Graduate Project

Sherwyn Braganza

December 13, 2022

Summary of Contents

1 Introduction and Background

The purpose of this project was to give us an opportuinity to experience i addressing a real world machine learning problem on data that we have some personal interest in working with. Apart from being able to choose our own dataset to work in, we were allowed to implement our choice of model architecture including whichever types of layers (e.g. dense, convolutional, recurrent, pooling, etc.) are appropriate for the problem.

2 The Dataset

The dataset I chose was obtained from Kaggle and contained songs and the measurable properties associated with them. The target variable for this dataset was the Genre the song belonged to.

The dataset contained 17,966 samples or songs with 17 features or attributes associated with the each sample. The features are as follows:

1. **Class** - The target variable pertaining to the Genre of the song
2. **Track Name** - Descriptive feature that holds the name of the song
3. **Artist Name** - Descriptive feature that contains the Artist's name.

4. **Popularity** - Metric Feature that corresponds to how popular the song is.
5. **Danceability** - Metric feature that corresponds to how easy it is to dance to the trace.
6. **Energy** - Metric feature that measures how lively the a song is.
7. **Key** - Categorical Feature that corresponds to the musical key a song is in
8. **Loudness** - Numeric Feature that corresponds to how lowd the song is.
9. **Mode** - Binary feature corresponding to the musical definition of mode.
10. **Speechiness** - Numeric Feature corresponding to the percentage of how much of the song is words.
11. **Acousticness** - Numeric Feature, measure of how acoustic a track is.
12. **Instrumentalness** - Numeric Featrue, measure of how much of the track is instrumental
13. **Liveness** - Numeric Feature, measure of reverberation time.
14. **Valence** - Numeric Feature, meaasure of how positive a track is.
15. **Tempo** - Numeric Feature, beats per minute of the track
16. **Duration** - Numeric Feature, duration of the track
17. **Time Signature** - Categorical Feature corresponding to the musical time signature

The target variable, the Class, which represents the genre contains values from $[0, 10]$ which corresponds to the following Genre(s):

0 - Acoustic/Folk

1 - Alt Music

2 - Blues

3 - Bollywood

4 - Country

5 - Hip Hop

6 - Indie

7 - Instrumental

8 - Metal

9 - Pop

10 - Rock

The data contained multiple samples with missing values or NaN values. The dataset was hence pre-processed to remove all these samples that had missing values. This resulted in removing 6183 samples from the original dataset and trimmed the dataset down to 11,813 samples.

The first two descriptive features - Track Name and Artist Name - don't really contribute much to the training and were ignored.

To get a better understanding of the data and which genre was showed more variance and prominence within a feature, I plotted each feature against the classes. The graphs below show my results.

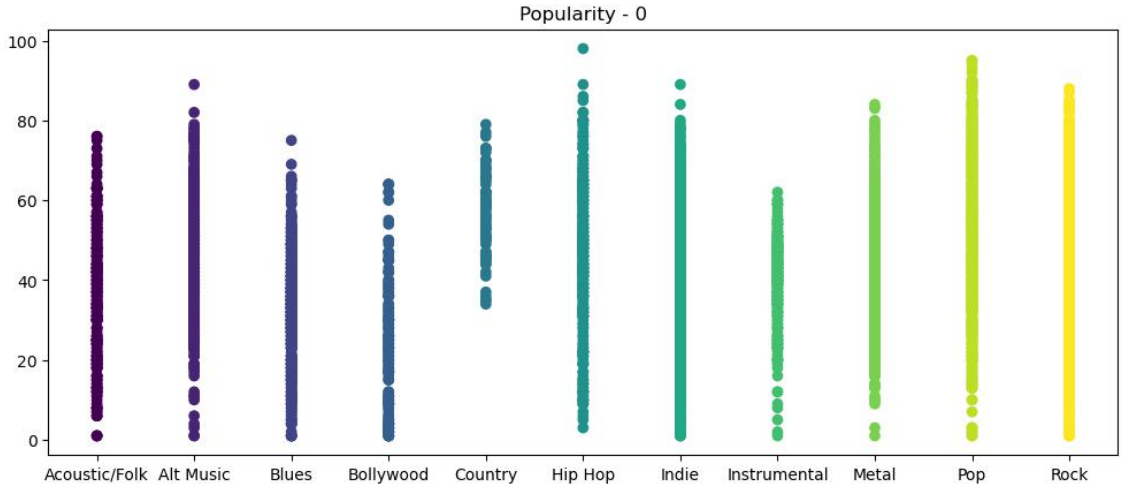

Figure 1: Popularity

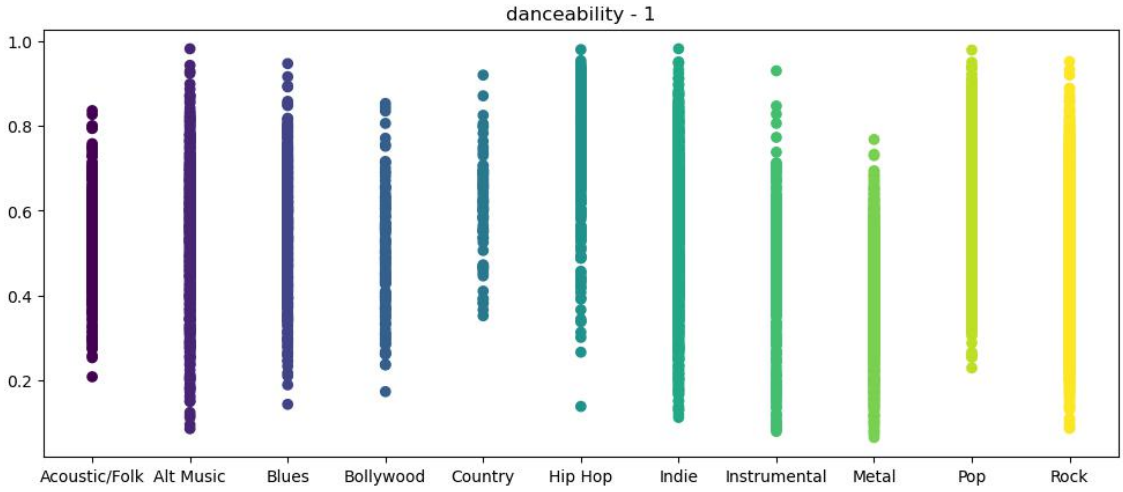

Figure 2: Danceability

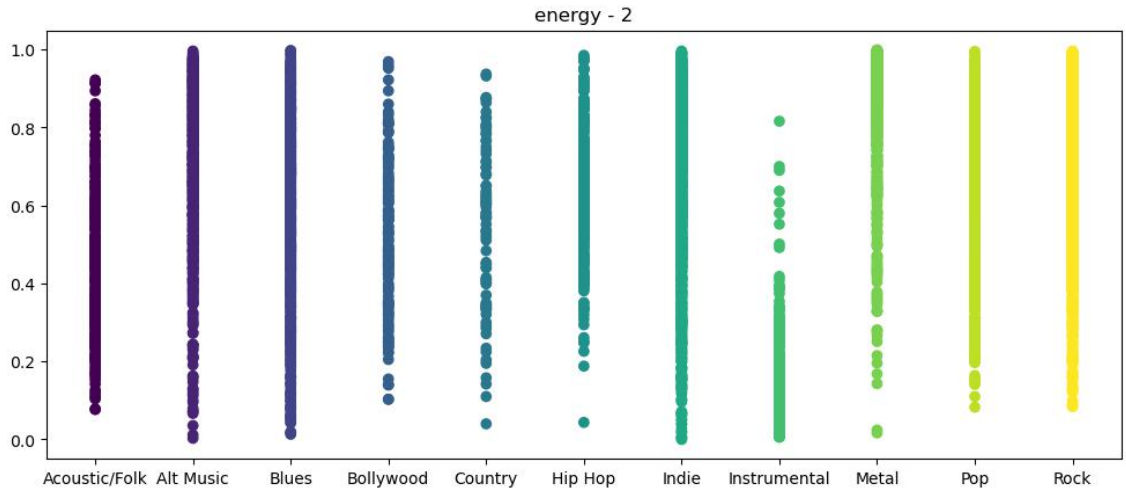

Figure 3: Energy

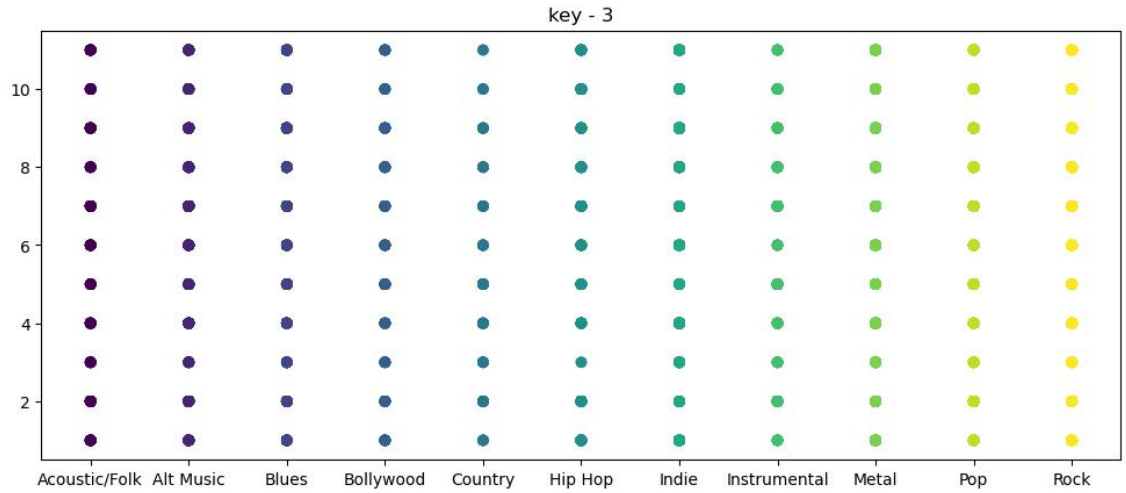

Figure 4: Key

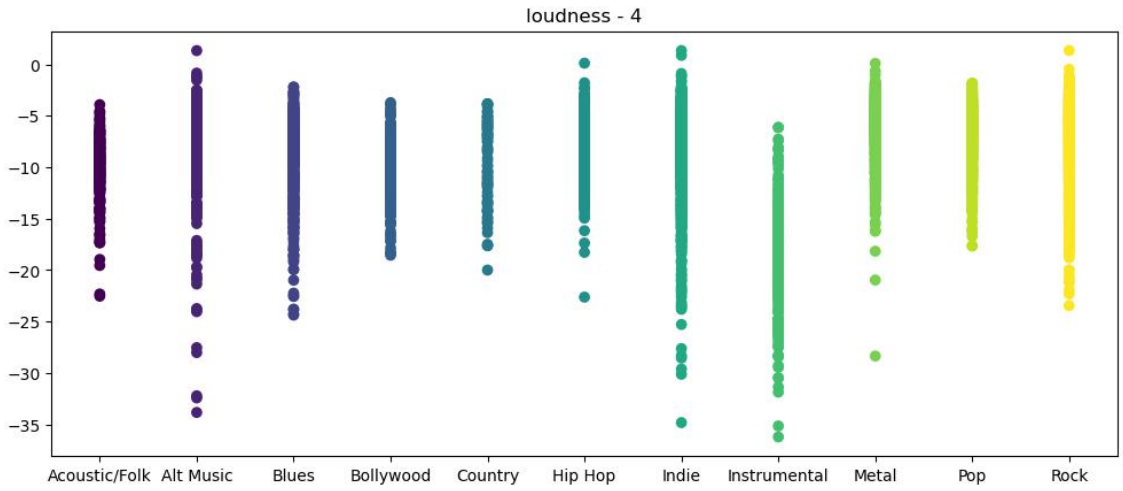

Figure 5: Loudness

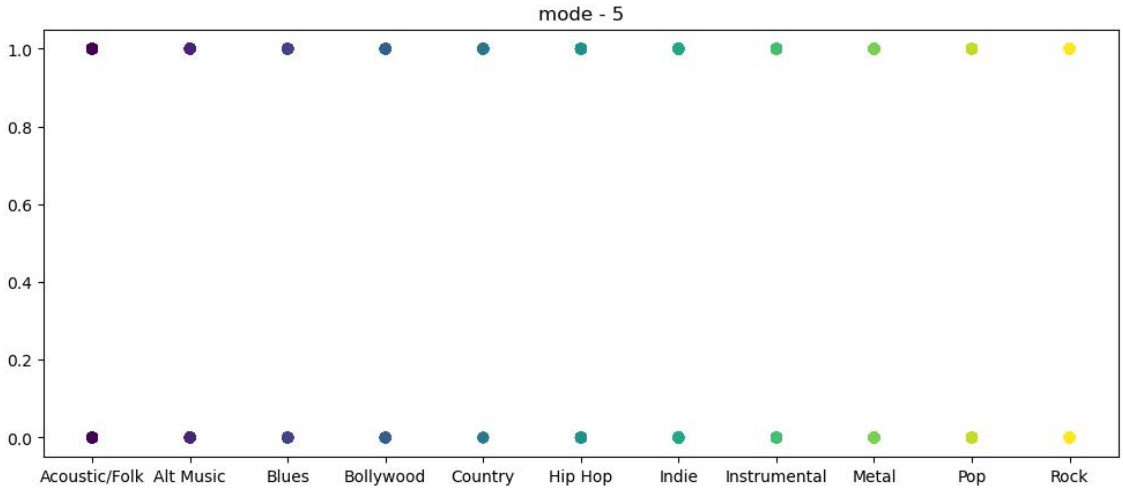

Figure 6: Mode

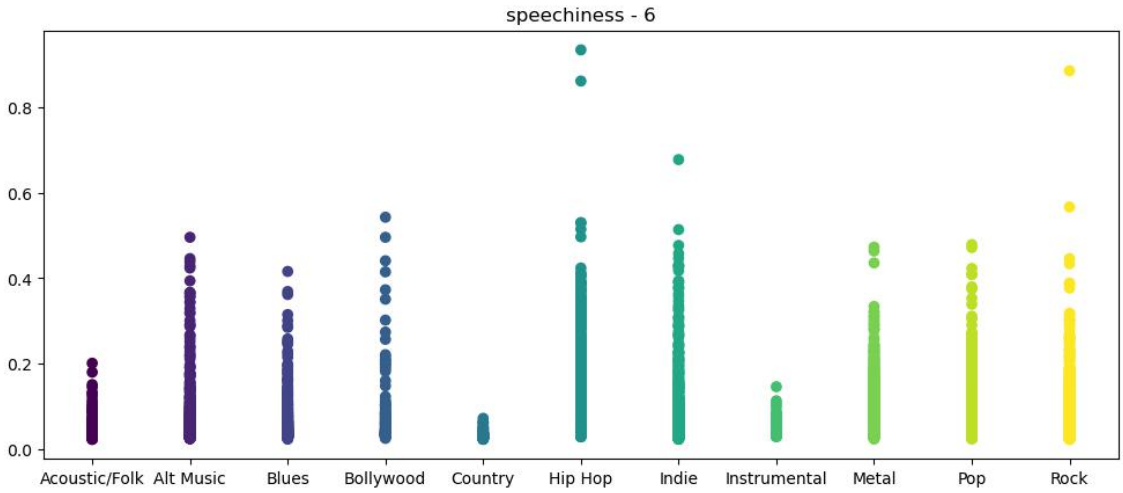

Figure 7: Speechiness

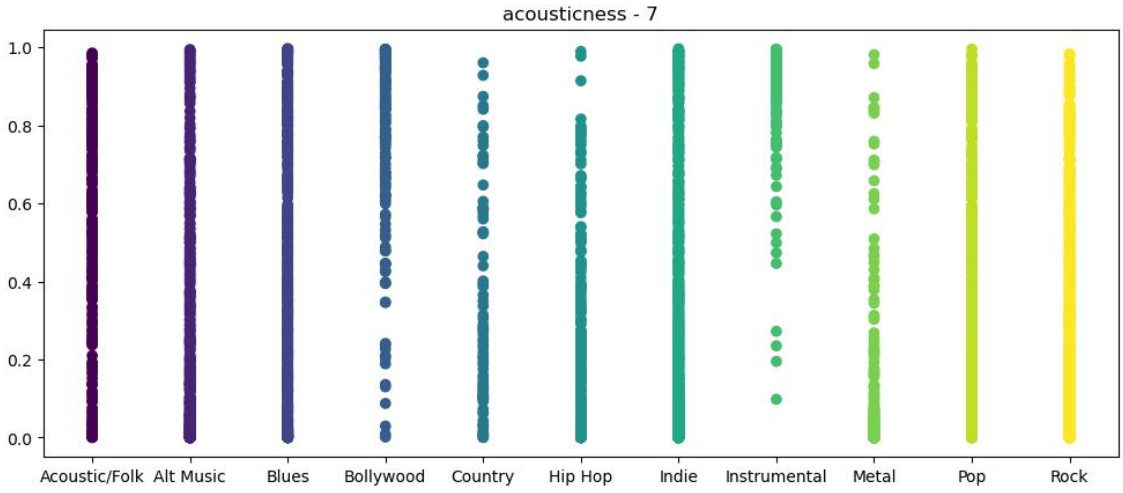

Figure 8: Acousticness

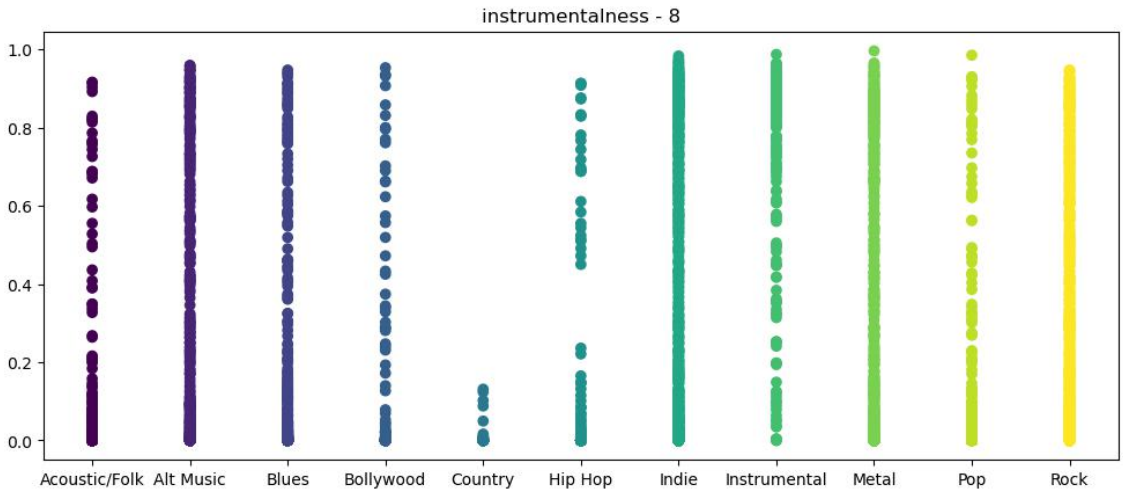

Figure 9: Instrumentalness

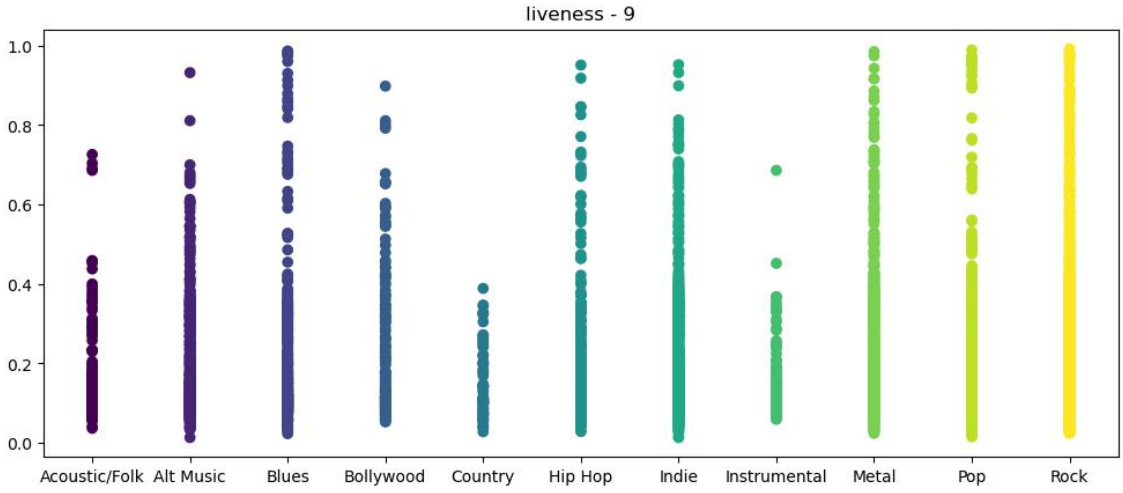

Figure 10: Liveness

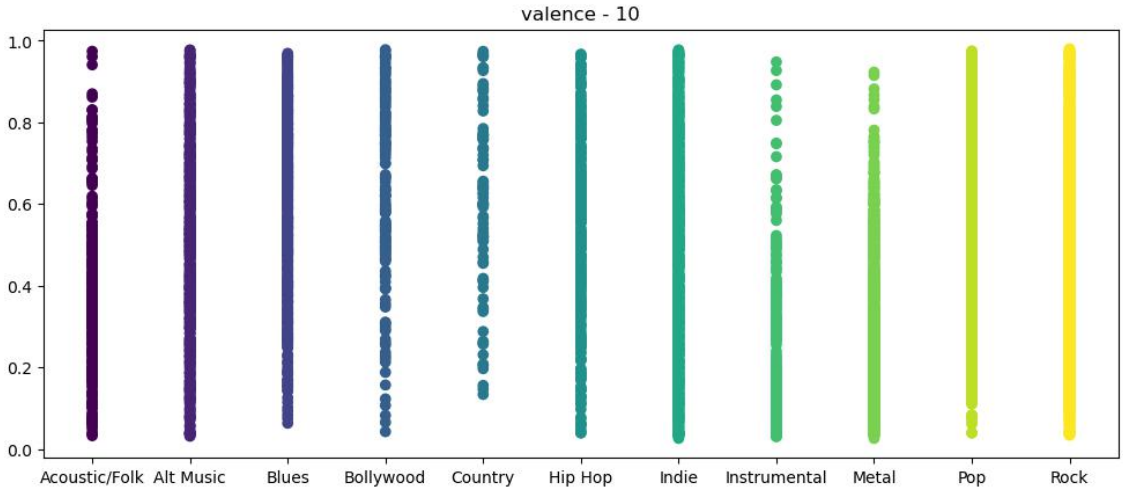

Figure 11: Valence

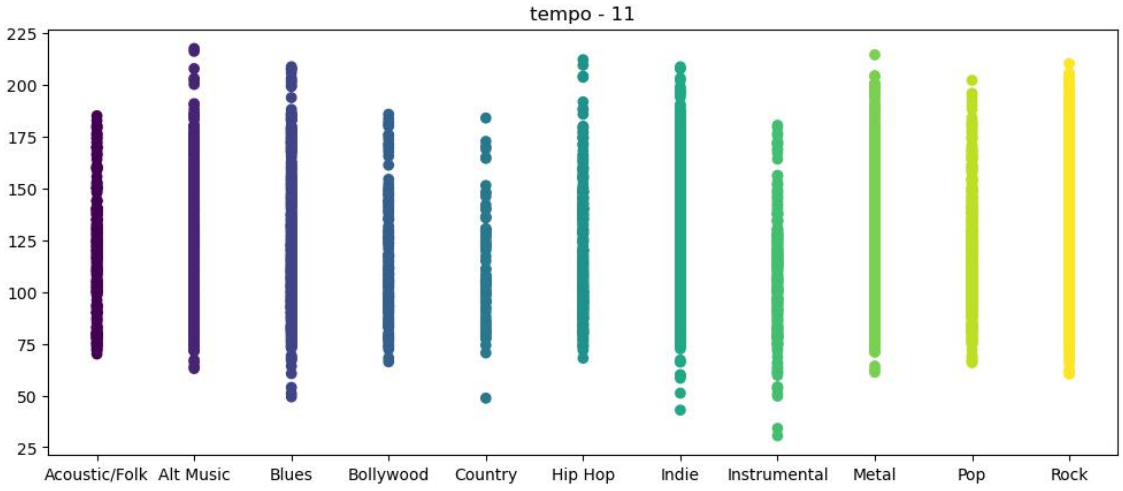

Figure 12: Tempo

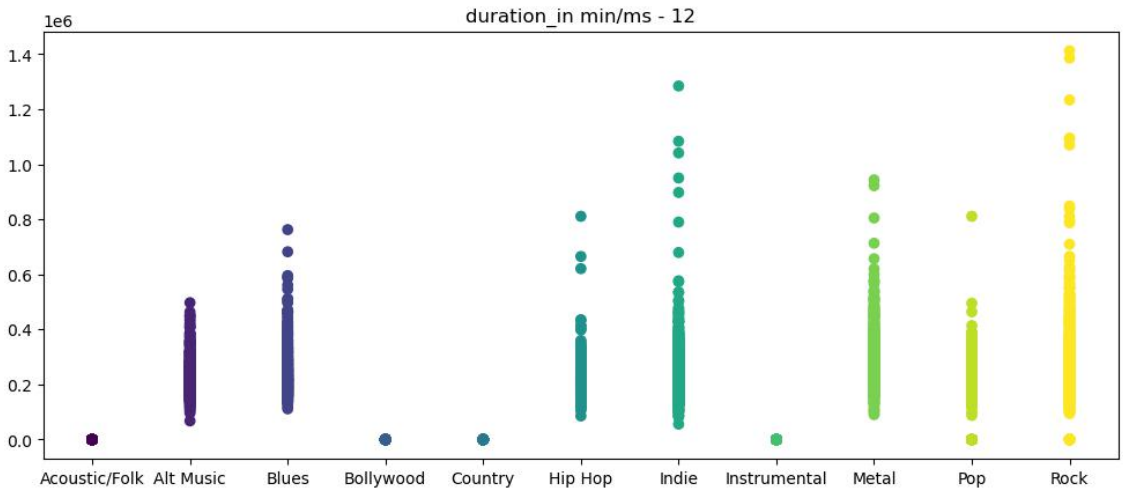

Figure 13: Duration

From these graphs we can see why music is very difficult to differentiate between easily since most classes aren't well differentiated within features. Most of these samples are pretty well distributed and show high variance within a feature.

2.1 Processing the Data

A lot of the data contained values that have non uniform range (eg. Loudness). These values would cause problems during training and prevent the model from training effectively. Therefore, these values were scaled to fit within the range $[0, 1]$.

2.2 Splitting Data into Test, Validation and Training Sets

The data was split into 60:20:20 Training, Test and Validation subdatasets. To ensure that the data was not skewed, the data was randomized before the split.

3 The Model Architecture

I came up with two model architectures to attempt to solve this challenge of music classification.

1. **Model 1** - Three Hidden Layers — 512, 256, 128 nodes
2. **Model 2** - Three Hidden Layers — 1024, 1024, 1024 nodes

3.1 Model 1

As mentioned above, the first model architecture had three hidden layers with 512, 256 and 128 densely connected nodes. These were arranged in a descending fashion with ReLU activation functions at each hidden layer. The output layer consisted of 11 nodes with a Softmax Activation Function.

3.2 Model 2

The second model architecture had three hidden layers with 1024 densely connected nodes each. Each hidden layer also had a ReLU activation function associated with them. The output layer consisted of 11 nodes with a Softmax Activation Function.

For the optimizer, I used Root Mean Square Propagation in both cases. To measure loss, I used Sparse Categorical Cross-entropy and I as a measure of how well the model performed, I used accuracy. Each model had a callbacks object associated with it that saved the model with the highest validation accuracy. The model was then trained for 20 epochs using a batch size of 20.

4 Results and Metrics

From the loss and accuracy figures we can see that both the models overfit very fast.

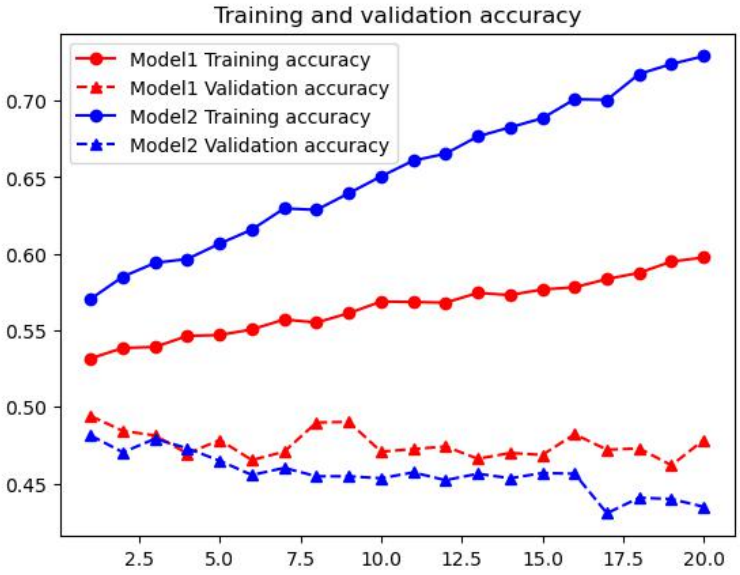

Figure 14: Model Accuracy

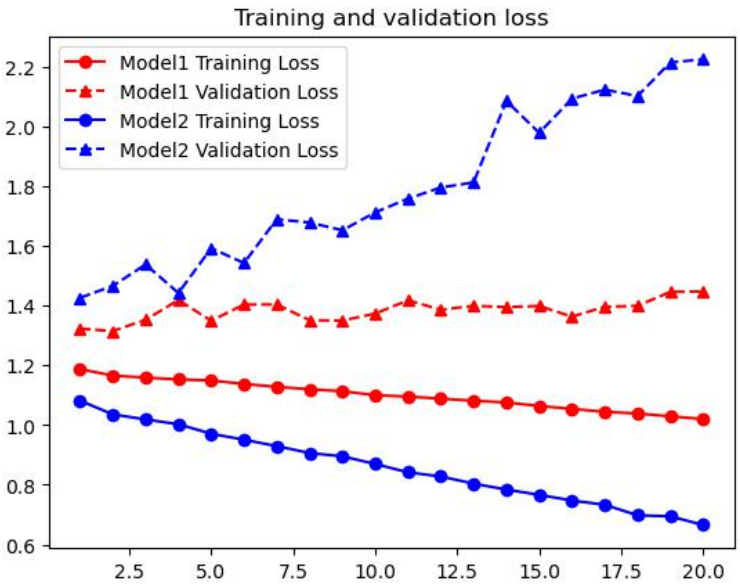

Figure 15: Model Loss

Since we are using validation loss as our metric, the best achieved model has been saved by the callbacks object. So before we test our model on the test data, we need to load the model weights from the saved model.

After doing that, I ran the test data through both the models and the following figures show the obtained confusion matrices for both.

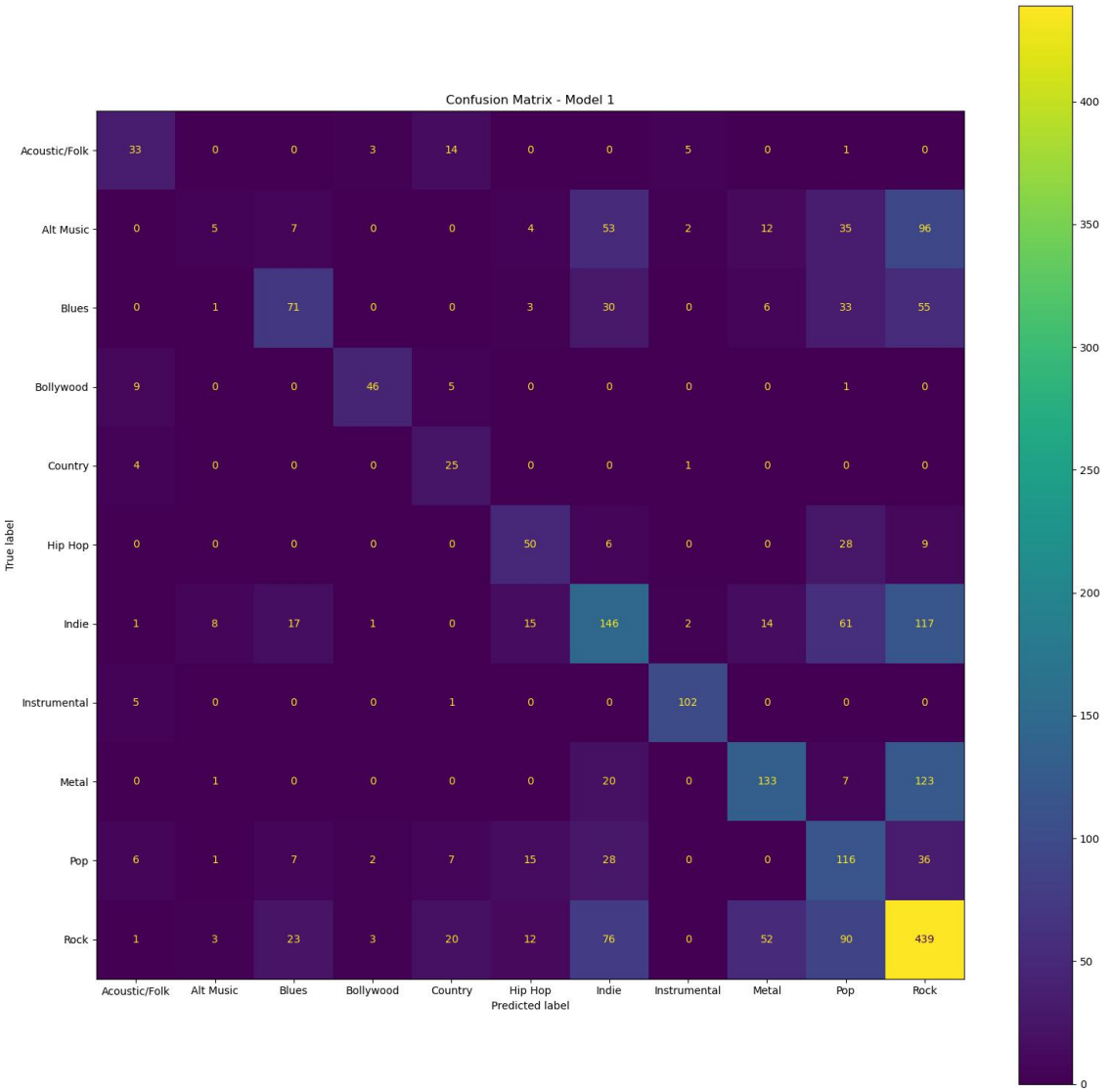

Figure 16: Model 1's Confusion Matrix

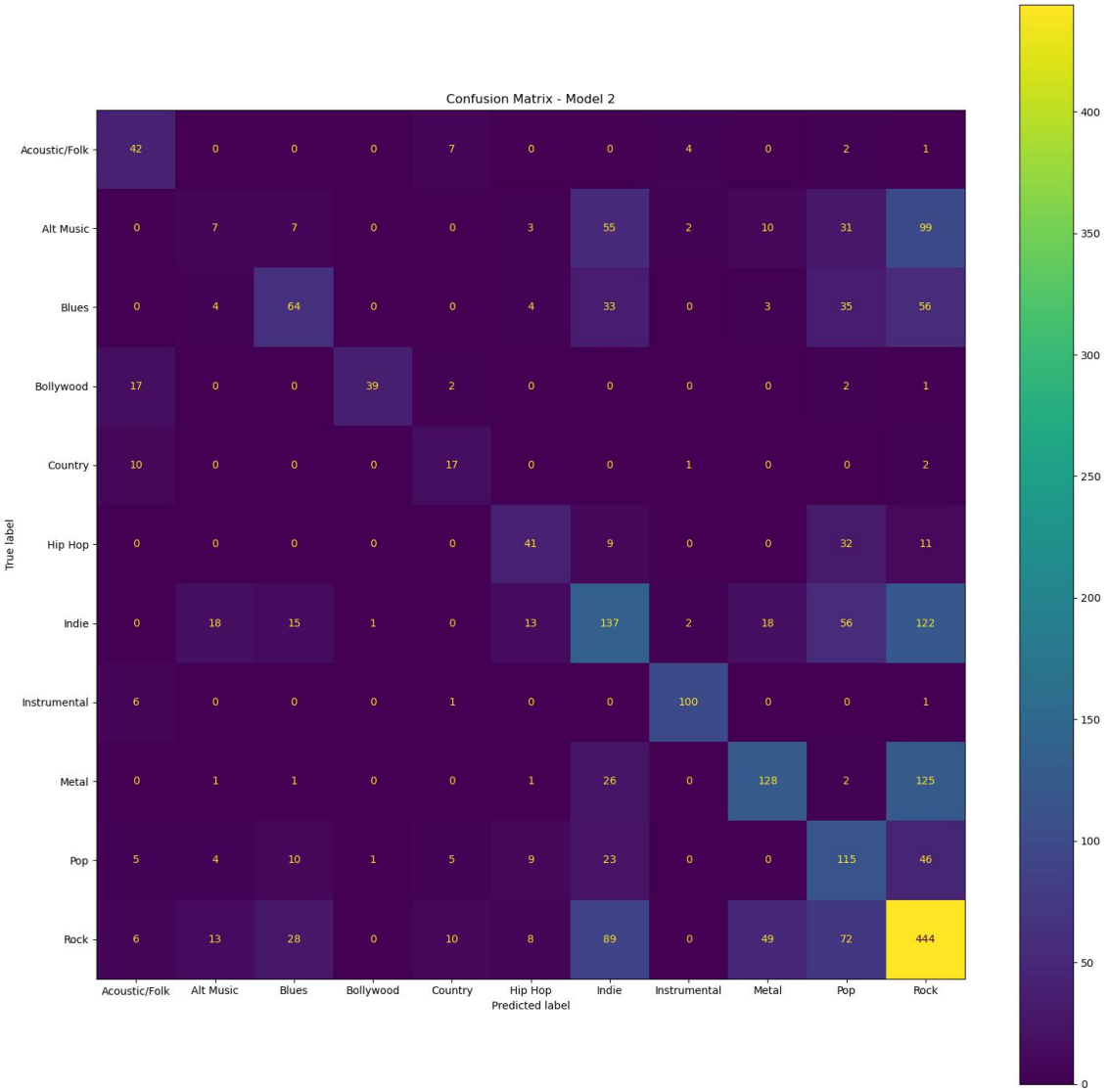

Figure 17: Model 2's Confusion Matrix

From this we see that Model 1 achieved an **accuracy of 49.32%** while Model 2 achieved an **accuracy of 47.97%.** This is quiet appreciable for 11 class classification model, as a base randomizing model would achieve an average accuracy of less than 10%.

5 Conclusion

My biggest takeaway from this project is that multiclass problems with data that's not clearly defined in feature space ,is very difficult to classify effictively. Moreover, I learned that music genres are very subjective, and based on the listeners perspective. Whats interesting is that songs that seem to belong to diametrically opposite genres sometimes show a significant amount of correlation amongst their measurable features.