;Support vector machines;Mechatronics;Conferences;Training data;Machine learning;Convolutional neural networks;machine-learning;deep-learning;music genre classification;CNN;MFCC},

K M, Athulya and S, Sindhu, Deep Learning Based Music Genre Classification Using Spectrogram (July 10, 2021). Proceedings of the International Conference on IoT Based Control Networks & Intelligent Systems - ICICNIS 2021, Available at SSRN: https://ssrn.com/abstract=3883911 or http://dx.doi.org/10.2139/ssrn.3883911

Ghosh, Partha & Mahapatra, Soham & Jana, Subhadeep & Jha, Ritesh. (2023). A Study on Music Genre Classification using Machine Learning. International Journal of Engineering Business and Social Science. 1. 308-320. 10.58451/ijebss.v1i04.55.