MUSIC CLASSIFICATION AND STATISTICAL ANALYSIS

Final report

Tyler Desharnais, Lucie Gaye, Brandon Rasgaitis, Julien Toulon

Abstract:

In this project, we will study music data. We will try to train our artificial intelligence to classify songs by their specific genre.
For this we will start by creating the artificial neural network. We will then look for a database containing twenty thousand songs and their spectrograms image files, in order to train it to recognize and classify them. Once trained we will test our intelligence.
Projects of this type already exist but we will try to improve them. Indeed there are many musical styles and some music cannot be classified in an absolute way. We can therefore try to improve the interface of analysis of the results.
Thus, after training the dataset and configuring the Convolutional Neural Network model we will be using, our AI would be able to place the song's spectrogram image on a graph or to give the influences drawn from each musical style and their nuances.
The commercial benefit of this project would be for the artists to better classify their music and better fit into the current trends. It could also allow them to statistically analyze the biggest commercial successes and understand what could lead them to success. By comparing the artist's music with the top of the charts, it could allow him to know if his music has a chance to appeal to the greatest number of people.
Our approach to better classify different genres required different models to be used. These models included, a long short-term memory network (LSTM), Convolutional neural network (CNN), linear model, and a Multilayer perceptron model. We found that each model proved to be better for classifying different genres of music based on the ROC-AUC scores and accuracy provided on training per epoch. Our models include Dropout, kernel regularizers, activity regularizers, and gaussian noise to improve accuracy and aid in reducing over fitting in our model.
Our results showed that different models of varying degrees of accuracy predicting the genre for each spectrogram. We found that LSTM models for example, predict hip-hop and Latin music much more accurately then the linear model and the CNN model. However the linear model performed better predicting the genre of classical music.

Introduction:

The overall problem being addressed in this project is the lack of nuance in music genre classification. Music is an art form that takes inspiration from many different technical aspects that make up the structure of a song. This makes the classification of particular songs to specific genres difficult as they can't be classified in an absolute way.

One of our solutions is to read specific data from a song's spectrogram, which is an image detailing the specific aspects that design a song's structure. The aspects that create the structure of said song include beat, tempo, decibel readings, length of time, danceability, frequency, etc. Improving the interface of genre classification to include statistical data of songs would allow artists and composers to better understand the technical aspects of how their song is designed, provide a tool to promote their music to a wider audience or better cater to a niche fan base, provide a tool to design and classify new genres by using specific data from different songs and their spectrograms, and finally it could aid in the commercial success of an artist and their music by detailing different statistics of a genre's commercial success. Statistical data includes a correlation heatmap, to see which song features correlate strongly with popularity. The results reveal that songs that have higher decibel readings, tend to correlate strongly with popularity. This means that people generally tend to like louder music. This is also true for dance-ability and it's correlation with popularity. The different chroma-gram and spectrogram images detail the song's varying Hz over time and the change in pitch over time for different song's such as 6 foot 7 foot by Weezy F Baby, and 21 Guns by Green day. These are two songs from different genres, rap and alternative rock.

Our model can then predict the genre by analyzing how the images for each song's spectrogram are different. Our results detailed that LSTM models are better for labeling genres of song's such as hip-hop, and Latin as the timing of different beat frequencies, Hz, and pitch change over short periods of time, whereas genres similar to rock performed better in our CNN model as the song's feature variances were much longer making a better image for the model. The MLP model excelled in predicting genres for hip-hop and metal, predicted dance and acoustic music averagely, and Latin music poorly.

Related Works:

There are many other Music Genre Classification deep learning models that use the same techniques we use in order to classify and label each song. One such example, used by Arsh Chowdhry who posted their project on their blog post at www.clairvoyant.ai involves creating a Convolutional Neural Network model to classify music samples into different genres. In their model, they are utilizing the GTZAN dataset, as well as the MNIST of sounds data set. The GTZAN data set is a collection of 10 genres with 100 audio files each 30 seconds in length. They also split their data set into two CSV files. One containing the mean and variance computed over multiple features, and the other containing the same structure, just split into 3 second audio files. Arsh also uses raw wave audio file images, spectrogram images, Spectral Rolloff graphs, chroma feature graphs, and zero crossing rate graphs to determine specific aspects that design the structure of a song. The data is analyzed using Librosa, a python package for audio analysis. They build their Music Genre Classification model using multiclass support vector machines, K-Means clustering, K-Nearest Neighbor algorithm, and a Convolutional Neural Network. This data is then used for the creation of a music recommendation system that automatically classifies song genres for the user in their CNN model. When building their model, they used the Adam optimizer, sparse categorical cross-entropy for their loss function, and dropout to avoid over fitting in their model. This is where our project differs. We will be analyzing the results of the data that makes up the structure of a song and predict what genre it's classified under. The dataset we use will be different as well as it was collected through spotify's API. We will also split each song up into 1 minute increments for analysis by our model. For example we take a rap song created by Eminem and have our program predict the genre and form its spectrogram. From there our AI will look at a piece of data such as beat and predict how similar it is to other genre beats and will classify them and their accuracies. This will allow an artist to design their songs using different technical aspects from other genres and provide a more in-depth way to classify their song. Instead of just labeling a song with the genre of rap, it will label it rap-metal for example as the beat has similarities with rock. What is similar about our approach is that we also used a CNN model as research revealed that it was the best model for accuracy, and we also adopted the K-Nearest Neighbor algorithm as well.

Data:

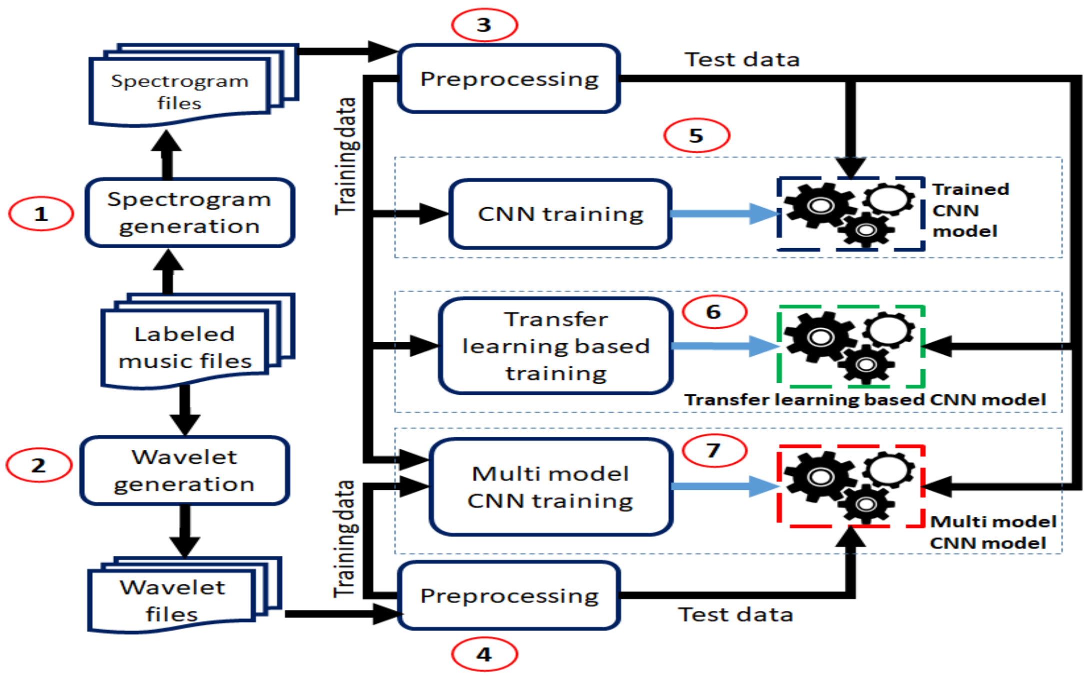
For our music genre classification model, we are working with a dataset of twenty thousand mp3 audio files, along with each mp3 files correlating spectrogram images. The audio files were collected through Spotify's developer API as it allows access to user related data, like playlists and music that the users saves within their music library. The data collected through Spotify's API is then saved into a CSV file which will be used as our training data for each model. This required that we manipulated our dataset. To do this, we had to import the following custom made scripts:

```
from prepare import load_Data, handle_nulls, prepare_df, change_dtypes
from preprocessing import spotify_split, scale_data, create_features
from explore import param_viz, corr_heatmap, catplot_viz
from librosa_Visuals import visualize_Chromagram_y_Spectogram
```

In our data manipulation, we read in our data frame, created features, dropped duplicates, changed the dtypes, and dropped the following columns: song, artist, year, duration_minutes, duration_seconds, explicit, Unamed: 0, and has_feat. After manipulating the dataset, we then split the data by popularity using spotify_split(). Our data set now only shows the different features of a song and its mean variance correlating with popularity. After, we create a new column in our data frame labeled genre before extracting and scaling the data. When extracting and scaling the data we use fit_transform to fit the label encoder and return the encoded labels for genre classification. The output of our data is saved in google drive as model_output which is where all of our spectrograms with their appropriately labeled genres will go.

Methodology:

The following illustration details the process of how our Music Genre Classification model will pre-process, train, and output our data.



As you can see from the graph, the data is the music files, which are split up into training and test sets that will have to be pre-processed. The training data is trained using a CNN model, some type of transfer learning based model, and a multi CNN model. The multi CNN model includes LSTM-CNN, multi Convolutional model, and LSTM. CNN models are most commonly used to analyze visual images, perfect for our spectrogram images of each song. LSTM models are a variety of recurrent neural networks that are capable of learning long-term dependencies useful for sequence prediction problems. This is incredibly useful for our music genre classification model as songs are broken up into multiple time frames. To take it a step further, each feature of a song such as pitch can be adjusted for some length of time making LSTM a much needed tool to analyze and provide statistical data on the different song features. The reason we chose LSTM over RNN is that LSTM makes smaller adjustments on the data it's training. It forgets and remembers things selectively. How this relates to music is that not all song's features are the same, so we want our model to "forget" every time a new song's audio file is ready for an epoch of training. Using this information, our group decided to use a LSTM-CNN model to combine the best of both features and accurately predict different genres with a higher degree of nuance as it uses both the length of time for each song feature, and the spectrogram image to identify differences in song's. The idea behind this was that this type of model would allow us to more accurately label each song.

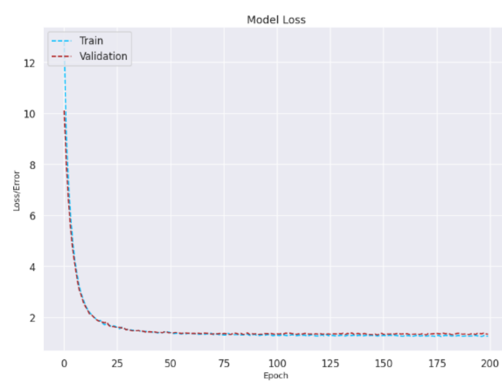
Experiment:

After detailing the methodology, we will now talk about the experiments. We implemented these two models, CNN and LSTM in google colab. After training the models, we have the following datas:

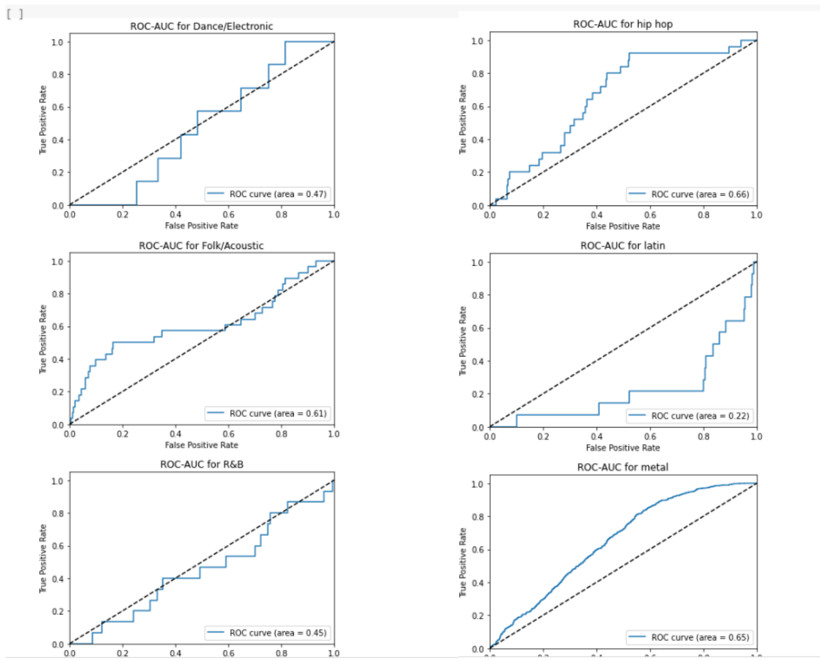
➔ CNN

First about the model accuracy and loss. For the CNN we can see the following graphics

```
Epoch 200/200
34/34 [==============================] - 0s 9ms/step - loss: 1.2745 - accuracy: 0.6826 - val_loss: 1.3436 - val_accuracy: 0.6588
15/15 [==============================] - 0s 5ms/step - loss: 1.3436 - accuracy: 0.6588
test accuracy:  0.6587982773780823
```



As we can see, the accuracy is pretty good. We also observed that each model is better for different types of music genre. To have more datas in this way, we can see the ROC-AUC diagram which show the false positives and false negatives for each type of genre. For the CCN, the ROC-AUC diagram looks like

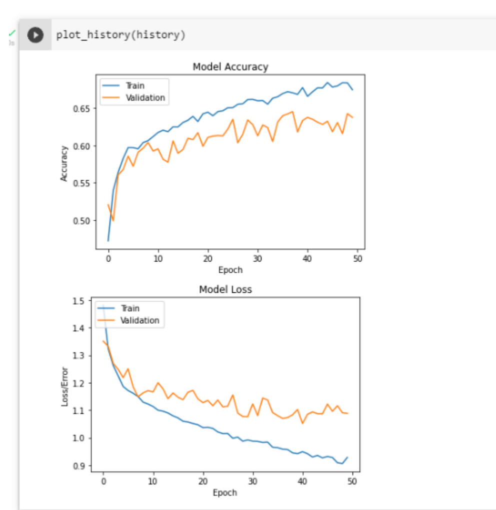


For the CNN, we can see that this model is better to predict metal and hip-hop genre than latin.

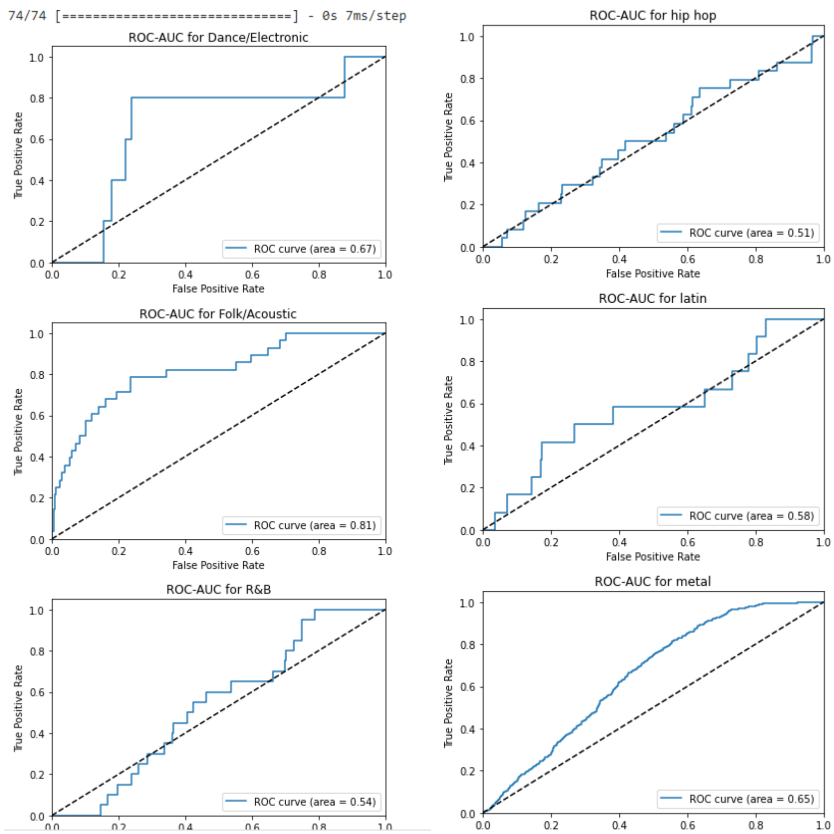
➔ LSTM

The LSTM model has also a pretty good accuracy as following

Epoch # 38 Performed Best



As we can see, the accuracy is close to the CNN model, around 0.65. Now we can take a look at the ROC-AUC diagram



As we can see, this model is better to predict the folk/acoustic and the dance/electronic genre.

Conclusion:

In result of this project, we were able to build A Neural Network model that provides a more nuanced genre labeling of songs as well as statistical data on commercial success, compared to similar existing tools.
We learned that different models were better than others at predicting genres, especially acoustic and hip hop. Convolutional Neural Networks are useful for accurately identifying and labeling image data. We also found out that LSTMs are useful handling time sensitive data.
Possible future works could include a feature to extract portions of a song such as beat, tempo, etc.
We could also use the extracted data to add to newly created songs for composers.
Another possible feature would be to create a Music recommendation system, and a playlist creation with cloud support.

References:

1.)Chowdhry, A. (2021, May 7). Music genre classification using CNN. Retrieved December 15, 2022, from https://www.clairvoyant.ai/blog/music-genre-classification-using-cnn

2.) Krohn, J., Beyleveld, G., & Bassens, A. (2019, September). Deep learning illustrated: A visual, Interactive Guide to Artificial Intelligence. Retrieved December 15, 2022, from https://learning.oreilly.com/library/view/deep-learning-illustrated/9780135116821/ch09.xhtml#ch09lev1sec5

3.) Li, A. (2021, May 20). Machine learning and recommender systems using your own Spotify Data. Retrieved December 15, 2022, from https://towardsdatascience.com/machine-learning-and-recommender-systems-using-your-own-spotify-data-4918d80632e3