National College of Ireland

Deep Learning Techniques for Music Genre Classification and Building a Music Recommendation System

MSc Research Project
Data Analytics

Jonathan Mendes
Student ID: x18179584

School of Computing
National College of Ireland

Supervisor: Dr Rashmi Gupta

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jonathan Mendes
Student ID:	x18179584
Programme:	Data Analytics
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr Rashmi Gupta
Submission Due Date:	28/09/2020
Project Title:	Deep Learning Techniques for Music Genre Classification and Building a Music Recommendation System
Word Count:	6336
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	28th September 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Deep Learning Techniques for Music Genre Classification and Building a Music Recommendation System

Jonathan Mendes
x18179584

Abstract

Recommendation mechanisms have been increasingly popular in recent years when a large number of people rely on the internet to discover solutions from a wide variety of choices. Due to the competition of the music market, companies are committed to providing personalized music to users in order to attract more buyers. Recommending music that takes into account the music features can enhance the user's listening experience and increase consumer service. In this study, a music recommendation system is built after classifying the tracks according to the genre. The convolutional neural network (CNN), convolutional neural network with long short-term memory (CNN-LSTM), and convolutional neural network with bidirectional long short-term memory (CNN-BiLSTM) models are used for the classification. CNN is considered as the base model. The CNN-LSTM and CNN-BiLSTM models are built on it. The data is taken from the free music archive (FMA) dataset. The content-based (CB) recommendation system with cosine similarity is used as a recommendation model. The features are extracted using a Mel-spectrogram from the audio files. On evaluation, it was observed that the CNN-LSTM model with CB recommendations performed the best. In the future, an ensembled model consisting of all 3 classification models to classify the genre along with a hybrid CB-CF system can be used.

1 Introduction

Digital streaming platforms, such as Soundcloud, Spotify, and YouTube music have drawn the interest of a great number of people, together with digital growth and the sharing of revenue from the music industry, worldwide. This demanding market, in which sustainability is challenging even for top companies, has left music service owners conscious that they need to supply their customers with not only a wide range of audio songs, but also more customized and sophisticated digital offerings on a regular basis. Computer technology fields such as artificial learning, advice application, and data processing are also of considerable significance in the music business. Customization of music services is a key problem in this area. The automated production of customized music playlists and the availability of a process of movement between them, based on certain attributes and measures, would surely contribute to significant achievements in the personalization and knowledge of music services, and even to increasing the satisfaction of their clients

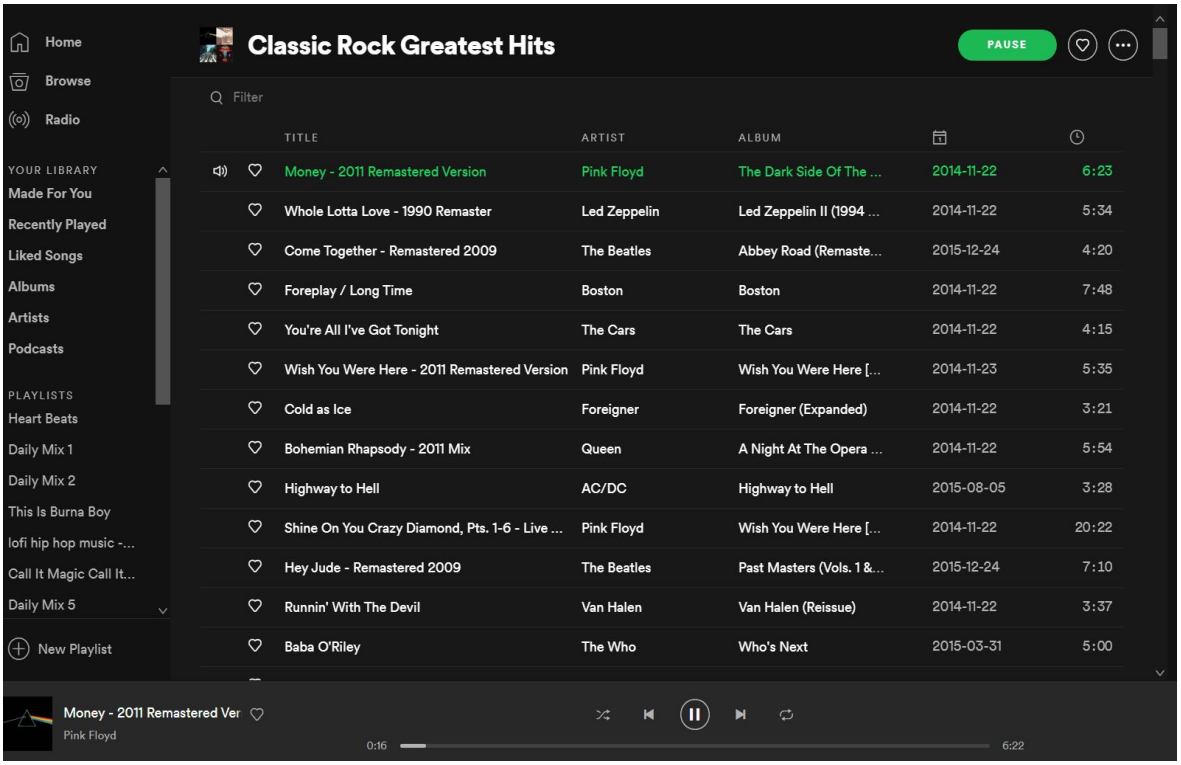

Figure 1: Spotify Music Recommendation

(Bakhshizadeh et al.; 2019). Figure 1 contains the Spotify dashboard through which audio tracks are recommended to their users.

With the broad adoption of the recommendation method, one troublesome and frequent issue is the cold-start crisis, when certain users or files are not detected from previous incidents, such as reviews or clicks. The cold-start challenge can result in the early failure of new items due to the low accuracy of the recommendation. The concern with this is that it would allow the new item to lose the ability to be recommended and remain "cold" (Gupta and Arora; 2016).

In this research neural network models like CNN, CNN-LSTM, and CNN-BiLSTM are used to classify songs based on the genre. **The CNN model as used in (Chang et al.; 2018; Abdul et al.; 2018; Choi et al.; 2016) is tuned and transformed according to the study's objective to function optimally and is considered as the base model. The CNN-LSTM and CNN-BiLSTM models are built on this base model with hyperparameter tuning and the findings are compared with the baseline CNN model**. The FMA dataset is used in this study which contains both audio and metadata. The audio files are in the mp3 format and its features are extracted with the Mel-spectrogram. A CB recommendation system with similarity metrics is used to provide the recommendations. The reason for selecting the above techniques is mentioned in Section 2. The classification models are compared based on the classification accuracy, logarithmic loss, f1 score, precision, and recall. The recommendation system is evaluated based on its genre accuracy. The entire system is evaluated by measuring how well the model recommends songs of the same class for 5 and 10 recommendations.

1.1 Research Questions

- To what extent can Mel-spectrograms be used to extract features of the audio track?
- How well can different neural networks models classify the audio data based on the genre?
- To what level can content-based models be used in recommendation systems?

1.2 Proposed Research Objectives

- Extract the features of the audio tracks in the FMA dataset in the mp3 format.
- Use neural network classification models like the CNN-LSTM and CNN-BiLSTM for music genre classification and compare it with the baseline CNN model
- Use a content based model with a cosine similarity metric to recommend audio tracks and resolve the cold-start problem.

The remainder of the paper is written as follows: In Section 2 all the work done concerning music recommendation systems is discussed. The Table 2 that consists of the important related literature is listed. Section 3 contains the methodology that is implemented in the research. In Section 4 the design specification for the study is explained with the help of a diagram. Section 5 contains the implementation of the models in detail. In Section 6 the evaluation of the models is discussed and compared with the help of graphs and metrics. Section 7 comprises of the conclusion and the future work to be carried out.

2 Related Work

There is a great emphasis on music recommendation systems in the current music business. The fundamental reason it exists is to automate the creation of custom playlists. An improvement to this is the development of a system for switching between playlists depending on the the past song's features, current mood of the consumer and the region and time. Through doing so, an innovative and professional audio application can be created that will further boost the music industry. Below the studies carried out by different researchers on music recommendation systems is discussed and the most relevant parts are used in this study. It is broken down into the datasets, feature extraction, classification models, recommendation models and the summary table.

2.1 Datasets

Although the implementation of novel machine learning models and algorithms to solve complex real-world issues is obviously of primary importance in every area of study, evaluating and comparing it with the current state-of-the-art is important for the technology to be widely accepted by research communities. For this it is critical to include an appropriate and relevant dataset for the study. Table 1 lists the most popular datasets used for studies based on music information retrieval (MIR).

GTZAN, a set of 1,000 audio clips consisting of 10 genres was the first benchmark dataset that was publicly accessible for genre recognition. It still continues to be the

Table 1: Dataset Comparison

Dataset	Clips	Artists	Year	Audio
GTZAN	1,000	approx. 300	2002	yes
CAL10k	10,271	4,597	2010	no
MagnaTagATune	258,633	230	2009	yes
Codaich	26,420	1,941	2006	no
FMA	106,574	16,341	2017	yes
OMRAS2	152,410	6,938	2009	no
MSD	1,000,000	44,745	2011	no
AudioSet	2,084,320	-	2017	no
AcousticBrainz	25,247,395	-	2017	no

most preferred dataset for genre recognition in spite of it various drawbacks such as distortions, mislabeling and repetitions. Also its size is relatively small and also lacks metadata. MagnaTagATune, AudioSet, AcousticBrainz and Million Song Dataset (MSD) are seen as challengers for a large dataset. MagnaTagATune includes audio, features and metadata but its usage is limited due to limited number of tracks and poor audio quality. Although, AudioSet and MSD are large datasets, researchers are forced to use online services to download the clips. The approach AcousticBRainz takes to resolve the copyright issue is to upload the track's music descriptors by asking the community. It will never distribute audio, even though it is the largest dataset.The FMA dataset contains both audio and metadata. It provides the qualities of permissive licensing, large scale, available audio, future proof and reproducible, metadata rich, easily accessible and quality audio (Defferrard et al.; 2016).

2.2 Feature Extraction

In this section several important feature extraction techniques are discussed that were used in previous researches. In audio signal processing the techniques for feature extraction can be classified into various groups. One of them is digital signal processing which is applied on the frequency and time domains. Another used technique is that of statistical descriptors like standard deviation, mean, median, etc. In these methods the raw audio signal is divided into N number of windows and are run M times (Elbir et al.; 2018). Elbir et al. (2018); Al Mamun et al. (2019) have used zero crossing rate, spectral centroid, spectral bandwidth and spectral rolloff as feature extractors. Zero crossing rate is described as the number of times in a signal that its sign changes over a period of time. The sign change is the movement of the signal between its positive and negative values. Spectral centroid is applied on the frequency domain and specifies the center of gravity point of the frequencies present in the frequency bin. Spectral bandwidth displays the difference in the weighted average amplitude between the frequency magnitude and the brightness. Spectral rolloff is described as the normalized frequency at the point where the total of the sound's power values at low frequency arrives at a specific rate in the overall power spectrum. Elbir et al. (2018) also utilize the spectral contrast, spectral rolloff and mel frequency coefficient cepstral coefficients (MFCC). Spectral contrast is the difference in the decibels between the peak and pit points on the signal's spectrum. It gives details about the changes in power in the sound, in audio processing. MFCCs

are a small group of features that explain the entire shape of the spectral pattern.

Han et al. (2018); Dai et al. (2015); Rajanna et al. (2015) use MFCC. Han et al. (2018) builds a recommendation system based on the MFCC feature similarity. The MFCC feature values, quantize the content. It is a method that extracts speech features. The first step for MFCC is intercept fragment where only the 30 seconds of the song is taken. It is followed by pre-emphasis where to compensate for the high-frequency part in which there is a suppression of the speech signal, the resonance peak of high frequency is highlighted, and the high-frequency portion is amplified using the speech signal of the pre-emphasis filter differential. After which, framing and windowing is carried out. Finally, the mel filter bank and discrete cosine transform steps are performed.

Nasrullah and Zhao (2019) has replaced the traditionally used MFCCs which in the audio's short-window frame extract the frequency content, with high dimensional spectrograms. The spectrograms take the benefits of the temporal structure and perform better in classifications. It is described as the frequency content over time obtained by squaring the short-time Fourier Transform's (STFT) magnitude of a signal. The spectrogram here captures the temporal variation and the frequency content for 3 second samples of the audio. Elbir and Aydin (2020) use Mel-spectrogram to extract features from the audio signal and pass it to the CNN model for classification. The audio tracks of 30 seconds are split into 6 parts of 5 seconds each and these samples are converted into Mel-spectrograms.

Chiliguano and Fazekas (2016) execute the Mel-spectrogram spectral analysis feature extraction method. A segment equal to 3 seconds of every clip is loaded at a 22,050 Hz sampling rate and is transformed into a mono channel. For each segment a spectrogram powered by a mel-scale consisting of 128 bands is computed from 1,024 sample windows with a 512 sample hop size. This results in a 130 frame spectrogram with 128 components. Finally, the spectrogram is transformed into a logarithmic scale in decibels taking the peak power as reference. Choi et al. (2016) evaluate the STFT, MFCC and Mel-spectrogram representations before passing it to the model. It is noted that when considering the audio input representations, utilizing Mel-spectrograms performed better as compared to MFCCs and STFTs.

To conclude, based on the performance and since it captures more characteristics of the audio, Mel-spectrograms is used in the models for feature extraction.

2.3 Classification Models

The accurate classification of the music genres is of utmost importance to music recommendations and music information retrieval. Al Mamun et al. (2019) propose machine learning techniques and a deep learning model to classify the genres in Bangle music. The accuracy of 74% was obtained by using the proposed neural network model which was significantly higher than that of the machine learning models used such as linear regression, logistic regression, support vector machine (SVM) and k-nearest neighbours (KNN). It is concluded that neural networks outperforms the traditional machine learning techniques. Karunakaran and Arya (2018) presents a 2 phase hybrid classifier to resolve the issue of blurry classification seen in the classifiers such as SVM, KNN, naive bayes classifier, fuzzy classifiers, neural networks and quadratic discriminant analysis for music genre classification. An accuracy of 90% is achieved when implementing the proposed model which is significantly higher than the other models used.

Choi et al. (2016) use fully convolutional neural networks (FCN) to propose an automated music tagging algorithm that is CB. Various models have been evaluated that consists of 2 dimensional convolutional layers along with sub-sampling layers. A 4-layer FCN architecture was proposed comprising of 4 convolutional layers with 2 max-pooling layers. It was concluded that the 4-layer architecture was outperformed by deeper models with increased layers and the deeper network benefited from large training data. Abdul et al. (2018) incorporate a deep convolutional neural network (DCNN) for the latent feature extraction of the music data. The Mel-spectrogram of the audio track and the metadata is used by the DCNN for the classification into genres. Instead of the sigmoid function, rectified linear units (ReLU) are used for faster convergence. Chang et al. (2018) use the CNN approach to classify the audio tracks into various genres based on the beats of the audio signal. This model uses ReLU with the MaxPool activation function and the audio signals are converted into Mel-spectrograms of 599 frames along with 128 frequency bins.

Elbir and Aydin (2020) use an approach based on the acoustic characteristics of the audio track for genre classification. A novel deep neural network is used to extract the representation features. Artificial dropout features and new layers have been added to reduce the validation error. Each layer compromises of a 2 dimensional convolutional layer, a ReLU activation function, a 2 dimensional maximum pooling layer and a dropout layer. Jiang et al. (2017) utilise a classification technique based on long short-term memory (LSTM) as the audio is sequential in nature.

Tao et al. (2019) propose a modified LSTM model to study the embedding of both the music and the user based on the temporal context and sequential data. The learning rate of the model is updated to increase the accuracy by the Adam optimizer.The authors in (Fulzele et al.; 2018) present a hybrid classification technique using LSTM and SVM which increased the accuracy. It obtained an accuracy of 89% which was significantly higher than their individual accuracies.The models are individually trained and then combined to display the final prediction. Random grid search is used for tuning the hyperparameters to optimize the values.

Sainath et al. (2015) use CLDNN that combines CNNs, LSTMs and deep neural networks (DNNs)h into one unified architecture. Its seen that the CLDNN gives a 4% to 6% rise in the WER as compared to LSTM which is the strongest among the 3 models. Using the DNN layers, converts the LSTM output to a more discriminate space and it is easier for the output targets to be easily predicted. Authors in (Adiyansjah et al.; 2019) adopt a convolutional recurrent neural network (CRNN) for feature extraction.The architectures of CNN and CRNN are compared and it is noted that CRNN performed better taking the precision, recall and f1 score into consideration. The CRNN model contained 2 layers of RNN with gated recurrent units (GRU) for summarising 2 dimensional patterns from the output of the of 4 CNN layers. Irene et al. (2019) adopt RNN for the purpose of sequence modelling and CNN for learning the audio descriptors. A two layer RNN model that is based on LSTM which includes multiple gates that help in the model understanding of what to forget and what to remember about the previous samples via an internal state is implemented. The exploding gradient phenomena and vanishing gradient related issues that are observed in RNNs are effectively addressed by LSTM.

For classification Kim et al. (2019) use a data augmentation-based deep residual bidirectional gated recurrent neural network which studies long-term dependencies and assures the information transmission validity via residual connections and bidirectional cells. Abdelhameed et al. (2018) propose a CNN model along with Bi-LSTM. In Bi-

LSTM one layer contains 2 LSTM blocks and the temporal information is processed in 2 opposite directions simultaneously.

To conclude the CNN, CNN-LSTM and the CNN-Bi-LSTM algorithms will be used in the model for genre classification.

2.4 Recommendation Systems

Recommendation systems can be divided into 2 groups: CB filtering and collaborative filtering (CF). In CF the relationships between Q users and R items is represented by a Q x R rating matrix and the recommendations are established on the obtained similarities between Q users or rows and R items or columns. There is no dependency on the features of the item in the recommendation process of CF. The cold-start issue cannot be resolved using CF. In CB the process for recommending is focused on the feature analysis that depict the items. This algorithm allows to recommend items where the ratings are absent thus reducing the cold-start problem. A few recommendation systems use a hybrid of both the CF and CB for improved recommendations. Through this the advantages of CB which is taking into account the attributes of the item and CF which is considering the user's feedback are combined. This hybridization by combining the methods can be classified in switching, weighted, feature combination, mixed, feature augmentation, cascade and meta-level methods (Chiliguano and Fazekas; 2016).

Nafea et al. (2019) found that the effective techniques for a recommendation system implementation are the cosine similarity metric,the Pearson correlation coefficient and the k-means clustering algorithm. The recommendation accuracy is measured using the the root mean squared error (RMSE) and the mean absolute error (MAE). The similarity metrics are used in content based filtering and it is used to calculate the distance or the closeness between 2 feature vectors. Its very important because it is used to find out which items will be preferred. There are 4 similarity metrics mentioned in (Nafea et al.; 2019) which are Euclidean distance, Manhattan distance, Pearson correlation and Cosine similarity. From these Cosine similarity is the most popular and obtained the best results. Cosine similarity is also used in the studies conducted in (Chen et al.; 2013; Shakirova; 2017).

In (Han et al.; 2018) recommendation systems are categorised as CB recommendations, CF and metadata-based recommendations. The outcome of the metadata-based recommendations lack relevancy and are simple. CF need a huge amount of data which leads to matrix sparsity issues and the cold start problem occurs. The CB recommendations are audio based and require relatively less user data. The Earth Mover's Distance algorithm is used as a distance metric.

To conclude since this study lacks user data, CB recommendation is used with cosine similarity which is most popular with better performance as compared to the others.

2.5 Important Literature Summary

In Table 2 the most important literature from Sections: 2.1, 2.2, 2.3, 2.4 are listed on which this study is based.

Table 2: Literature Summary Table.

Author(s) and Title	Aims and objective	Models Applied	Dataset	Findings relevant to the review
Defferrard et al. (2016) - "FMA: A Dataset for music analysis"	Comparing different datasets used for the evaluation of several tasks in MIR, which is a field concerned with searching, browsing, and organizing large collections of music	NA	FMA dataset with 106,574 clips, 16,341 artists and audio data	FMA dataset provides the qualities of permissive licensing, large scale, available audio, future proof and reproducible, metadata rich, easily accessible and quality audio
Choi et al. (2016) - "Automatic Tagging using Deep Convolutional Neural Networks"	Evaluate the STFT, MFCC and Mel-spectrogram representations before passing it to the model	Feature extraction by Mel-spectrogram	MagnaTagATune Dataset and MSD	Considering the audio input representations, utilizing Mel-spectrograms performed better as compared to MFCCs and STFTs
Abdul et al. (2018) - "An Emotion-Aware Personalized Music Recommendation System Using a Convolutional Neural Networks Approach"	Incorporates a DCNN for the latent feature extraction of the music data.	DCNN	Million Song Dataset	The Mel-spectrogram of the audio track and the metadata is used by the DCNN for the classification into genres. Instead of the sigmoid function, ReLU is used for faster convergence. For increased processing speed the DCNN is executed in parallel on the GPU using the keras library.
Sainath et al. (2015) - "Convolutional, Long Short-Term Memory, Fully Connected Deep Neural Networks"	Recommend music that is multi-context-aware based on user's audio preferences	The combination of CNNs, LSTMs and DNN (CLDNN)	300 thousand English-spoken utterances, 3 million utterances	CLDNN gives a 4% to 6% rise in the WER as compared to LSTM which is the strongest among the 3 models.Using the DNN layers, converts the LSTM output to a more discriminate space and it is easier for the output targets to be easily predicted
Abdelhameed et al. (2018) - "Deep Convolutional Bidirectional LSTM Recurrent Neural Network for Epileptic Seizure Detection"	To detect the epileptic seizure in raw EEG signals using a deep convolutional BiLSTM system	Convolutional BiLSTM	EEG dataset from the Department of Epileptology at Bonn University	The model outperforms the previous models implemented with respect to classification accuracy in spite of working on raw EEG signals
Han et al. (2018) - "Music Recommendation Based on Feature Similarity"	To propose a music recommendation model on the basis of content similarity and use audio content similarity to make accurate recommendations when user data is absent	Earth Mover's Distance algorithm	Created database consisting of 6 kinds of music which are quiet, happy, sad, romantic, excited and inspirational	CF needs a huge amount of data which leads to matrix sparsity issues and the cold start problem occurs. Use the CB recommendation model When user data is lacking and to resolve the cold-start problem
Nafea et al. (2019) - "A Novel Algorithm for Course Learning Object Recommendation Based on Student Learning Styles"	To explore the best similarity metrics which are to be used in recommendation systems	Similarity metrics of Euclidean, Manhattan, Person correlation coefficient and Cosine	Moodle log-file at AAST	Cosine similarity is the most popular and obtained the best results.

3 Methodology

It is observed that 'Cross Industry Standard Process for Data Mining' (CRISP-DM) is best fit for this research (Chapman et al.; 2000). The CRISP-DM technique is altered and split into 6 stages according to the research of building a music recommendation system which can be seen in Figure 2. A detailed understanding of every stage in this process is discussed in this section:

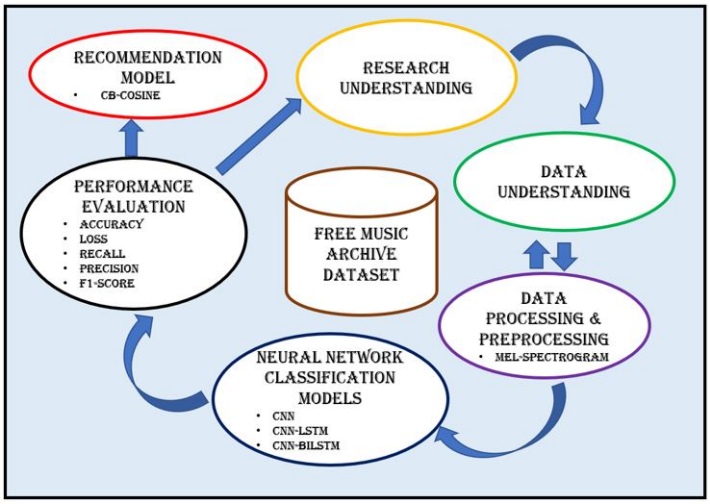

Figure 2: CRISP-DM

3.1 Research Understanding

The understanding of the objectives of the research is covered in the study's initial phase where there is a transformation of the gathered knowledge to a problem of machine learning. The aim and complete planning of the study are discussed in this stage. The aim of this study is to build a music recommendation system using deep learning techniques. The automated process of recommending songs resolves several issues faced using manual methods even with expert participation. A song also known as an anchor track would have to be inputted by a user and tracks with similar features to it will be recommended by the model. This will help the ever-competitive music industry grow further and increase user satisfaction.

3.2 Data Understanding

The FMA dataset Defferrard et al. (2016) is used which is open and easily accessible as mentioned in Section 2.1. It offers high-quality and full-length audio, features that are pre-computed, combined with user-level and track metadata, etc. The fma_small dataset consisting of 8000 audio files in the mp3 format is considered for this research. Each audio track is of 30 seconds with 8 balanced genres as seen in Figure 3. The total size of the dataset is 7.2 GB. The dataset can be downloaded using the link `https://github.com/mdeff/fma`. In addition to it, the track.csv file present in fma_metadata is also used which holds the song's metadata such as the title, ID, genre, tags, artists, and play counts for the 106,574 audio tracks.

3.3 Data Cleaning, Selection and Audio Pre-processing

The small dataset is used which consists of 8,000 tracks of 30s each in the mp3 format with 8 balanced genres as seen in Figure 3. The track metadata is also considered. Both the files are read into python using pandas. The audio tracks that are corrupt or have a duration of fewer than 28 seconds are discarded. 6 audio tracks are discarded in the process. A 28 seconds cutoff was selected to splice all of the tracks into equal parts of 10. The track CSV file is filtered out by 'small' in the 'subset' column giving 8000 rows.

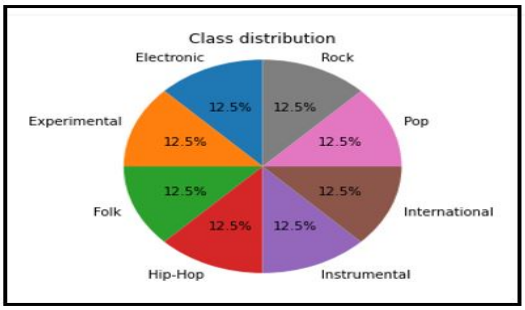

Figure 3: Balanced Data

The data from the audio track in the mp3 format is converted into a Mel-spectrogram. As mentioned in Section 2.2, it is observed that the Mel-spectrogram is most effective as compared to other representations (Choi et al.; 2016). Firstly, a specific sampling rate of 22,050 is used to process the audio signal. A window size of 2,048 (n_fft) samples the input. Each time a hop of size 512 (hop_length) is made to sample the upcoming window. Secondly, the time domain is transformed to the frequency domain by computing each window's Fast Fourier Transform (FFT). Thirdly, the Mel scale is generated when the whole frequency spectrum is divided into 128 (n_mels) frequencies that are evenly spaced. Finally, the spectrogram is generated by decomposing the signal's magnitude for each window into its components which corresponds to the Mel scale's frequencies[1]. Figure 4 displays the Mel-spectrograms of the 8 genres in the dataset.

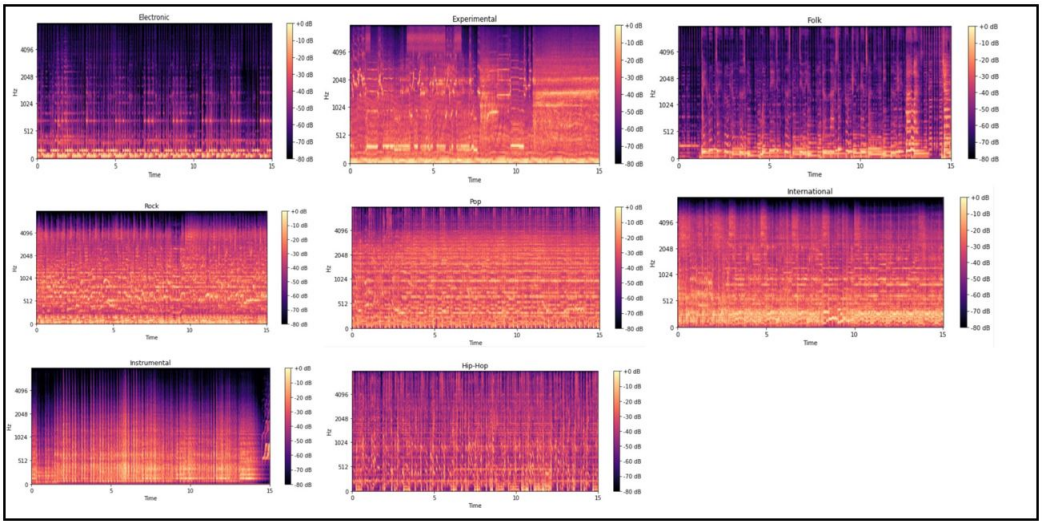

Figure 4: Mel-Spectrograms of the 8 Genres

[1]https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0

3.4 Neural Network Classification Models

The CNN, CNN-LSTM, and CNN-BiLSTM models are used in this research for feature engineering and classification. CNN is the base model used in various researches as mentioned in Section 2.3 and it has been tuned in this study to function optimally. These models are used for extracting the important features and for genre classification to aid in the recommendations. They take into consideration the spectrogram's frequency features along with the patterns of the time sequence as seen in Section 2.2. The 3 models are discussed below:

3.4.1 Convolutional Neural Networks

CNN, as used in (Chang et al.; 2018; Abdul et al.; 2018; Choi et al.; 2016), are techniques used to classify objects that comprise of a spatial neighborhood. CNNs are basically made up of 3 forms of recurrent layers: convolutional, pooling and fully connected. In the convolutional layer, multiple filters are used as feature detectors and they move over the input signal and then convolve with it. A stride measure that is configurable manages the filter's shifting distance over the input signal. The subsequent features are down-sampled for a reduction in the dimensions to reduce the computational power and decrease overfitting in the pooling layer. The fully connected layer is the final layer of the CNN (Abdelhameed et al.; 2018). A CNN model is implemented as seen in Section 2 where it is used in a number of studies as a good classifier. The CCN has 24 layers consisting of a convolution layer with a 2x2 kernel, batch normalization layer, average pooling 2-dimensional layer, flatten layer, dropout layer, and dense layers with ReLU and Softmax activation.

3.4.2 Convolutional Neural Networks with Long Short-Term Memory

LSTM, as implemented in various studies such as in (Tao et al.; 2019; Fulzele et al.; 2018), where it collects and incorporates the past data as information to forecast future outputs. In LSTM the long-term dependency issue is resolved, contains feedback connections and complete sequences of data can be processed[2]. From Section 2 the CNN-LSTM model is implemented in which feature are extracted with the CNN layers and the sequence prediction is carried out by the LSTM layers. The CNN-LSTM model consists of 26 layers with the 23 layers similar to the CNN model and an addition of 2 LSTM layers with 128 neurons each. The reshape and permute layers are added in between the CNN and LSTM layers for feature dimension reduction before the output of the CNN is passed to the LSTM layer. The output of the LSTM layers is inputted to the dense layers which are responsible for producing a feature representation of a higher order that can be easily separated into the various classes as needed (Sainath et al.; 2015).

3.4.3 Convolutional Neural Networks with Bidirectional Long Short-Term Memory

A single layer in BiLSTM as used in (Abdelhameed et al.; 2018), contains 2 LSTM blocks that process information that is temporal at the same instance in opposite directions. One of the LSTM blocks processes the time instance of the segment in the forward direction beginning from the starting time instance until the end. The same segment is processed

[2]https://medium.com/@premtibadiya7/music-genre-classification-using-rnn-lstm-1c212ba21e06

by the other LSTM block in the reverse order. Each of these 2 blocks generates its own outputs. The BiLSTM's final output at every single instance of time is measured by integrating the outputs of each block (Abdelhameed et al.; 2018). The CNN-BiLSTM model consists of 26 layers. The initial layers are that of CNN as seen in the CNN model mentioned in Section 5.4.1 followed by the reshape and permute layers for feature dimension reduction to synchronize with the input of the BiLSTM layer. A BiLSTM layer is followed by an LSTM layer with both having 128 neurons each and finally, the dense layers.

3.5 Content Based Recommendation Models

The content-based recommendation systems recommend objects which are identical to those that were used or rated before by the user which infers that the recommendations are based on the content of the item preferred previously. CB recommendation systems contain various similarity metrics that are used to calculate the distance or closeness between 2 feature vectors. From Section 2.4 the cosine similarity which is the most common similarity metric is used (Nafea et al.; 2019). A CB recommendation system will be used in this study because the dataset lacks user data and to resolve the cold-start issue as mentioned in Sections 2.1 and 2.4.

3.5.1 Cosine Similarity

The cosine similarity calculates the angle between the 2 feature vectors. It is calculated as the scalar product divided by the product of the magnitudes as seen in Equation 1

$$\cos(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{xy}}{\|\mathbf{x}\|\|\mathbf{y}\|} = \frac{\sum_{i=1}^{n} \mathbf{x}_i \mathbf{y}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{x}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{y}_i)^2}} \tag{1}$$

The c(x,y) values lie between -1 and 1 in general and within 0 and 1 if the x and y coordinates are non-negative values (Nafea et al.; 2019).

4 Design Specification

The architecture design seen in Figure 5 is followed to build an efficient music recommendation system that successfully generates recommendations. In the below data flow architecture design all the techniques and tools used for developing a music recommendation system are displayed. It is divided into 3 components which are: the user layer; the data persistence layer and the business logic layer. The data persistence layer contains the data source which is the FMA songs in the mp3 format and the track's metadata in the CSV format. The data is loaded in Jupyter, cleaned and the corrupt files are discarded using Python. This component connects with the business logic layer. The data preprocessing, neural network classification models and recommendation model are present inside the business logic layer. The features are extracted through Mel-spectrograms using Librosa and are stored in a NumPy array. This array is inputted to the three neural network classification models which use TensorFlow and Keras. **The ModelCheckpoint of Keras is used to store the models in the HDF5 format. The model corresponding to an optimal point of low loss and high accuracy is used for the recommendation model.** Finally, the recommendation model is used to recommend the appropriate audio tracks. The training models are run on the Colab notebook

due to the availability of a larger GPU and quicker processing. This layer connects with both the other layers which are the user layer and the data persistence layer. The user layer recommends audio tracks to the user depending on the track selected by the user.

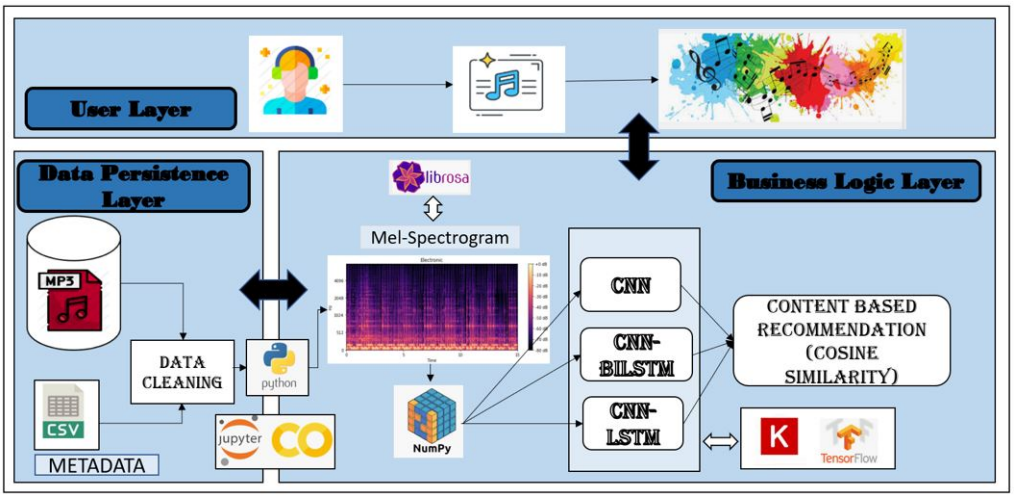

Figure 5: Data Flow Design Architecture

5 Implementation

This section describes the overall implementation of developing a music recommendation system. The following procedure is executed in order to implement the methodology mentioned in Section 3.

5.1 Data Gathering and Data Cleaning

The small dataset of FMA is used which consists of 8,000 tracks of 30s each in the mp3 format with 8 balanced genres. Along with this dataset, the track metadata is also considered. Both the files are read into Python using Pandas. The rest of the cleaning process is mentioned in Section 3.3

5.2 Feature Extraction

The mp3 audio tracks are processed one at a time. Each track is converted into a Mel-spectrogram using the Librosa library. **The Mel-spectrogram is stored in a buffer and sliced into 10 parts for faster processing**. The sliced images are then converted into a NumPy array which is used in the neural network classification models. **The buffer is cleared after each track is processed for the optimal usage of the RAM**.

5.3 Train-Test-Validation Split

The data is split at 3 levels. It is first split into train and test before storing it into a NumPy array. 500 songs covering all the genres is extracted and stored in the test section as NumPy arrays. This test data is used to evaluate the recommendation system's

accuracy based on the genre. The train data in a NumPy array is split and shuffled further into train and test in the ratio 9:1 using Sklearn's model_selection. The test data is used to evaluate the model and the training data is further split into train and validation in the ratio 9:1 which is utilized for the neural network classification models.

5.4 Neural Network Classification Models

5.4.1 Convolutional Neural Network

The CNN structure which is the base model consists of 4 layers of 2D convolutional with filters as (64-128-254-512). The input shape to the CNN model is (128, 128, 1). Each convolutional layer has a kernel of 2 dimensions which is used for the extraction of specific features from the input. There is also a batch normalization layer with each convolutional for the purpose of scaling and normalizing the activation or inputs.

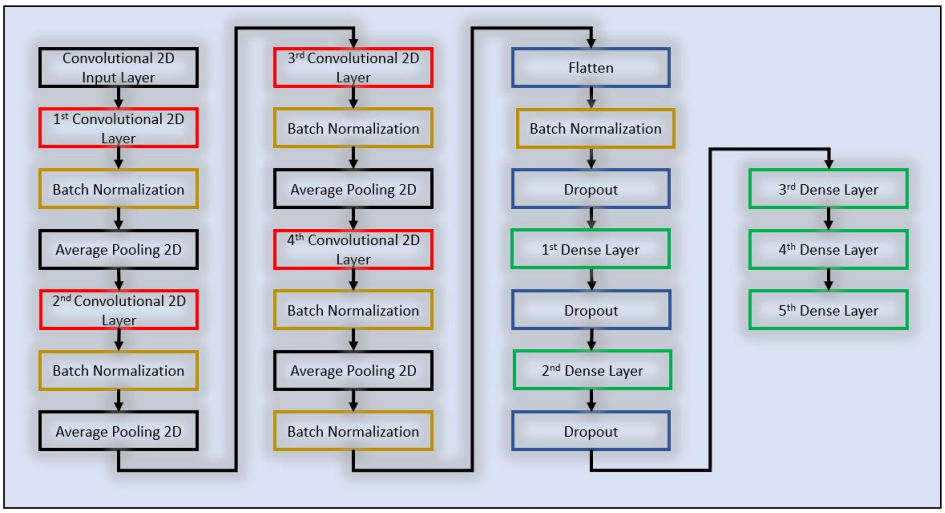

Figure 6: CNN Architecture

The ReLU activation is used for quicker convergence and for decreasing the vanishing gradient issue. AveragePooling2D layers are implemented with the same strides which retain the less important information with a certain pool size for downsizing. 3 Dropout layers with varying rates of 0.6,0.5 and 0.25 are considered for preventing the overfitting of the data. The 5 Dense layers having neurons as (1024, 256, 64, 32, 8), contain the ReLU activation functions for the first 4 layers and the final dense layer consists of the Softmax function because the classification is multi-class. Modelcheckpoint is used as explained in Section 4. The CNN model is built with TensorFlow and Keras. The flow and layers of the CNN model are seen in Figure 6.

5.4.2 Convolutional Neural Network with Long Short-Term Memory

The CNN-LSTM model consists of a similar CNN structure as mentioned in Section 5.4.1 . 2 LSTM layers are inserted after the convolutional blocks. A Reshape and a Permute layer are added between the CNN and LSTM layers for feature dimension reduction so that the output of the CNN layer is synchronized with the input of the LSTM layer. Each LSTM layer contains 128 neurons and has an input shape of (128, 128, 1). The LSTM layers are followed by 5 Dense layers with filters as (1024-256-64-32-8). The CNN-LSTM

model is built with Tensorflow and Keras. The flow and layers of the CNN-LSTM model are seen in Figure 7.

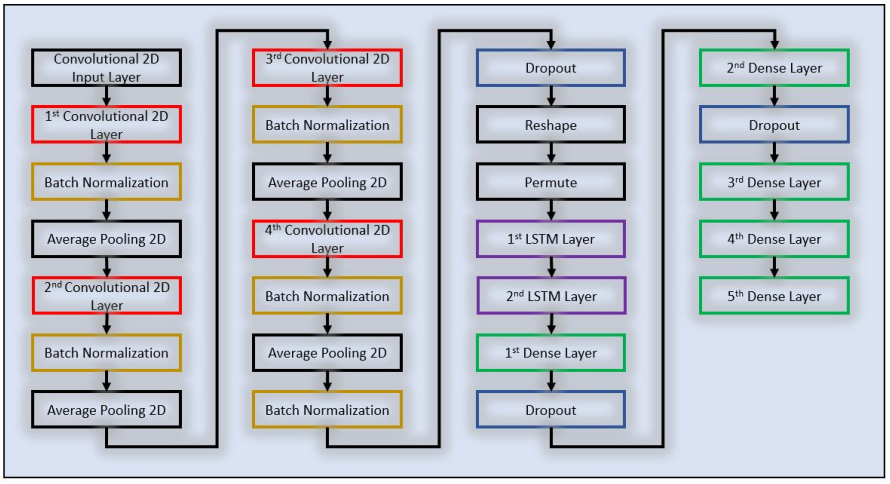

Figure 7: CNN-LSTM Architecture

5.4.3 Convolutional Neural Network with Bidirectional Long Short-Term Memory

The CNN-BiLSTM model consists of a similar CNN-LSTM structure as mentioned in Sections 5.4.1 and 5.4.2. From the 2 LSTM layers, the first is a bi-directional LSTM.

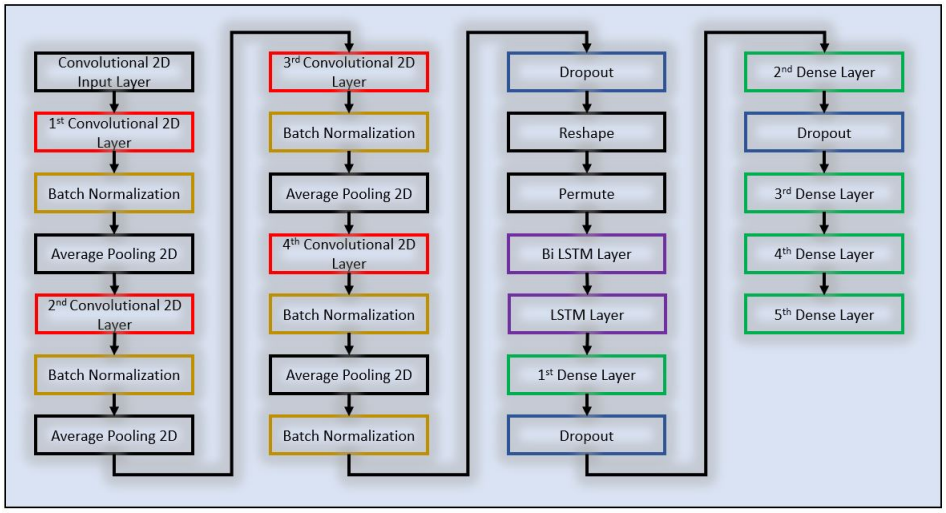

Figure 8: CNN-BiLSTM Architecture

This BiLSTM layer contains 128 neurons taking an input shape of (128, 128, 1). The CNN-BiLSTM model is built with Tensorflow and Keras. The flow of and the layers of the CNN-BiLSTM is seen in Figure 8.

5.4.4 Hyperparameter Tunning

On creating the model architecture several parameters were correctly set for the model to perform well and optimally deliver results adapted to this study's objectives. In order

to prevent the model from overfitting the train data, a dropout parameter is set. By tuning this hyperparameter overfitting is prevented. Multiple dropouts of 0.6, 0.5, and 0.25 are set for the CNN model whereas for the CNN-LSTM and CNN-BiLSTM models dropouts of 0.6, 0.5, 0.5, and 0.25 are set. Since this is a multi-class study, the categorical cross-entropy is used. The Adam optimizer is used in this study for all 3 models with a learning rate of 0.00005. Modelcheckpoint which is part of the callback function of Keras saves the model at certain points in the training with its weights. The model with the optimal point of high accuracy and low loss is considered. The 3 models were run for 29 epochs.

5.5 Recommendation Model

The optimal Neural Network classification model using ModelCheckpoint in the HDF5 format is used in the CB recommendation model. The test data as mentioned in Section 5.3 is passed to the model. A similarity score is computed for all the songs with respect to the anchor audio track. The cosine similarity metric as mentioned in Section 3.5.1 recommends a similar song to the anchor audio track from the data inputted to the model. This part is implemented in Google Colab and the number of recommendations can be configured depending on the requirement. In this study its set to 5 and 10. Figure 9 displays the recommendations for the anchor audio track which is inputted.

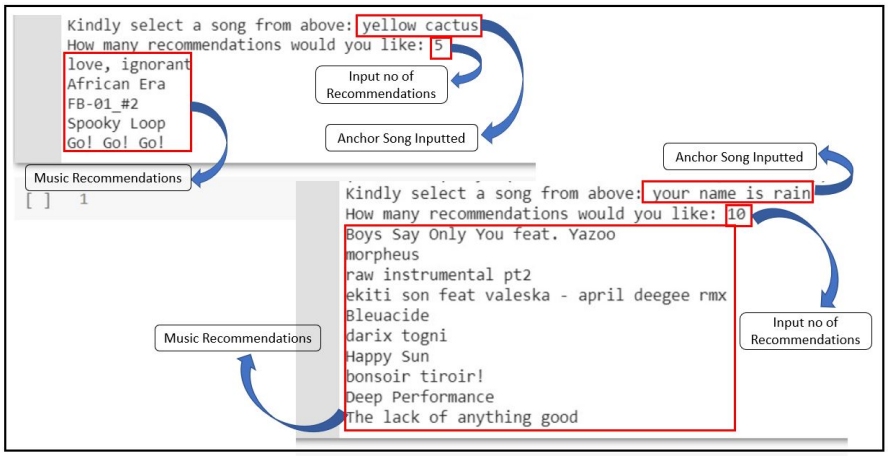

Figure 9: Music Recommendations For Anchor Audio Track

6 Evaluation, Results & Discussions

This section is divided into two parts: firstly, the neural network classification models are evaluated based on the classification accuracy, logarithmic loss, precision, recall, and f1 score; secondly, the recommendation model is evaluated based on the genre accuracy for 5 and 10 recommendations. The following are the basis on which the model is evaluated.

6.1 Classification Model Evaluation

The 3 neural network classification models are evaluated based on the classification accuracy, logarithmic loss, precision, recall, and f1 score. Table 3 contains the evaluation metrics for the 3 models considering the test data. It is seen that the CNN-LSTM model

Table 3: Classification model evaluation with test data

Classification Models	Classification Accuracy	Logarithmic Loss	Precision	Recall	F1 Score
CNN	0.69	0.93	0.77	0.63	0.69
CNN-LSTM	0.72	1.06	0.75	0.70	0.72
CNN-BiLSTM	0.71	1.03	0.74	0.68	0.71

has the best accuracy among the 3 with good accuracy, precision, recall, and f1 score. The accuracy could not increase further due to data constraints and short sequences of 3 seconds each that are passed to the LSTM layer. Hence, the CNN-LSTM model is considered to be the best model for genre classification. Figure 10 displays the CNN, CNN-LSTM and CNN-BiLSTM model's classification accuracy, logarithmic loss and f1 score considering the train validation data in a graphical representation over 29 epochs. The blue line graph represents train and yellow represents validation.

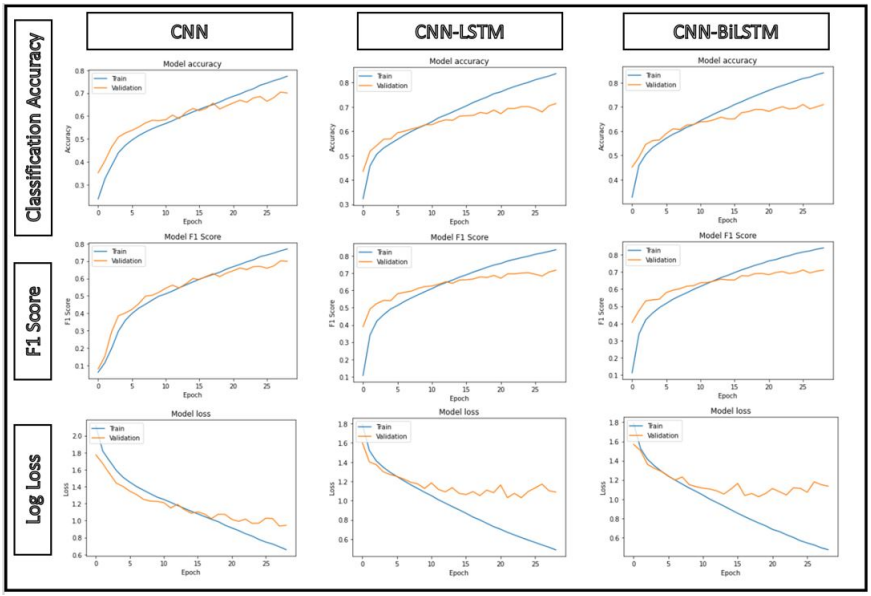

Figure 10: Classification model's train-validation graphical representation of classification accuracy, f1 score and log loss

From the graphs in Figure 10 it is inferred that the accuracy and f1 score of all the models increase gradually with the increase in epochs however the loss decreases. It is also observed that the CNN-LSTM model has the best validation accuracy with a smooth curve increasing gradually.

6.2 Recommendation Model Evaluation

Figure 11 contains the accuracy of all the genres for 5 and 10 recommendations. The test data of 500 songs were used for the evaluation. It signifies how many times will an anchor song of one genre be recommended songs of the same genre by the model. The recommendation model is evaluated as present in (Elbir et al.; 2018; Elbir and Aydin; 2020). The recommendation model with the CNN-LSTM classifier has the best overall

genre accuracy and the model performed better in recommending songs of the same genre as compared to (Elbir et al.; 2018; Elbir and Aydin; 2020)

	Recommendation with CNN		**Recommendation with CNN-LSTM**		**Recommendation with CNN-BiLSTM**	
Genre	First 5 Songs	First 10 Songs	First 5 Songs	First 10 Songs	First 5 Songs	First 10 Songs
Electronic	0.90	0.91	0.96	0.93	0.94	0.95
Experimental	0.79	0.79	0.94	0.97	0.82	0.81
Folk	0.80	0.77	0.94	0.92	0.97	0.95
Hip-Hop	0.93	0.93	0.98	0.97	0.94	0.94
Instrumental	0.23	0.18	0.83	0.48	0.47	0.35
International	0.98	0.97	0.99	0.99	0.99	0.99
Pop	0.77	0.78	0.92	0.91	0.91	0.90
Rock	0.90	0.88	0.96	0.97	0.95	0.95

Figure 11: Recommendation Model Results

6.3 Results & Discussions

The CNN-LSTM model performed the best as compared to CNN and CNN-BiLSTM. A combination of LSTM for handling sequences with CNN improved the classification as seen in the findings from Section 6.1. Although the accuracy could be increased by taking longer sequences to be inputted to the LSTM layers. A CB recommendation system was used due to the dataset lacking user information. All of the models with the recommendation system had their share of good genre accuracy. However, the instrumental genre had accuracy on the lower side. This is partly because the data used to test the recommendation model is picked at random, hence a possibility of data imbalance. Also, the instrumental genre has features very close to a few genres like rock. Thus, an instrumental genre track can be recommended tracks of the rock genre.

7 Conclusion and Future Work

Recommendation systems have been increasingly important in today's society, being implemented in a number of areas. This system can recommend audio tracks to the user depending on the similarity of the audio features. In this study, the FMA dataset with audio data is used and its features are extracted using a Mel-spectrogram. The CNN, CNN-LSTM, and CNN-BiLSTM models are used to classify the music according to the genre. The CNN model is tuned and used as a base model. The CNN-LSTM and CNN-BiLSTM models are built on the base model and the results are evaluated. It was observed that the CNN-LSTM model performed better than the rest considering the classification accuracy, precision, and recall. Due to the dataset lacking user data, a CB recommendation system with cosine similarity is used for recommendations to resolve the cold-start problem. The music recommendations (5 and 10 recommendations) considering the genre accuracy of all the 3 models are good however, the CNN-LSTM model with cosine similarity gave the best results. 5 and 10 recommendations were given for an anchor track using the best model.

In the future, I propose to use user data along with the audio features and build a hybrid recommendation system combining CF and CB. Larger sequences will be inputted to the LSTM and Bi-LSTM layers to enhance the classification and better accuracy. Since all the models gave close results an ensemble model can be implemented which will consist of the combination of the CNN, CNN-LSTM, and CNN-BiLSTM models.

References

Abdelhameed, A. M., Daoud, H. G. and Bayoumi, M. A. (2018). Deep convolutional bidirectional lstm recurrent neural network for epileptic seizure detection, *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)* pp. 139–143.

Abdul, A., Chen, J., Liao, H.-Y. and Chang, S.-H. (2018). An emotion-aware personalized music recommendation system using a convolutional neural networks approach, *Applied Sciences* **8**: 1103.

Adiyansjah, Alexander, G. and Derwin, S. (2019). Music recommender system based on genre using convolutional recurrent neural networks, *Procedia Computer Science* **157**: 99–109.

Al Mamun, M. A., Kadir, I., Rabby, A. S. A. and Al Azmi, A. (2019). Bangla music genre classification using neural network, *2019 8th International Conference System Modeling and Advancement in Research Trends (SMART)*, pp. 397–403.

Bakhshizadeh, M., Moeini, A., Latifi, M. and Mahmoudi, M. T. (2019). Automated mood based music playlist generation by clustering the audio features, *2019 9th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 231–237.

Chang, S., Abdul, A., Chen, J. and Liao, H. (2018). A personalized music recommendation system using convolutional neural networks approach, *2018 IEEE International Conference on Applied System Invention (ICASI)*, pp. 47–49.

Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. R. H. and Wirth, R. (2000). Crisp-dm 1.0: Step-by-step data mining guide.

Chen, C., Tsai, M., Liu, J. and Yang, Y. (2013). Music recommendation based on multiple contextual similarity information, *2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, Vol. 1, pp. 65–72.

Chiliguano, P. and Fazekas, G. (2016). Hybrid music recommender using content-based and social information, *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2618–2622.

Choi, K., Fazekas, G. and Sandler, M. (2016). Automatic tagging using deep convolutional neural networks.

Dai, J., Liu, W.-J., Ni, C., Dong, L. and Yang, H. (2015). "multilingual" deep neural network for music genre classification.

Defferrard, M., Benzi, K., Vandergheynst, P. and Bresson, X. (2016). Fma: A dataset for music analysis.

Elbir, A. and Aydin, N. (2020). Music genre classification and music recommendation by using deep learning, *Electronics Letters* **56**(12): 627–629.

Elbir, A., Bilal Çam, H., Emre Iyican, M., Öztürk, B. and Aydin, N. (2018). Music genre classification and recommendation by using machine learning techniques, *2018 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pp. 1–5.

Fulzele, P., Singh, R., Kaushik, N. and Pandey, K. (2018). A hybrid model for music genre classification using lstm and svm, *2018 Eleventh International Conference on Contemporary Computing (IC3)*, pp. 1–3.

Gupta, S. and Arora, S. (2016). Handling cold start problem in recommender systems by clustering demographic attribute.

Han, H., Luo, X., Yang, T. and Shi, Y. (2018). Music recommendation based on feature similarity, *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pp. 650–654.

Irene, R. T., Borrelli, C., Zanoni, M., Buccoli, M. and Sarti, A. (2019). Automatic playlist generation using convolutional neural networks and recurrent neural networks, *2019 27th European Signal Processing Conference (EUSIPCO)*, pp. 1–5.

Jiang, M., Yang, Z. and Zhao, C. (2017). What to play next? a rnn-based music recommendation system, *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 356–358.

Karunakaran, N. and Arya, A. (2018). A scalable hybrid classifier for music genre classification using machine learning concepts and spark, *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, pp. 128–135.

Kim, H., Kim, G. Y. and Kim, J. Y. (2019). Music recommendation system using human activity recognition from accelerometer data, *IEEE Transactions on Consumer Electronics* **65**(3): 349–358.

Nafea, S. M., Siewe, F. and He, Y. (2019). A novel algorithm for course learning object recommendation based on student learning styles, *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*, pp. 192–201.

Nasrullah, Z. and Zhao, Y. (2019). Music artist classification with convolutional recurrent neural networks, *2019 International Joint Conference on Neural Networks (IJCNN)* pp. 1–8.

Rajanna, A. R., Aryafar, K., Shokoufandeh, A. and Ptucha, R. (2015). Deep neural networks: A case study for music genre classification, *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 655–660.

Sainath, T. N., Vinyals, O., Senior, A. and Sak, H. (2015). Convolutional, long short-term memory, fully connected deep neural networks, *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4580–4584.

Shakirova, E. (2017). Collaborative filtering for music recommender system, *2017 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 548–550.

Tao, Y., Zhang, Y. and Bian, K. (2019). Attentive context-aware music recommendation, *2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC)*, pp. 54–61.