Simple Music Genre Classification Using MFCCs Means and Variances

CS 332 Section 001

Machine Learning

12/11/2024

By Sam Edwards

900250707

Abstract

This project showcases a potential method for simple music classification by using the means and variances of Mel-Frequency Cepstral Coefficients (MFCCs) as features for machine learning algorithms/models. This method hopes to reduce computational run time while classifying 4 simple genres of music. This project found that Random Forest Classifiers and Long Short-Term Memory Recurrent Neural Networks had the best accuracy, with roughly 74% and 75% respectively.

1 TABLE OF CONTENTS

2 LIST OF FIGURES AND TABLES

2.1 LIST OF FIGURES

2.2 LIST OF TABLES

3 INTRODUCTION

3.1 INTRODUCTION

Music genre classification is an important application of machine learning in audio processing. By assigning genres to audio tracks, it facilitates tasks such as music recommendation, organization, and retrieval. This project focuses on a simplified approach to classification, leveraging the widely used Mel-Frequency Cepstral Coefficients (MFCCs) to represent the spectral and timbral characteristics of music.

3.2 PURPOSE

The purpose of this report is to explore the effectiveness of using simple statistical features, specifically the means and variances of MFCCs, for classifying music genres. By analyzing the performance of this approach, the project aims to determine whether it can provide reliable results while minimizing computational complexity.

3.3 SIGNIFICANCE

This project highlights the utility of simple feature-based methods for music genre classification, particularly in scenarios where computational resources are constrained. It may serve as a foundation for further exploration, offering insights into how basic features like MFCC statistics can deliver robust classification performance, paving the way for practical applications in resource-limited environments.

3.4 Overview

This report begins with an exploration of music genre classification using MFCC means and variances as features, followed by a detailed description of the project design, data preprocessing, and testing procedures. It presents the results of classification experiments, comparing model performances and analyzing statistical significance. The report concludes with a discussion of key findings, limitations, and recommendations for future research in this domain.

4 Background and Related Work

4.1 Context and Scope

This project is situated in the field of audio processing and machine learning, focusing on simple music genre classification. The scope of this study narrows to a simplified approach that uses the means and variances of Mel-Frequency Cepstral Coefficients (MFCCs) to classify four genres.

4.2 Definitions and Key Terminology

- Mel-Frequency Cepstral Coefficients (MFCCs): Numerical representations of the short-term power spectrum of an audio signal, designed to mimic how humans perceive sound. Used for analyzing timbre or tone quality. Each MFCC can be shown as a list of numbers.
- Spectral Features: Characteristics of an audio signal that describe its frequency content, which help capture the timbral and harmonic qualities of sound. For example, MFCCs.
- Spectrogram: A visual representation of the spectrum of frequencies in an audio signal as they vary with time.
- Random Forest Classifier: An ensemble machine learning algorithm that combines multiple decision trees to classify data, effective for small datasets and robust to noise.
- LSTM (Long Short-Term Memory): A type of Recurrent Neural Network (RNN) designed to handle sequential data by remembering long-term dependencies, making it suitable for audio data.
- Convolutional Neural Network (CNN): A deep learning model that analyzes image-like data (e.g., spectrograms) by detecting patterns and features, such as frequencies in audio.
- Stratified K-Fold Validation: A cross-validation technique that splits data into subsets while preserving the class distribution, ensuring balanced training and validation sets.

4.3 Theory and Concepts

- Stratified K-Fold Cross-Validation: This validation method ensures that each fold contains approximately the same proportion of classes as the original dataset. It provides a more reliable evaluation of model performance, especially for imbalanced datasets, by maintaining genre distribution consistency across folds.
- Hyperparameter Tuning: The process of optimizing parameters that control the learning process of a model (e.g., learning rate, depth of trees) to improve its performance.

- Normalization: The process of scaling the features (e.g., MFCC values) to a standard range, typically between 0 and 1 or with a mean of 0 and standard deviation of 1. This helps a model focus on patterns without having a single feature outweigh others.

4.4 RELEVANT HISTORY AND DEVELOPMENT

Early work in this area focused on simpler, traditional machine learning methods such as K-Nearest Neighbors (K-NN) and Support Vector Machines (SVM). These methods were initially successful, especially when combined with well-established feature extraction techniques like MFCCs. As research progressed, the shift toward deep learning models occurred, with CNNs and LSTM networks being adopted for their ability to learn complex patterns directly from raw data, such as spectrograms or audio sequences.

4.5 REVIEW OF PRIOR RESEARCH

Several studies have addressed the challenges of music genre classification using different machine learning algorithms and feature extraction techniques. Patil and Nemade (2017) demonstrated that combining MFCCs with K-NN can provide reliable results, even with smaller datasets. They highlighted the importance of MFCCs in representing the timbral qualities of audio and showed that simpler models could achieve competitive performance when optimized correctly [1]. More recently, Gessle and Åkesson (2019) compared CNNs and LSTMs for genre classification, showcasing the ability of CNNs to process spectrograms for pattern recognition and the strength of LSTMs in analyzing sequential data such as audio [2].

4.6 COMPARISON TO EXISTING APPROACHES

This project differs from previous work by focusing on a simplified classification approach using the means and variances of MFCCs. Unlike studies with large amounts of data and computational resources, this project aims to provide a more resource-efficient solution. By reducing the number of genres and using basic statistical features from MFCCs, the approach minimizes the computational demands while still achieving reliable results.

5 APPROACH

5.1 PROJECT DESIGN

In comparison to what I’d call Complex Music Classification, which has 500+ genres, one or more labels per song, rhythm, tempo, instrumentation, harmony, and spectral features, I chose to do what I call simple classification. Simple classification consists of 4 genres, those being Rock, Electronic, Folk, and Classical. Each audio clip has one label, and I only use spectral features to classify each audio clip.

This project used a dataset from hugging face with 19900 samples each with its own music genre label. This data was then prepared and preprocessed (see 5.3) then split into different splits for training, validation, and testing. (see 5.5) The newly processed dataset was then run through a Random Forest Classifier, a CNN, and a LSTM. The Random Forest Classifier and CNN used hold-out validation whilst the LSTM used stratified K-fold validation.

Python was used in this project for simplicity and sklearn was used for the Random Forest Classifier, validation splits, and evaluation. Tensorflow was used for the CNN and LSTM model and pandas were used for dataset manipulation. Additionally, the Librosa library was used for feature extraction, data viewing, and audio playback.

5.2 Algorithms/Models Used

This project utilized three machine learning models for music genre classification: Random Forest Classifier, LSTM (Long Short-Term Memory) Network, and CNN (Convolutional Neural Network).

- Random Forest Classifier: An ensemble method that combines multiple decision trees. It works well with smaller datasets and is less sensitive to noise.
- LSTM: A type of RNN that handles sequential data, capturing long-term dependencies in audio. It requires more data and computation but can offer high accuracy.
- CNN: Analyzes spectrogram-like features (MFCCs) to detect patterns in frequency and time. It excels with larger datasets but struggled with smaller ones in this study.

5.3 Data Preparation and Preprocessing

Data preparation involved extracting 20 MFCCs per audio sample using the Librosa python library and using their mean and variance for classification. Key preprocessing steps included:

- Data Cleaning: Removed irrelevant genres (e.g., Ambient Electronic and International), reducing the dataset from 19,900 samples to 7,761 samples.
- Grouped similar ones (e.g., Folk and Country) to avoid genres with too little samples.
- Normalization: Feature values were normalized to ensure consistency.

These steps ensured clean and standardized data for training the models

5.4 Equipment and Tools

Programing Language: Python

Libraries: numpy, pandas, tqdm, librosa, tenserflow, sklearn, skopt

Software: Google Collab, models trained using T4 GPU

5.5 Testing Procedures

To ensure correct model functionality, several testing techniques were applied. For data splitting, the Random Forest and CNN models used a 70%/15%/15% split for training, validation, and testing, while the LSTM model used an 80%/20% split* for training and testing. Hold-out validation was used for the Random Forest and CNN models during training, while Stratified K-Fold cross-validation with 5 folds was applied to the LSTM model. Afterward, 5-fold stratified cross-validation was used to determine the standard deviation of cross-validation results for all models, providing insights into their consistency and stability. (See 6.2, 7.2, and 7.5 for CNN interpretation)

In terms of hyperparameter tuning, Bayesian optimization was employed to optimize the hyperparameters of both the Random Forest and LSTM models. Performance evaluation metrics included accuracy, loss, precision, recall, F1-score, and standard deviation, ensuring a comprehensive assessment of each model's performance. These procedures ensured that the models were correctly trained, optimized, and evaluated. (See 9.1 for hyperparameters)

*80% training was split again into 80%/20% training/validation during stratified k-fold validation.

5.6 Precautions and Controls for Accuracy

To prevent overfitting, all models had early stopping implementation, stopping when the validation loss of each epoch had stopped improving. Additionally, stratified k-fold was chosen in order to equally break the dataset up when validating. Hyperparameters were tuned on the training dataset. Random seeds were set to ensure reproducibility.

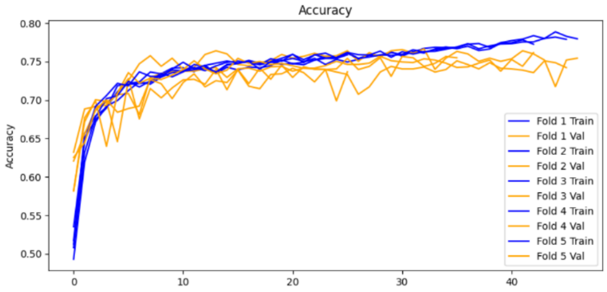
5.7 Reproducibility

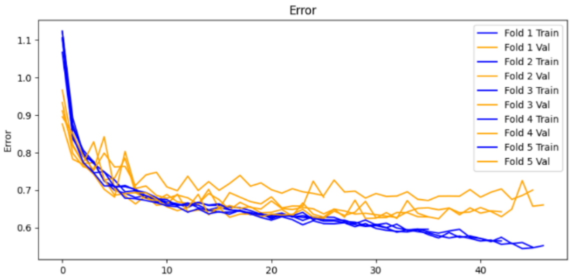
All results can be reproduced by using the same dataset and code. (available in 9.2)

6 Results

6.1 Summary of Findings

The models tested for music genre classification showed similar levels of performance with slight deviations. The Random Forest Classifier achieved an accuracy of approximately 74.8% with a loss of 0.6747 on the test dataset (using 15% for testing in a hold-out validation setup). The CNN model performed slightly lower with an accuracy of 73.2% and a loss of 0.6918. In contrast, the LSTM model demonstrated the highest accuracy of 75.5% with a loss of 0.6300 when using Stratified K-Fold cross-validation (20% test set). As a note, the accuracy and loss values shown in figure 1 and 2 are similar but not equal to performance on the testing set, which performed slightly better. This is discussed more in 6.3.



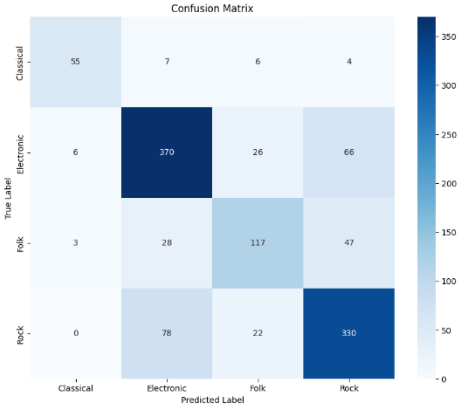


(Figure 1: Comparison of Training and Validation accuracy on the LSTM model over 5 folds.)

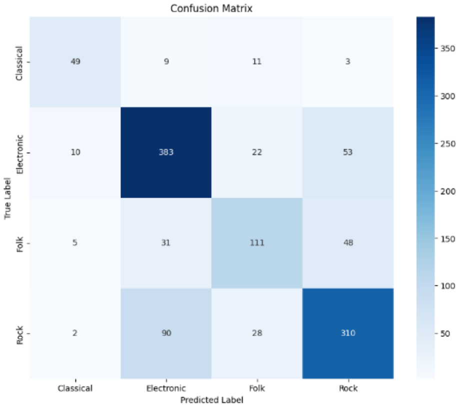
(Figure 2: Comparison of Training and Validation loss on the LSTM model over 5 folds.)

6.2 Calculations and Data Analysis

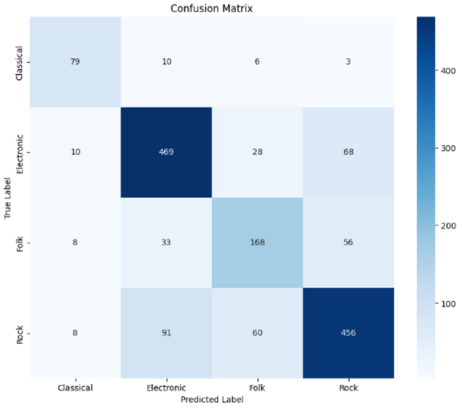
The cross-validation (CV) results for the models showed varying performance in terms of both accuracy and consistency. For the Random Forest model, tested with a 15% test set and using 5-fold cross-validation on the entire dataset, the standard deviation of the CV scores was approximately 0.0085. This low standard deviation indicates that the Random Forest model performed consistently across the different folds, providing reliable results with stable accuracy. The CNN model, also tested with a 15% test set and using 5-fold cross-validation, showed a higher standard deviation of approximately 0.0331, indicating greater variability in performance across the folds. The CNN did not perform well with smaller dataset splits, and its average accuracy was 59.9%, highlighting its struggle with reduced data and its sensitivity to smaller training sets. In contrast, the LSTM model, tested with a 20% test set and using 5-fold cross-validation on 80% of the dataset, achieved the best consistency with a standard deviation of approximately 0.0066. The LSTM demonstrated more stable performance across folds, indicating that it could effectively capture differences in the MFCC means and variances.



(Figure 3: A Confusion Matrix of Labels on the Random Forest Classifier on 15% of the dataset)



(Figure 4: A Confusion Matrix of Labels on the CNN on 15% of the dataset)



(Figure 5: A Confusion Matrix of Labels on the LSTM on 20% of the dataset)

6.3 Error and Uncertainty Analysis

Across five folds, the Random Forest Classifier and LSTM model the accuracy of each fold fluctuated by roughly ±2.5% whilst the CNN fluctuated by ±10%. This is most likely due to incorrectly labeled data or confusing data (see 7.5) within the dataset itself, as stratified k-fold accounts for equal sample sizes. For the CNN the higher inconsistency is caused by the smaller splits of data used for validation, causing the CNN to not have enough samples to accurately train.

Additionally, I found that rerunning any of the models could produce roughly up to ± 0.8% variation in average accuracy. This is mostly likely also because of bad data. (see 7.5)

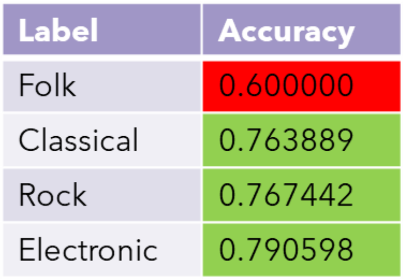
6.4 COMPARISON OF MODELS

The Random Forest model achieved a precision of 0.7479, recall of 0.7485, and an F1-score of 0.7474, indicating a well-balanced performance with consistent accuracy in classifying music genres. The CNN model (using hold-out validation) showed slightly lower results, with a precision of 0.7301, recall of 0.7321, and F1-score of 0.7299, reflecting its struggles with smaller datasets and greater variability in performance. The LSTM model, however, outperformed both the Random Forest and CNN, with a precision of 0.7546, recall of 0.7547, and an F1-score of 0.7542. These results demonstrate that while the Random Forest offers stability, the LSTM provides the best overall classification performance in terms of precision, recall, and F1-score. The accuracies of the Random Forest Classifier and LSTM Model, although different, are not unique enough to classify them as different in terms of reliability. This is because of the aforementioned issues with the dataset. (See 6.3)

	Random Forest (Using stratified k-fold validation)	**CNN (Using stratified k-fold validation)**	**CNN (Using Hold-out validation)**	**LSTM (Using stratified k-fold validation)**
Accuracy	74.8%	59.9%	73.2%	75.5%
Loss	0.6747	0.8542	0.6918	0.6650
Precision	0.7479	Not Tested	0.7301	0.7546
Recall	0.7485	Not Tested	0.7321	0.7547
F1-Score	0.7474	Not Tested	0.7299	0.7542
Standard Deviation of Cross Validation	0.0085	0.0331	NA	0.0066

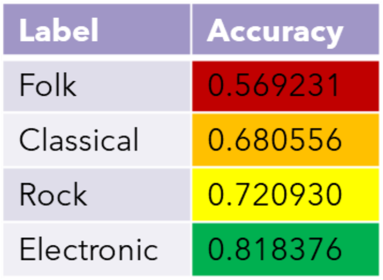
(Table 1: Table of performance metrics)

As shown by table 1, the LSTM model performed slightly better than the Random Forest Classifier in all categories.



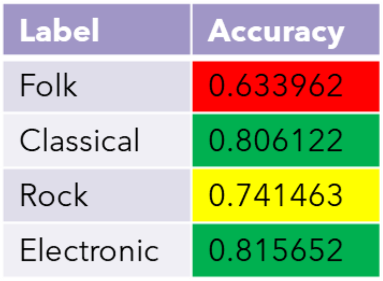
Label	Accuracy
Folk	0.600000
Classical	0.763889
Rock	0.767442
Electronic	0.790598

(Table 2: Random Forest Classifier Label Accuracies on 15% of the dataset)



Label	Accuracy
Folk	0.569231
Classical	0.680556
Rock	0.720930
Electronic	0.818376

(Table 3: CNN with Hold-out Label Accuracies on 15% of the dataset)



Label	Accuracy
Folk	0.633962
Classical	0.806122
Rock	0.741463
Electronic	0.815652

(Table 4: LSTM Label Accuracies on 20% of the dataset)

7 Discussion of Results and Conclusion

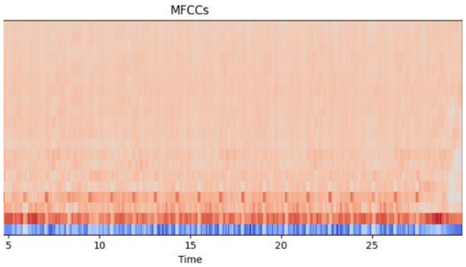
7.1 Key Findings

The Random Forest model was the most stable, providing consistent performance with a balanced precision, recall, and F1-score. However, the LSTM model outperformed all models in terms of classification accuracy, achieving the highest precision, recall, and F1-score. The CNN model, while effective for pattern recognition, struggled classifying MFCC means and variances with smaller datasets, leading to lower accuracy and more variability.

7.2 Interpretation of Results

The Random Forest Classifier performed well with the smaller dataset. Because it's an ensemble model it is able to classify messy audio data that might have a lot of noise. Additionally, MFCCs and the means and variances pulled from them are complex. Random Forest Classifiers can handle complex, multi-dimensional data efficiently by evaluating different combinations of features across decision trees.

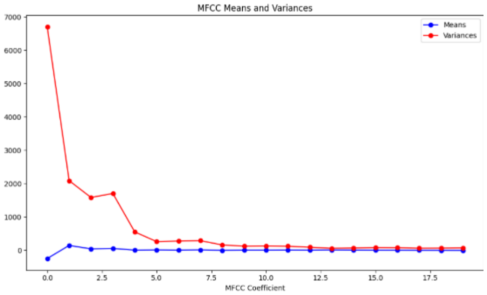
The CNN performed slightly worse than the other models not only because of the dataset, but also due partially in part to a lack of features. A CNN would perform much better on an audio spectrogram or a full list of MFCCs, but for this project I chose to use the means and variances of twenty MFCCs. This is a significant shrink in usable data. Additionally, when testing the model using stratified K-fold cross validation, it no



(Figure 6: An example of MFCCs over a 30 second audio clip.)

longer had enough data to produce accurate results across each fold, highlighting the issue with limited features and data. The CNN model would most likely perform much worse on a smaller dataset.

The LSTM performed well, even though it was working with a smaller dataset. LSTM models for music classification are typically used for much larger datasets, so to see higher accuracy in a smaller dataset is promising. It was able to recognize differences in the genres differently than the Random Forest Classifier, showing the difference in processes. I think that this model would perform better on a larger scale.



(Figure 7: An example of MFCCs Means and Variances over a 30 second audio clip.)

7.3 Comparison with Theory and Prior Research

Although the accuracy of my models were lower than the results of the research done in the past, it should be noted that number of features used in my models were much lower and less complex. For example, the research done by N. M. Patil and M. U. Nemade had upwards of 40 unique features, including the MFCCs means and variances. [1] I found that to extract these features it took much more computational power than just the ones I used. I found success in classifying my dataset when audio that was distinctly one genre was classified.

7.4 Implications of Findings

I think my results imply that it may be possible to classify distinct music genres with less features than previous examples. Additionally, it shows that with smaller datasets, a Random Forest Classifier works reliably and is able to classify music that is distinctly one genre. This type of classification with less features may be useful for tools that don’t need complex music classification.

7.5 Limitations

The main limitation came from the dataset chosen for this project. Originally containing 19900 samples, this dataset seemed promising, but a combination of misclassified audio not discovered until after testing and unbalanced label counts cause the most error on this project. For example, one sample found was labeled rock, but the audio itself was of an interview of a rock song. Additionally, the lack of computational power when extracting features caused some challenges when originally designing this project, leading to the current outcome.

7.6 Improvements and Future Work

I think the most relevant improvement that could be made to this project could be choosing a different dataset with cleaner data. I think this would improve the performance of my simple classification models by 10%-20%, just based off incorrect samples personally reviewed. For future work, a larger dataset would hopefully provide more insight into the performance of the CNN and LSTM models, with more opportunity to perform on more samples. A larger and/or dataset could also help with the inconsistencies of the Folk genre.

7.7 Conclusions

In conclusion, the Random Forest Classifier and LSTM models performed the best with the most reliable accuracies and F1-scores. More work is needed to fully experiment with each model and what the limits of the performance metrics are with a larger dataset. The means and variances of MFCCs worked better as features than originally expected, showcasing the potential behind this type of classification.

8 References

[1] N. M. Patil and M. U. Nemade, “Music Genre Classification Using MFCC and KNN,” 2017. [Online]. Available: https://svv-research-data.s3.ap-south-1.amazonaws.com/220044-Music%20Genre%20Classification%20Using%20MFCC,%20K-NN%20and%20SVM%20Classifier.pdf

[2] G. Gessle and S. Åkesson, A comparative analysis of CNN and LSTM for music genre classification, Dissertation, 2019. [Online]. Available: https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-260138.

9 Appendices

9.1 Hyperparameters

Found using Bayesian optimization using Gaussian Processes.

Random Forest Classifier

- n_estimators: 656
- max_depth: 13
- criterion: entropy

LSTM

- Learning Rate: 0.002452612631133679
- LSTM Layer 1: 99
- LSTM Layer 2: 107
- Batch Size: 64

9.2 Code and Dataset

- Dataset: https://huggingface.co/datasets/lewtun/music_genres
- Code: https://colab.research.google.com/drive/1RH1JAQhrFTZeGtoTGlw-ZVznJ6zlaqPc?usp=sharing