Music Genre Classification

Dhruv Patel, Ishan Kumar, Avnish Asthana, Gio Visco
GithubLink

Project Topic:

Music is a huge industry with a diverse range of genres. Experts have been attempting to understand the complexities of sound and how different genres of music differentiate from each other on a mathematical level. Our project's goal is to identify music genres utilizing feed forward neural networks. We would like to see if numerical data extracted from audio files can predict the genre of music. This will be accomplished by using Librosa, a Python library, to extract tempo and spectral bandwidth from audio. This project utilizes two datasets generated by extracting data from the Librosa Library. Dataset 1 contains the only means of the features, while Dataset 2 also contains the variances of all the features. One of the key questions we aim to address is if one of knowing the distribution of the key features improves accuracy.

Music Files and Feature Extraction:

The GTZAN dataset is used throughout the project. This is a set of ten different genres, ranging from rock to reggae. There are 100 songs associated with each of these genres. As a result, we have 1000 audio files in our dataset. Every sound file has a length of 30 seconds and is formatted in .au format. The genre is as follows: 'blues', 'classical', 'country', 'disco', 'hiphop', 'jazz', 'metal', 'pop', 'reggae', and 'rock'.

We will be utilizing the librosa library to extract features from the audio files. From each audio file, we can extract the time series and the sampling rate. These can be utilized to extract the various features of the dataset.

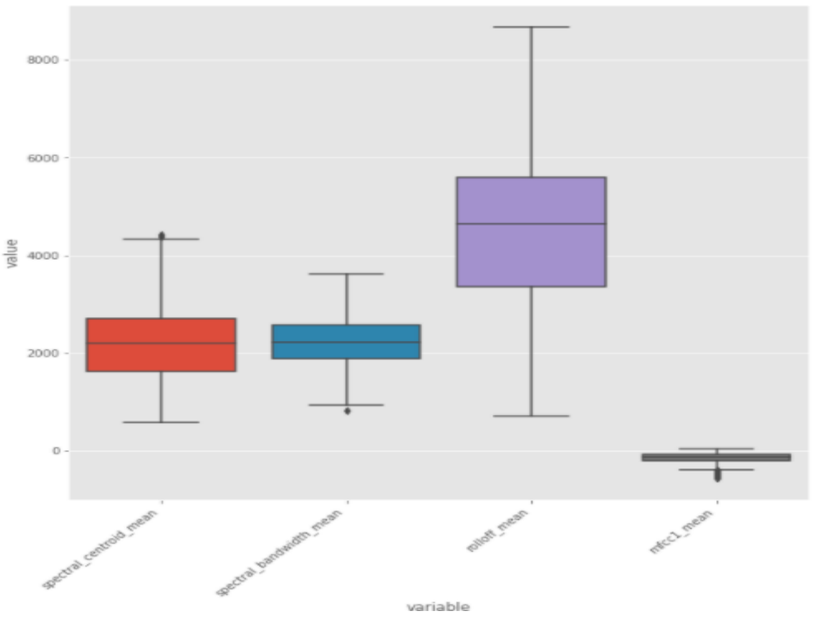
Our first data frame contains 26 numerical features and 2 categorical features. The categorical features are the name of the file and the label of the file. The label is later encoded into a numerical value. The majority of the 26 numerical features consist of mel frequency cepstral coefficients. The Mel scale is a scale that relates the perceived frequency of a tone to the actual measured frequency. Essentially, MFCC features represent phonemes as the shape of the vocal tract is manifest in them. There are 20 dimensions for these coefficients, so our dataset contains the 20 means for each of the mfcc. These coefficients are widely known as one of the most important features to apply machine learning models on audio files.

The other features present are the mean value of the following: spectral bandwidth, rolloff, zero crossing rate, chroma stft, spectral centroid and tempo. The number of times the amplitude of the speech signals passes through a value of zero in a given time interval/frame is measured by the zero-crossing rate, which is a key feature for identifying percussive sound.

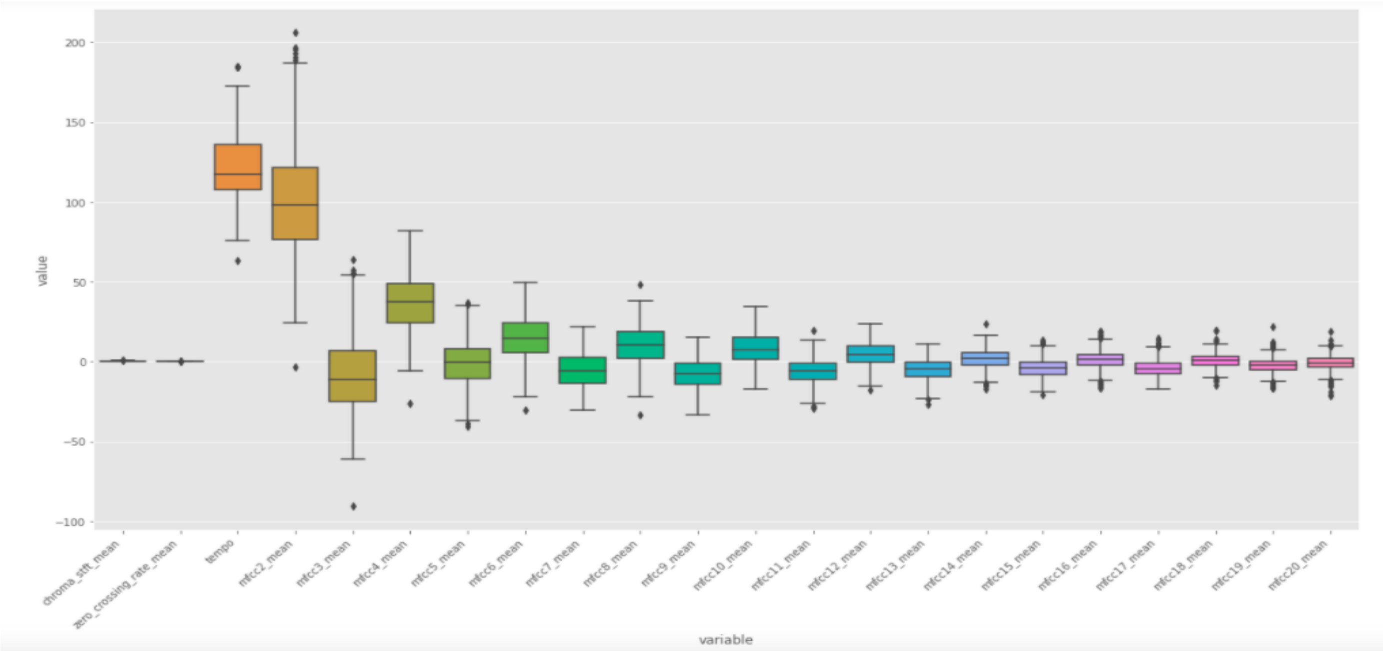
Our second data frame contains additional features representing the variances of all features that we had a mean listed for, nearly doubling the amount of features we have. We would like to explore if understanding the distribution of some of these features plays a key role in classifying the genres of each audio file.

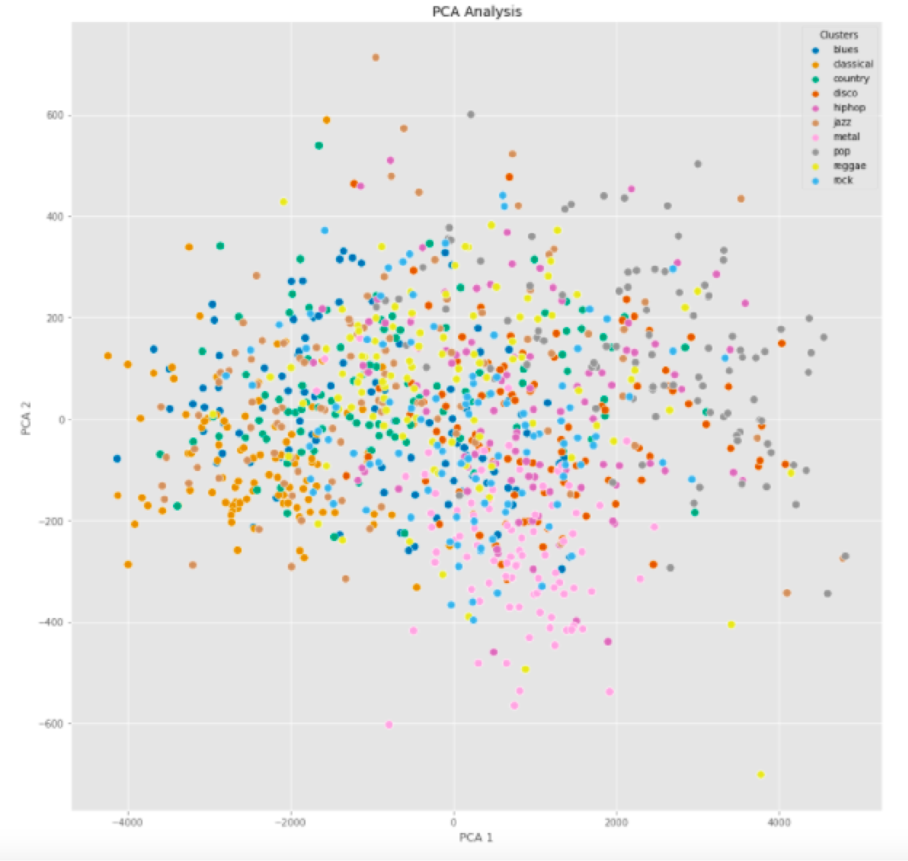
Exploratory Data Analysis

An exploratory analysis had to be performed because we wanted to understand the distribution of our features. Because we were using audio files that we turned into a CSV file, it was crucial that we find out how these values correspond to each other and what each value can do to help us figure out which model to use to classify the genres. In addition, we wanted to distinguish the clusters in a PCA analysis and if their distribution makes logical sense.



Our exploratory_analysis.ibynb file consists of different plots we created to help us understand the main characteristics of the data. We made a box plot of the spectral_centroid_mean, spectral_bandwidth_mean, rolloff_mean, and a single mfcc_mean to understand the variability between these different means and how it compares to the mffcc_means we are using to classify the data. Our plot showed us that the mfcc mean was very small, ranging from about -50 to 200, compared to the other variables, which ranges in the thousands. Because of this we knew that plotting all of the mfcc_means in that box plot would not be appropriate to understand the ranges. This is why in our second box plot, we included: chroma_stft_mean, zero_crossing_mean, tempo (doesn’t have a mean), along with all of the other mfcc_means to further understand the mean values of all the variables.





Additionally in our exploratory_analysis.ibynb file, we ran a PCA Analysis on the means of all the features of each song in the dataset, as plotted above. Here we can see that metal (pink dots) is the most distinguishable out of all the genres as there is less interference from songs of other genres towards the bottom of the plot. The plot also shows that blues, jazz, country, and classical music all have a good amount of overlap towards the left side of the plot. Pop and disco music also overlap a lot with each other on the right side of the plot, which makes sense as these genres use similar instruments and have similar sounds to the ear. Reggae, hip hop, and rock are spread out in the middle of the plot. Looking at this plot we can anticipate which genres the model will have a harder time distinguishing from each other.

The group also ran and plotted a second PCA using the variances of all the song features. This plot clustered all ten genres much closer together to the point where we could not distinguish any of the genres from our own observations.

We then used a correlation matrix to see how the variables compared to each other. Ideally, a correlation matrix would give us more insight on how the variables positively and negatively correlate. However for the mfcc_means there was no consistency shown in the matrix between each other, which was not helpful in understanding the variable importance. However something that the correlation matrix showed us was that the spectral centroid, spectral bandwidth, and rolloff are positively correlated with one another.

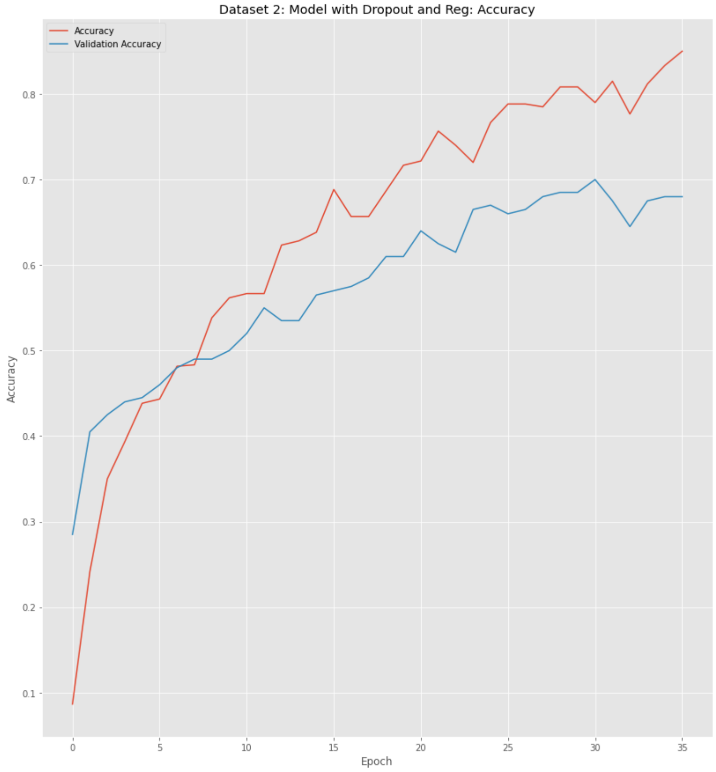
Models

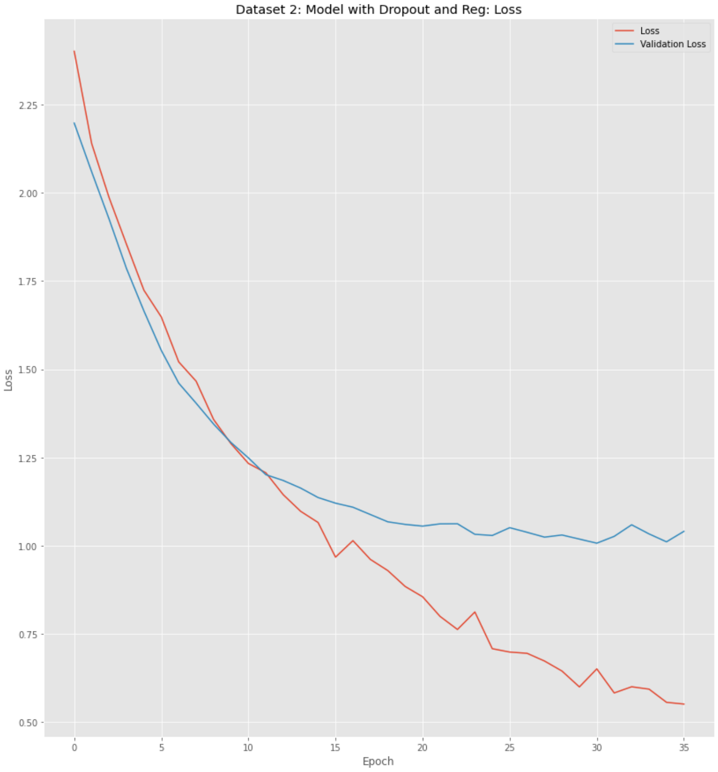
The primary model used for the analysis of our genre classification were feed forward neural networks. Each dataset was split into a 60/20/20 Train, Test, Validation split for the training of the neural network. Each dataset had three neural networks associated with them. This is one of the best models to use for this dataset, as the neural network will automatically adjust the weights for all of these features and designate different levels of importance to each. Other models such as decision trees rely on knowing where the decision tree splits on each of the feature values, and if that split is the best.

The first neural network simply contained 3 hidden layers with 256, 128 and 64 neurons respectively. The second neural network implemented 3 dropout layers, one after every hidden layer with 20% dropout rate each. The final neural network implemented L2 regularization for each of the dense layers. All these hyper parameters were tweaked throughout our process arriving at the values shown above.

Two techniques were used to reduce overfitting for our neural network. The first method used was to implement L2 regularization. This seems to perform better in terms of limiting our model from overfitting rapidly. Our key approach, one never discussed in the course, was to utilize a callback function to end training based on the delta of our validation dataset's loss. The patience level of this callback was set to 5, which decides how long it waits to see if the validity loss has decreased. To prevent neural networks from overfitting, training is halted if the validity loss does not decrease within the patience level.

Results and Analysis


Dataset 2: Model with Dropout and Reg: Accuracy


Dataset 2: Model with Dropout and Reg: Loss

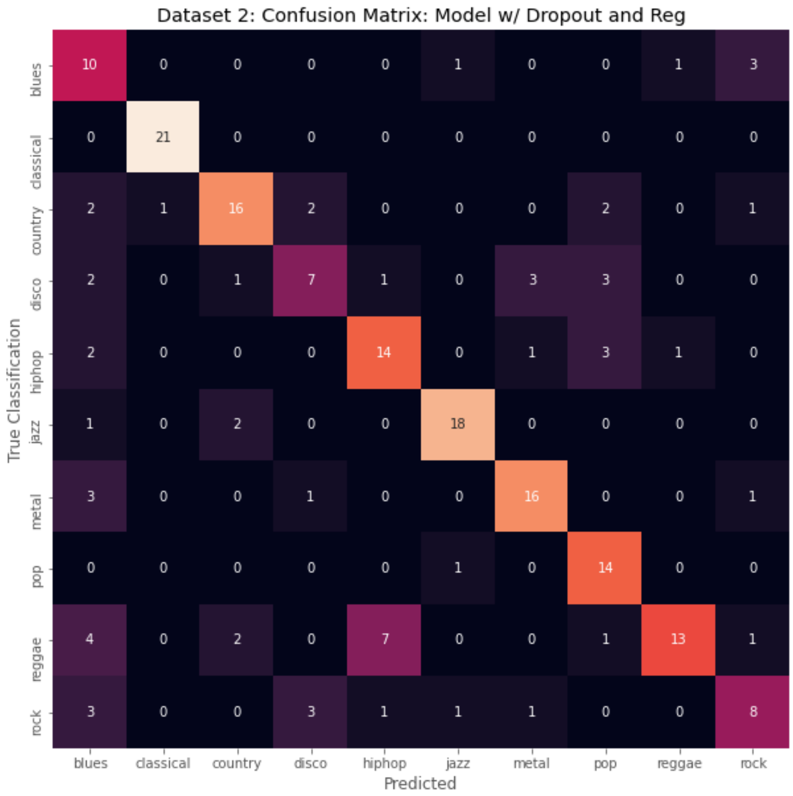
The above figure contains two graphs relating to the network with dropout & L2 regularization. The graph on the left shows the accuracy and validation accuracy of the model and the graph on the right shows loss and validation loss. The accuracy graph shows a steady increase in both accuracy and validation accuracy as epochs continue. Due to our callback function the training stops when the validation accuracy begins to plateau which can be seen here.The loss graph shows a steady decrease in both loss and validation loss as epochs continue. Both of these graphs show expected results for a successful model increase in accuracy and decrease in loss over time

Model Type	**Dataset 1** **Test Accuracy**	**Dataset 2** **Test Accuracy**
Normal Network	59.0%	65.5%
Network w/ Dropout	58.5%	69.0%
Network w/ Dropout & L2 Regularization	60.5%	68.5%

Note: Dataset 1 contains only the means of each of the features. Dataset 2 contains both the means and variances of each feature.

The above table summarizes the results for test accuracy across each of the models on the two separate datasets, it is clear that the networks trained on dataset 2 perform better than those trained on dataset 1. Accuracy is our best evaluation metric for these different models because our dataset is perfectly balanced (100 samples of each genre), hence no need for an F1, ROC, or AUC score. Of the three separate models trained on dataset 1 the network with dropout and L2 regularization slightly outperforms the others, but the differences in their performances are basically insignificant. As for the models trained on the second dataset there is a clear gap in performance between the normal network and each of the networks containing dropout layers. It can be concluded that the addition of dropout layers between our hidden layers greatly benefitted our accuracy for models trained on this dataset. The addition of L2 regularization did not affect the performances of this model, however it does provide another metric which reduces overfitting.

The following figure is a confusion matrix representing the classification of test data for our third model, which included Dropout and L2 regularization. When analyzing the following heatmap, we can see that this neural network perfectly classified classical music. This makes sense as classical music is highly distinguishable from other modern day genres. All models also had high levels of success classifying jazz and metal music, as they all contain very distinct characteristics. The network seems to classify a large proportion of reggae as hip hop and blues. Finally, the network had trouble identifying rock, blues and disco. These make sense as these genres tend to encompass a wide variety of different styles within them, while jazz and classical music tend to have one very familiar style.



Dataset 2: Confusion Matrix: Model w/ Dropout and Reg

	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
blues	10	0	0	0	0	1	0	0	1	3
classical	0	21	0	0	0	0	0	0	0	0
country	2	1	16	2	0	0	0	2	0	1
disco	2	0	1	7	1	0	3	3	0	0
hiphop	2	0	0	0	14	0	1	3	1	0
jazz	1	0	2	0	0	18	0	0	0	0
metal	3	0	0	1	0	0	16	0	0	1
pop	0	0	0	0	0	1	0	14	0	0
reggae	4	0	2	0	7	0	0	1	13	1
rock	3	0	0	3	1	1	1	0	0	8

Discussion and Conclusion

In conclusion, the different neural networks had very similar accuracy results on dataset 1 while the network with dropout and L2 regularization performed the best. Additionally we can see that dataset 2 had a better accuracy than dataset 1 on all three neural networks. Within dataset 2, the normal neural network was still the weakest in terms of accuracy, but in contrast to dataset 1 the network with only dropout outperformed the network with dropout and L2 regularization.

Since dataset 2 had a significantly higher accuracy than dataset 1, we can deduce that the variance of all the music features was a more effective way of engineering the features for the model to distinguish between the genres. We can also see that implementing dropout and L2 regularization to our network prevents our model from overfitting on the data. This led to the higher accuracies of those models on both datasets.

From looking at our confusion matrix heat map, we could see that jazz and classical music were the easiest genres for the models to identify. On the other hand, rock and disco were the most difficult genres for our models to classify. This is a little different than what we anticipated from our PCA analysis.

A possible way to improve is to normalize the dataset. The reason is that some of the features are in vastly different ranges, as shown from the boxplots mentioned above. Another possible way to improve the model is to test out a convolutional neural network on the spectral data of our songs. Running a filter through the dataset may possibly help the convolutional layers identify patterns within the classifications more effectively. A final way to improve our model is to increase the diversity of genres in our dataset.

Overall, we believe that the analysis performed by our neural networks provided sufficient results to be able to classify our audio files with their respective genres with fairly high accuracy. Throughout our different iterations of the dataset, we understood how interacting with the layers and tweaking the hyper parameters affect the accuracy and limit the overfitting of our neural networks. In the end, we were successfully able to find significant distinctions between the genres, we hope to be able to improve our models with the suggested approaches in the future.