MUSIC GENRE CLASSIFICATION

EE517 Project Report

Ashwin Telagimathada Ravi
Vignesh Muthuramalingam
Vineeth Rajesh Ellore

Abstract

This project aims to classify music genres using a Multinomial Logistic Classifier and a Multilayer Perceptron(MLP). We have compared multiple approaches including using only time domain features, only frequency domain features and a combined model comprising both time and frequency domains. We have also successfully reduced the number of features of MLP and logistic classifier by half while still maintaining high accuracy.

1. Introduction

Music genre classification has a vast number of applications from recommending music to users in musical apps like Spotify to apps like Shazam which is used to recognize the song from an audio clip. In this project, we classify music genres based on features extracted from time and frequency domains. Specifically, we classify the music into 4 genres: classical, jazz, metal and pop. We use time domain and frequency domain features and compare their performances using multinomial logistic classifier and a multi-layer perceptron model.

2. Dataset

Marsyas (Music Analysis, Retrieval, and Synthesis for Audio Signals) is an open source software framework for audio processing with specific emphasis on Music Information Retrieval Applications. It gives access to the GTZAN dataset [1]. This dataset gained popularity after it was first used in the paper [2]. The files were collected in 2000-2001 from CDs, radio and microphones to represent different recording conditions.The dataset consists of 1000 audio tracks each 30 seconds long. It contains 10 genres, each represented by 100 tracks. We have used the four most distinct genres for this project; classical, jazz, metal and pop.Thus our data set has 400 tracks that are all 22050Hz Mono 16-bit audio files in .wav format.

3. Feature Extraction

The time domain and frequency domain features are extracted from the audio files using librosa [3], a python package for music and audio analysis.

3.1 Time Domain Features

3.1.1 Root Mean Square (RMSE)

RMSE corresponds to the root mean square energy of a signal. RMSE can be calculated as:

$$\sqrt{\frac{1}{N}\sum_{n=1}^{N}|x(n)|^2}$$

For this project, RMSE is calculated framewise for the frame length of 2048 and hop length of 512 and the average of RMSE across all frames is used as the RMSE feature of the audio clip.

3.1.2 Zero Crossing Rate

A zero-crossing is a point where the sign of a mathematical function changes (positive to negative or vice versa). Zero crossing rate is calculated as the total number zero crossings divided by the length of the frame. Frame length of 2048 and hop length of 512 were considered in our analysis. The average of zero crossings across all frames is used as the zero crossing feature of the audio clip, It usually has higher values for highly percussive sounds like those in metal and rock.

$$zcr = \frac{1}{T-1}\sum_{t=1}^{T-1}\mathbb{I}\{s_t s_{t-1} < 0\}$$

Formula to calculate the Zero Crossing Rate

3.1.3 Tempo

Tempo is the speed or pace at which a passage of music is played. It is measured in beats per minute. Similar to the above features, tempo is also calculated framewise for the default frame length of 2048 and hop length of 512, and the average is taken as a representative feature.

Tempo Marking	Definition
Prestissimo	Very Very Fast (>200bpm)
Presto	Very Fast (168-200bpm)
Allegro	Fast (120-168bpm)
Moderato	Moderately (108-120bpm)
Andante	Walking Pace (76-108bpm)
Adagio	Slow and Stately (66-76bpm)
Lento/Largo	Very Slow (40-60bpm)
Grave	Slow and Solemn (20-40bpm)

Figure 1. Name of the tempo marking V/s Beats per minute(bpm)
Source:https://www.musictheoryacademy.com/how-to-read-sheet-music/tempo/

3.2 Frequency Domain Features

3.2.1 Spectral Centroid

Spectral Centroid is a measure which indicates the centre of mass of the spectrum i.e., the frequency around which most of the spectral energy is centered. Spectral centroid can be calculated as:

$$f_c = \frac{\sum_k S(k) f(k)}{\sum_k fk},$$

where S(k) is the spectral magnitude of frequency bin k and f(k) is the frequency corresponding to bin k.

In librosa, each frame of the magnitude spectrum is normalized and is treated as a separate distribution over the interval, from which the centroid is extracted per frame. We again use the default frame length of 2048 and hop length of 512, and average it over all the frames.

3.2.3 Spectral Contrast

Spectral contrast is the difference between the maximum and minimum magnitudes in the frequency domain. Higher values represent narrow-band signals and lower values represent broad-band noise. We use the FFT window size of 2048, hop length of 512, 6 frequency bands and a minimum frequency of 200Hz. We then average it over all time intervals to obtain the representative feature.

3.2.4 Spectral Rolloff

Spectral rolloff is the frequency value below which a given percentage (85% by default) of the total energy in the spectrum lies. Spectral rolloff is given by:

$$\sum_{n-1}^{R_t} M_t[n] = 0.85 * \sum_{n-1}^{N} M_t[n].$$

where Rt is the spectral rolloff frequency and M[n] is the total energy of the audio signal.
We use the default window size of 2048, hop length of 512 and 85% roll percent, and average it over all the intervals.

3.2.5 Mel Frequency Cepstrum Coefficients (MFCC)

MFCC features are based on Short Time Fourier Transforms (STFT). First, STFT is calculated with *n_fft* window size (2048 by default), hop length (512 by default) and a Hann window. Next power spectrum is computed and applied to triangular MEL filter banks. Finally, the discrete cosine transform of the logarithm of all filter energies is calculated to obtain MFCCs. More information about MFCC can be found at [5].
Here we take the first 5 coefficients averaged over all frames.

4. Evaluation

We implement different models for time domain features, frequency domain features, frequency domain features with MFCCs, and all the features combined. We experiment with interaction terms to check their performances. We demonstrate the performance of neural networks for different train-test splits. All our analysis is carried out on IBM SPSS® Statistics.Our analysis considers the relative importance of features, multicollinearity, R-squared values and accuracy of predictions..

4.1 Multinomial Logistic Classification

Multinomial regression is used to explain the relationship between one nominal dependent variable and one or more independent variables. Multinomial logistic classification is a method that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes. It is a model that is used to predict the probabilities of the different possible outcomes of a categorically distributed dependent variable, given a set of independent variables.

We have evaluated four multinomial logistic classifier models, a time domain model, a frequency domain model, combined features model and the significant features model..

Table 1 shows that all three of our models perform better than their respective null models. We observed that the -2*Log Likelihood values of all the models are lower than that of the null model and a value of 0.000 ($p<0.001$) in the significance column indicates that the final model performs better than that of the null model which has only intercept value and considers all other weights to be zero.

Model		-2 Log Likelihood	Chi-square	Degrees of freedom	Sig.
Time domain	Intercept only	1109.035			
	Final	637.626	471.409	9	0.000 (<0.001)
Frequency domain	Intercept only	1109.035			
	Final	108.378	1000.658	24	0.000 (<0.001)
Combined	Intercept only	1109.035			
	Final	48.490	1060.545	33	0.000 (<0.001)

Table 1. Model fitting information

4.1.1 Time Domain Model

The time domain model considers the three time domain features root mean square energy (RMSE), zero crossing rate (ZCR) and tempo.
We observed that the collinearity between the time domain features is minimal. The time domain model achieves a top accuracy of 66.5%. In the adjoining confusion matrix 0,1,2 and 3 corresponds to classical,jazz,metal and pop respectively.

Observed	0.0E+000	1.0E+000	2.0E+000	3.0E+000	Percent Correct
0.0E+000	82	14	3	1	82.0%
1.0E+000	29	55	10	6	55.0%
2.0E+000	3	4	64	29	64.0%
3.0E+000	2	11	22	65	65.0%
Overall Percentage	29.0%	21.0%	24.8%	25.3%	66.5%

Table 2. Confusion matrix for the Time domain model.

4.1.2 Frequency Domain Model

Frequency domain model takes into account 5 MFCC features, spectral centroid, spectral contrast and spectral rolloff. The frequency domain model suffers from multicollinearity problems. Spectral centroid and spectral rolloff have very high collinearity of -0.965 and MFCC1 & MFCC2 have high collinearity of 0.732. However, it still achieves a high accuracy of 95%. In the adjoining confusion matrix, 0,1,2 and 3 corresponds to classical,jazz,metal and pop respectively.

Observed	0.0E+000	1.0E+000	2.0E+000	3.0E+000	Percent Correct
0.0E+000	95	5	0	0	95.0%
1.0E+000	6	90	1	3	90.0%
2.0E+000	0	1	99	0	99.0%
3.0E+000	0	4	0	96	96.0%
Overall Percentage	25.3%	25.0%	25.0%	24.8%	95.0%

Table 3. Confusion matrix for the frequency domain model.

4.1.3 Combined features model

Combined features model includes all the features from both time and frequency domains.

	Predicted				
Observed	0.0E+000	1.0E+000	2.0E+000	3.0E+000	Percent Correct
0.0E+000	96	4	0	0	96.0%
1.0E+000	5	95	0	0	95.0%
2.0E+000	0	0	100	0	100.0%
3.0E+000	0	0	0	100	100.0%
Overall Percentage	25.3%	24.8%	25.0%	25.0%	97.8%

Table 4. Confusion matrix for combined features model.

We found high collinearity between spectral centroid & spectral rolloff, MFCC1 & MFCC2, spectral rolloff and ZCR and spectral centroid and ZCR as shown in figure 2.

Figure 2. Correlation heatmap for the combined features

In terms of accuracy and R-squared values, the combined features model performs the best among the 3 models.

Pseudo R-Square

Cox and Snell	.692
Nagelkerke	.738
McFadden	.425

Pseudo R-Square

Cox and Snell	.918
Nagelkerke	.979
McFadden	.902

Pseudo R-Square

Cox and Snell	.929
Nagelkerke	.991
McFadden	.956

Figure 3. Pseudo R-square values for time domain, frequency domain and combined features model

High pseudo R-squared values for the combined model shows that it performs better than the models with only time domain or frequency domain features. The confusion matrix in table 4 for the combined model shown below proves that it achieves the highest accuracy of 97.8% among the 3 models.

4.1.4 Best features model

The likelihood ratio test of the combined features model shows that not all features are significant. Only the first 4 MFCC features, ZCR, RMSE and tempo have a significance value of 0.000 (less than a p value of 0.001).

Feature	-2 log likelihood	Chi-square	df	Sig.
Intercept	48.703	0.213	3	0.976
Spectral centroid	48.619	0.128	3	0.988
Spectral contrast	51.259	2.768	3	0.429
Rolloff	19.034	0.544	3	0.909
Mfcc2	76.335	27.845	3	0.000
Mfcc1	132.354	83.864	3	0.000
Mfcc3	74.119	25.629	3	0.000
Mfcc4	133.772	85.282	3	0.000
Mfcc5	48.504	0.014	3	1.000
Zcr	56.676	8.185	3	0.042
Rmse	73.543	25.052	3	0.000
tempo	70.485	21.994	3	0.000

Table 5. Likelihood ratio test table for combined features model.

However, due to the high collinearity between MFCC1 and MFCC2, we have only considered MFCC1 for our best model. Hence, our best features model has MFCC1, MFCC3, MFCC4, ZCR, RMSE and tempo as independent variables.
The best features model which only uses about half the number of total features achieves a comparable accuracy of 91.5%.

	Predicted				
Observed	0.0E+000	1.0E+000	1.0E+000	3.0E+000	Percent Correct
0.0E+000	92	8	8	0	92.0%
1.0E+000	12	83	83	4	83.0%
2.0E+000	0	1	1	1	98.0%
3.0E+000	0	5	5	93	93.0%
Overall Percentage	26.0%	24.3%	24.3%	24.5%	91.5%

Table 6. Confusion matrix for the best features model

4.1.5 Forward and Backward Stepwise Regression using Interaction terms

We performed forward and backward stepwise regression using interactions terms with top 4 features MFCC 4, MFCC 3, MFCC 1 and ZCR. We got an accuracy of 94% using forward step regression and 91.1% using backward step regression.

4.2 Multilayer Perceptron

A multilayer perceptron (MLP) is a class of feedforward artificial neural networks (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. It's multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

4.2.1 Classification using Time Domain features

In the first approach, we have experimented with MLP in SPSS using only time domain features such as Zero Crossing rate(ZCR), tempo and root mean square energy(RMSE). We set the minimum number of neurons to 1 and maximum number of neurons to 500. We used gradient descent optimization technique for our MLP. We tried multiple train-test splits such as 60-40, 70-30, 80-20 and 90-10 and multiple learning rates such as 0.4, 0.1 and 0.01 to optimise the model accuracy. We ran each of the variations five times as SPSS split the data randomly and calculated the mean accuracy for each variation.

From the above experiments, we found that a train-test split of 60-40 and learning rate of 0.4 gave the best accuracy of 71.5%.

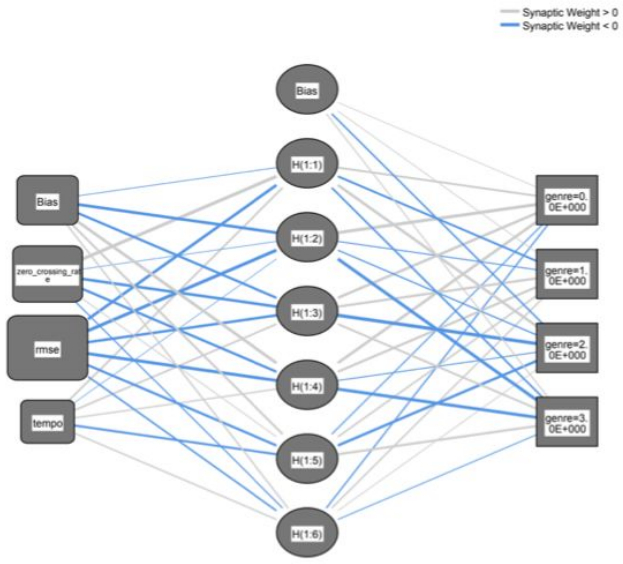

Figure 4. MLP for best time domain model

train-test split	Accuracy
90-10	51.1%
80-20	65.3%
70-30	66.7%
60-40	71.5%

Table 7. Test Train split and accuracy tabulation

4.2.2 Classification using Frequency Domain features

In the second approach, we have experimented with MLP in SPSS using only frequency domain features such as Spectral Centroid, Spectral Contrast, Spectral Rolloff and five Mel Frequency Cepstrum Coefficients(MFCC) . We set the minimum number of neurons to 1 and maximum number of neurons to 500. We used gradient descent optimization technique for our MLP.
We tried multiple train-test splits such as 60-40, 70-30, 80-20 and 90-10 and multiple learning rates such as 0.4, 0.1 and 0.01 to optimise the model accuracy. We ran each of the variations five times as SPSS split the data randomly and calculated the mean accuracy for each variation.

From the above experiments, we found that a train-test split of 90-10 and learning rate of 0.4 gave the best accuracy of 96.4%.

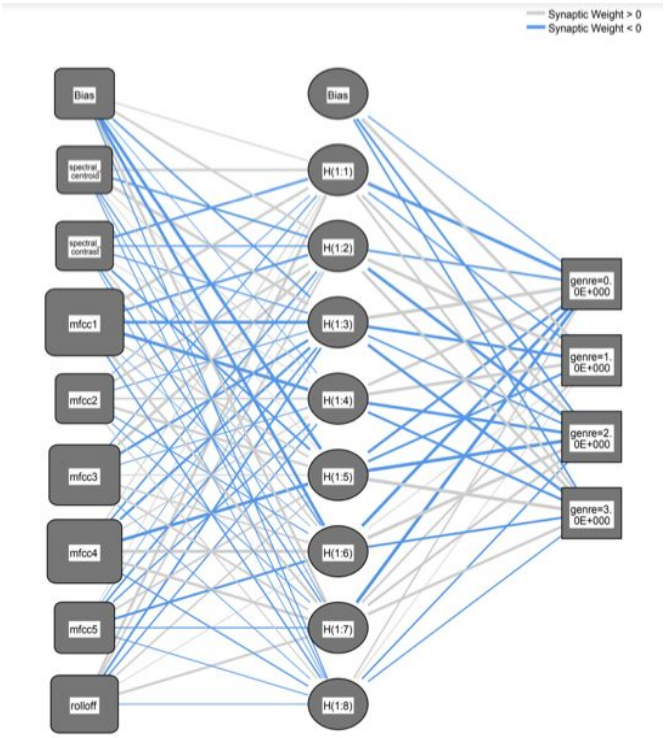

Figure 5: MLP for best Frequency domain model

train-test split	Accuracy
90-10	96.4%
80-20	92.5%
70-30	93.8%
60-40	90.7%

Table 8: Test Train split and accuracy tabulation

4.2.3 Classification using both time domain and frequency domain features

In the third approach, we have experimented with MLP in SPSS using both time and frequency domain features as mentioned above. We have used gradient descent optimization technique for our MLP. We tried multiple train-test splits such as 60-40, 70-30, 80-20 and 90-10 and multiple learning rates such as 0.4, 0.1 and 0.01 to optimise the model accuracy. We ran each of the variations five times as SPSS split the data randomly and calculated the mean accuracy for each variation.

From the above experiments, we found that a train-test split of 90-10 and learning rate of 0.4 gave the best accuracy of 97.9%.

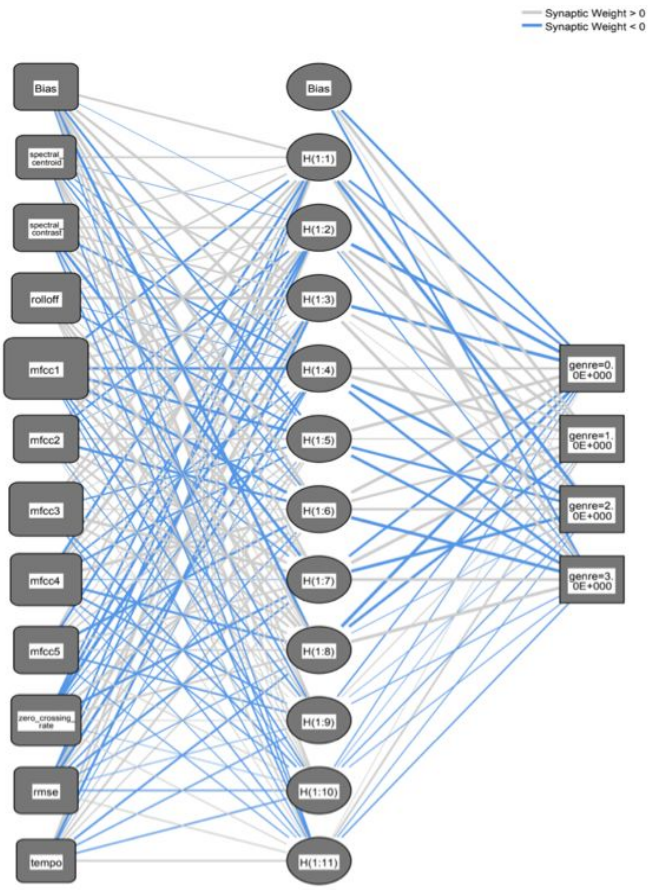

Figure 6: MLP for best Combined features model

MLP train-test split	Accuracy
90-10	97.9%
80-20	95%
70-30	93.6%
60-40	93%

Table 9: Test Train split and accuracy tabulation

4.2.4 MLP Results

MLP train-test split	Time domain	Freq. domain	All features combined	Best 5 features
90-10	51.1%	96.4%	97.9%	95.2%
80-20	65.3%	92.5%	95%	95.9%
70-30	66.7%	93.8%	93.6%	92.4%
60-40	71.5%	90.7%	93%	94.3%

Table 10: Tabulation of experimentation results

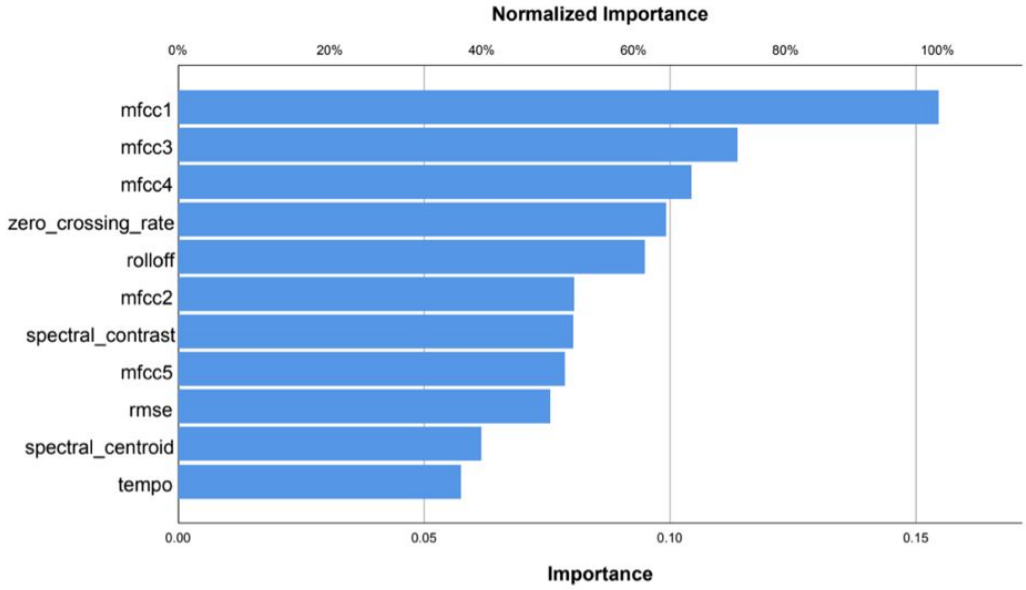

Figure 7: Example of Normalized importance Graph

The above normalized importance graph shows an example for one of the iterations for the best train-test split and learning rate in the combined feature model. We used this graph to extract the most occuring Top six features such as MFCC 4, MFCC 3, MFCC 1, Zero Crossing Rate, RMSE and Rolloff and based on normalized importance.

5. Feature Reduction

We have attempted to reduce the number of features in both MLR and MLP with a marginal decrease in accuracy. In this approach, we have taken the best six features from the combined model as shown in the previous section. In order to predict the number of minimal features required for classification, we experimented by running models with top 3, top 4, top 5 and top 6 features with train-test split of 80-20 and learning rate of 0.4 using gradient descent optimization. We ran each of the variations five times as SPSS split the data randomly and calculated the mean accuracy for each variation.

From the above experiments, we found that using Top five features gave comparable accuracy of 95.3% to the overall combined feature model with accuracy of 97.9%.

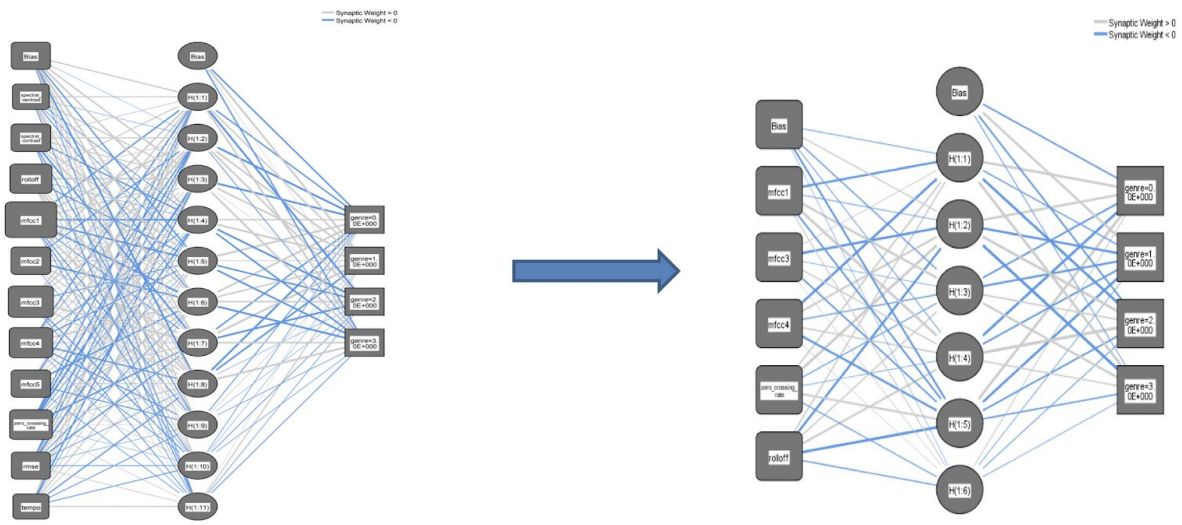

Figure 8: Combined feature model to Reduced complexity model

5. Results

Table 11 shows the percentage accuracy of prediction for all the models we have analysed. As we can see, the frequency domain features give better results than time domain features. The highest accuracy is achieved by the combination of both frequency and time domain features.

Model	Time domain	Freq. domain	All features combined	Best 6 features
Multinomial Logistic Classifier	66.5%	95%	97.8%	93.5%
Multi-layer perceptron	71.5%	93.8%	97.9%	95.9

Table 11: Final Results

6. Conclusion

We used Multinomial Logistic Classification and Multi-layer Perceptron to classify music genres. Frequency domain features performed than time domain features. We have reduced the number of features by half while still maintaining high accuracy.

References

[1] http://marsyas.info/downloads/datasets.html *GTZAN dataset*
[2] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," in IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293-302, July 2002.
[3] https://librosa.github.io/ Librosa python package
[4] Davis, Stan and Paul Mermelstein. “Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Se.” (1980).
[5] https://arxiv.org/pdf/1804.01149.pdf *Hareesh Bahuleyan, “Music Genre Classification using Machine Learning Techniques”*
[6] http://cs229.stanford.edu/proj2018/report/21.pdf