Music Genre Classification

Dmytro Kuzmyk
University of Illinois at Chicago (UIC)
Chicago, IL 60607
dkuzmy3@uic.edu

Charic Farinango Cuervo
University of Illinois at Chicago (UIC)
Chicago, IL 60607
cfarin2@uic.edu

Jieun Yoo
University of Illinois at Chicago (UIC)
Chicago, IL 60607
jyoo49@uic.edu

Members	netID	Question asked to (Group #)	Question answered from (name)
Dmytro Kuzmyk	dkuzmy3	16	Syed Raza
Jieun Yoo	jyoo49	9	Jennifer Alonso
Charic Farinango	cfarin2	7	Only got 2 questions

Abstract

In order to analyze the music files, we used the python *librosa* library to extract the features from each file. We completed it in two processes, one extracting only MFCC for the neural network and the other extracting 26 different features for kNN, Random Forests and SVM. Once we had all the features we needed, we applied them to the models for training and testing. Results were mixed depending on music genre.

1 Introduction/Problem Statement

With the development of online music players, online stores, databases and the always evolving music industry, we sought to build a machine learning system that would automatically classify music tracks based on their genre, given a set of music files.

2 Related Work

Initial work on music classification focused on spectral features. Tzanetakis and Cook's [1] classic work used supervised learning to implement some of the first automatic music classification algorithms. By extracting three sets of features; namely, timbral texture, rhythmic content, and pitch content, they were able to classify a set of ten music genres with 61% accuracy offline and 44% in real time. Ellis [2] continued in that vein and achieved a 57% accuracy rate using a combination of mel-frequency cepstral coefficients (MFCCs) and chroma vectors.

Since Tzanetakis and Cook's seminal work, a variety of machine learning approaches [3, 4] have been applied to music genre classification. Mandel and Ellis [5] extracted MFCCs as song-level features on the *uspop2002* dataset. Their SVM using the Mahalanobis distance achieved 69% accuracy, and their SVM using the KL divergence between single Gaussians achieved an 84% accuracy rate.

Setiadi et. al [6] achieved 80%, 77.18%, and 76.08% accuracy in music classification using SVM, kNN, and Naive Bayes, respectively, on a Spotify music dataset.

In contrast to the above traditional machine learning methods that summarize information into a vector and thus lose temporal information, recent neural network applications take advantage of the temporal structure of audio files. Spectrograms [7], which are 2-D representations of frequency over time have been used as inputs in neural networks. For example, Bisharad and Laskar [8] achieved a 85.36% accuracy rate by training a convolutional neural network on the GTZAN dataset and a 86.06% accuracy rate by training on the MagnaTagATune dataset. Nasrullah and Zhao's [9] Convolutional Recurrent Neural Network (CRNN) achieved a F1 score of 0.937 on the *artist20* music database.

3 Computational Resources

We used the python programming language to access and analyze our data. We used the following data libraries: Sklearn, Numpy, Pandas and TensorFlow [10], and Matplot. For audio processing, we used Librosa [11].

4 Dataset

A variety of open-source datasets are available for music genre classification [12]. We chose the GTZAN dataset, which was the first publicly available music dataset [1]. We downloaded the dataset from the kaggle.com website [13].

The GTZAN dataset consists of 1000 audio tracks, each 30 seconds long. It includes ten genres: blues, classical, country, disco, hiphop, jazz, metal, pop, reggae, and rock.

A 80/20 training/test split was used.

5 Approach

Upon importing the GTZAN dataset, we needed to convert the music files into something a machine could understand, an example of it is a spectrogram.

For the Neural Network, we used only one feature, the MFCC. In order to get the MFCC for each file, we used librosa to convert the files into signals. The signals then were divided into segments so that we could extract the various MFCC for splits of the signal in order to get more accurate labeled data.

For kNN, Random Forests and SVM we wrote python code to extract 26 features: chroma frequencies, spectral rolloff, zero crossing rate, rmse, spectral centroid, spectral bandwidth, and MFCCs.

Descriptions of these features are as follows. Chroma frequencies describe the energy of the pitch class. The spectral rolloff measures the skewness of the power spectrum. It can help distinguish between voiced and unvoiced music. The zero crossing rate is the rate at which the signal changes sign from positive to negative in the time domain. The rms measures signal energies in a time window. The spectral centroid is the weighted mean of sound frequencies. The spectral bandwidth describes the extent of power transfer function around the center frequency.

We extracted 20 MFCCs. MFCCs were originally developed for automatic speech recognition. For our purpose, these describe the shape of the spectral envelope. They are the mathematical coefficients that compose the envelope of the time power spectrum of sound.

5.1 Neural Network

The neural network was built using the TensorFlow.Keras library. The single MFCC feature was used. The data was split into 80/20 train/test sets. The neural network model was built with 4 layers, using Relu for the first three layers and softmax for the last layer. The layers were constructed with the following sets of neurons: 512 for layer 1, 256 for layer two, 218 for layer three, and 10 for the last layer. The learning rate used was 0.001. We trained on 70 epochs with batch size of 64.

5.2 kNN

The kNN classifier looks at the k neighbors of the input and assigns the input to the class with the most number of examples among such neighbors [14]. We performed 10-fold cross-validation to determine the best value for k, which was found to be 7. Results are shown in Figure 1.

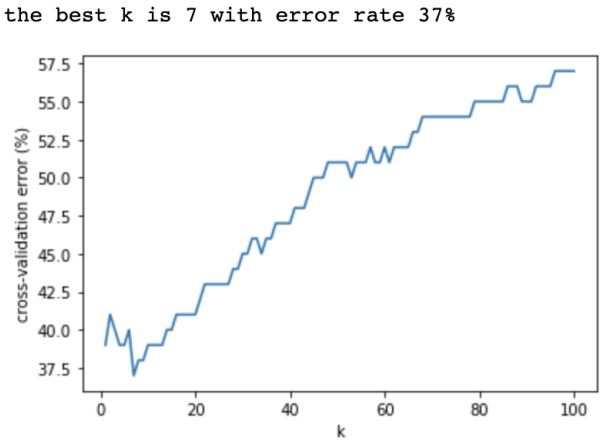

Figure 1: Cross-validation error for kNN

5.3 Random Forests

The method of random forests involves training an ensemble of decision trees on a random subset of input features [14]. We trained 100 trees with a max depth of 5.

5.4 SVM

Support vector machines are a discriminant-based method where we only need to estimate where the class boundaries lie. Ultimately, the parameter of the linear model, the weight vector, can be written down in terms of a subset of the training set, which are the so-called "support vectors" [14]. As the data is probably not linearly separable, as shown in Figure 2, a kernel and a soft margin SVM might be useful.

A grid search hyperparameter optimization was used to find out the type of kernel and the remaining hyperparameter values. For a soft margin SVM with RBF kernel, the possible gamma and C values were [1e-3, 1e-4] and [1, 10, 100, 1000] respectively. For a soft margin with linear kernel the possible values for C were [1, 10, 100, 1000].

After tying all possible combinations, the best estimator has the following characteristics: cost c: 100, kernel: RBF, gamma: 0.01

5.5 Optimization

The first testing on the Neural Network showed a big discrepancy between training accuracy and testing accuracy. See Figure 3.

This is a sign of overfitting. In order to fix overfitting, implemented two strategies: Kernel Regularizers on L2 0.001 and Dropout on 0.2 rate.

The results (Figure 4) still show some overfitting but the error rate is below 5

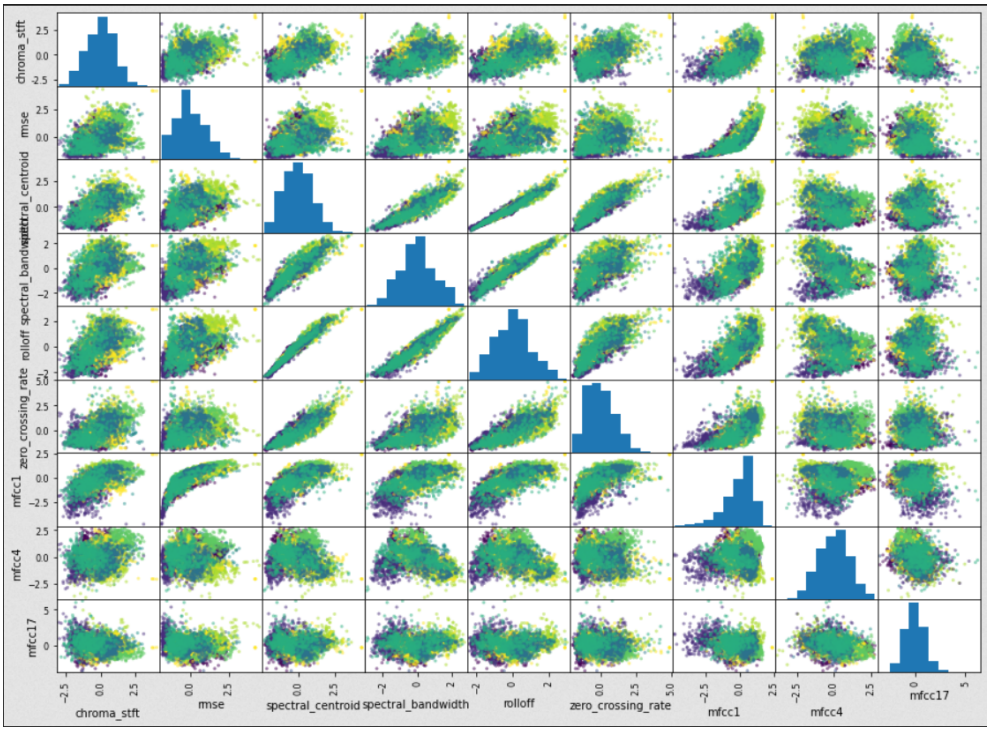

Figure 2: scatter plot for pairs of features

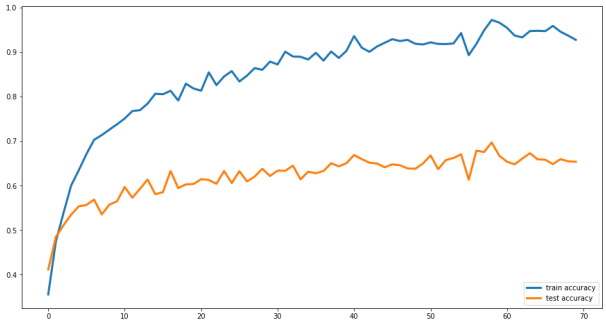

Figure 3: Training vs testing accuracy - overfitted

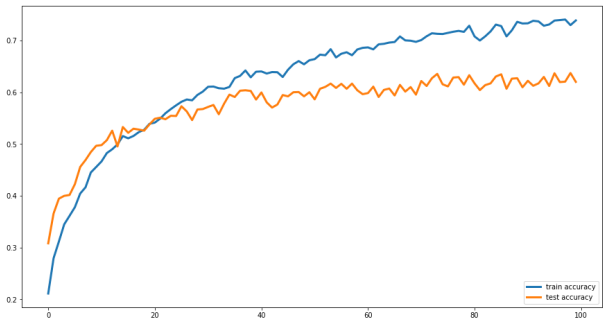

Figure 4: Training vs testing accuracy

6 Evaluation and results

6.1 Performance Metrics

Performance metrics included precision, recall, accuracy, and f1-scores. Accuracy and f1-scores were the most useful. A f1 score is the harmonic mean of precision and recall. High f1 scores result from both high recall and high precision. So, f1 scores can be used to help us compare different classifiers.

6.2 Neural Network

The average accuracy is 62% which is good given that the music files provided are not of the best quality. As we can notice, the accuracy for classical music (label 1) is the highest, this is probably because the classical music files are the cleanest in terms of noise. This music genre also has the most unique and distinguishable features, which makes it easier to classify.

Genre	Label
Blues	0
Classical	1
Country	2
Disco	3
Hip hop	4
Jazz	5
Metal	6
Pop	7
Reggae	8
Rock	9

	precision	recall	f1-score	support
0	0.75	0.84	0.79	235
1	0.86	0.86	0.86	248
2	0.79	0.45	0.57	236
3	0.59	0.58	0.58	244
4	0.62	0.36	0.46	231
5	0.67	0.83	0.74	257
6	0.52	0.42	0.46	230
7	0.59	0.64	0.61	247
8	0.39	0.61	0.48	218
9	0.55	0.57	0.56	252
accuracy			0.62	2398
macro avg	0.63	0.62	0.61	2398
weighted avg	0.63	0.62	0.62	2398

Figure 5: Classification Report for Neural Network

6.3 kNN

kNN had a 60% classification accuracy. The f1-scores for each genre were classical: 0.95, metal: 0.79, pop: 0.71, country: 0.62, blues: 055, jazz: 0.54, hiphop: 0.53, reggae: 0.43, disco: 0.42, and rock: 0.42. We found that the classical music genre had the highest accuracy. Metal and pop also had high classification accuracies. We hypothesize that classical music differed the most in rhythm and beat from the other genres, so it had a high classification accuracy. Reggae, disco, and rock had more similar rhythm and beat, so had lower classification accuracy.

Genre	Label
Blues	0
Classical	1
Country	2
Disco	3
Hip hop	4
Jazz	5
Metal	6
Pop	7
Reggae	8
Rock	9

	precision	recall	f1-score	support
0	0.62	0.50	0.55	16
1	0.91	1.00	0.95	20
2	0.53	0.77	0.62	26
3	0.40	0.44	0.42	18
4	0.57	0.50	0.53	24
5	0.67	0.45	0.54	22
6	0.85	0.73	0.79	15
7	0.67	0.75	0.71	24
8	0.50	0.38	0.43	16
9	0.42	0.42	0.42	19
accuracy			0.60	200
macro avg	0.61	0.59	0.60	200
weighted avg	0.61	0.60	0.60	200

Figure 6: Classification Report for kNN

6.4 Random Forests

Random Forests had a 57% classification accuracy. The f1-scores for each genre were classical: 0.90, metal: 0.76, hip hop: 0.62, jazz: 0.58, pop: 0.68, reggae: 0.57, blues: 0.54, country: 0.41, disco: 0.35, and rock: 0.25. Similar to kNN, classical music had the highest classification accuracy. Similar to kNN, disco and rock had the lowest classification accuracy. We hypothesize this is due to

the fact that classical music had a more distinctive rhythm and beat than other genres, whereas disco and rock were similar in rhythm and beat to each other, so it was harder to distinguish these genres.

Genre	Label
Blues	0
Classical	1
Country	2
Disco	3
Hip hop	4
Jazz	5
Metal	6
Pop	7
Reggae	8
Rock	9

	precision	recall	f1-score	support
0	0.46	0.65	0.54	17
1	1.00	0.83	0.90	23
2	0.33	0.53	0.41	15
3	0.39	0.32	0.35	22
4	0.64	0.60	0.62	15
5	0.61	0.55	0.58	20
6	0.90	0.66	0.76	29
7	0.74	0.48	0.58	29
8	0.52	0.63	0.57	19
9	0.20	0.36	0.26	11
accuracy			0.57	200
macro avg	0.58	0.56	0.56	200
weighted avg	0.63	0.57	0.59	200

Figure 7: Classification Report for Random Forest

The confusion matrix is shown in Figure 8.

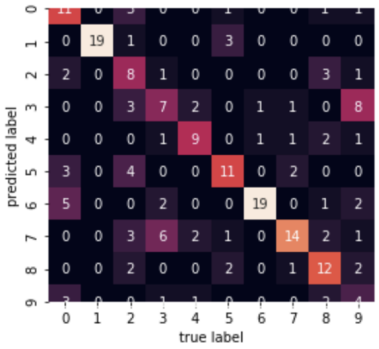

Figure 8: Confusion matrix for Random Forest

Feature importances (Figure 9) were calculated for random forests. They were computed as the mean decrease in impurity within each tree. We found that the most important features were: chroma_stf, rmse, and mfcc4. The least important features were mfcc14 and mfcc20.

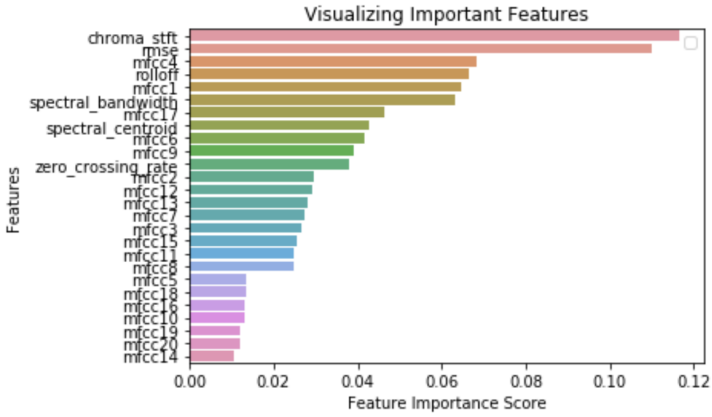

Figure 9: Feature importance scale for Random Forests

The fact that the chroma_stft was the most important feature is not surprising. The property of pitch refers to the ordering of sounds on a frequency scale. Chroma features work by aggregating "all

spectral information that relates to a given pitch class into a single coefficient" [15]. Since pitch is one of the key attributes used to describe music, it made sense that it ranked high in feature importance.

In future implementations of this project, we would refine feature selection. For example, we used 20 MFCCs coefficients, and it would be interesting to further explore results and tradeoffs using different coefficients. For example, we dropped the lowest five features and found that the accuracy dropped one percent to 0.56.

6.5 SVM

SVM achieved an accuracy of 64%. In this case, the F1 score indicates that the top three genres with better performance were 'classical': 0.89, 'metal':0.76, and 'Jazz':0.71. On the other hand, the lowest scores were seen in 'rock': 0.32, 'disco': 0.54 and 'reggae': 0.59. The results remain consistent with the previous classifiers and our hypothesis about more distinguishable genres.

Genre	Label
Blues	0
Classical	1
Country	2
Disco	3
Hip hop	4
Jazz	5
Metal	6
Pop	7
Reggae	8
Rock	9

	precision	recall	f1-score	support
0	0.59	0.72	0.65	18
1	0.80	1.00	0.89	16
2	0.71	0.57	0.63	30
3	0.48	0.61	0.54	23
4	0.62	0.67	0.65	15
5	0.75	0.68	0.71	22
6	0.68	0.87	0.76	15
7	0.61	0.61	0.61	18
8	0.65	0.54	0.59	24
9	0.42	0.26	0.32	19
accuracy			0.64	200
macro avg	0.63	0.65	0.64	200
weighted avg	0.63	0.64	0.63	200

Figure 10: Classification Report for SVM

7 Comparison of results

The accuracy rates of random forests, kNN, neural network, and support vector machines were 57%, 60%, 62%, and 64%, respectively. The support vector machine had the highest accuracy and random forests had the lowest accuracy.

Once we obtained the best and worst classifier, we perform another experiment. As defined before, for each class we had 100 examples per genre. This could be considered a small amount of examples per class. Therefore, we proceed to upsample the dataset taking as a base the original dataset with 1000 examples. After converting the audio files into machine readable signals, we extracted 6 segments in the time domain per audio file. As a result, the final dataset has 5992 examples shown in Figure 11.

label	blues	classical	country	disco	hiphop	jazz	metal	pop	reggae	rock
col_0										
count	600	600	599	600	599	594	600	600	600	600

Figure 11: Upsampled dataset

Once the upsample dataset was obtained, we trained and tested it with random forest and SVM. Grid search was implemented again for SVM. After trying all possible combinations, the best estimator has the following characteristics: c: 1000, kernel: RBF, gamma: 0.001. The results are shown in Figure 12.

Although the random forest results are very similar, the results from the SVM improved. In this case the average accuracy is 80%. The F1 score indicates that the top three genres with better performance were 'classical': 0.91, 'metal': 0.88, and 'pop': 0.88. On the other hand, the worst score is observed for 'disco' and 'rock' with 0.69.

	precision	recall	f1-score	support
0	0.63	0.57	0.60	136
1	0.89	0.81	0.85	121
2	0.27	0.42	0.33	80
3	0.33	0.46	0.39	89
4	0.46	0.44	0.45	120
5	0.59	0.62	0.60	120
6	0.82	0.58	0.68	170
7	0.83	0.55	0.66	183
8	0.52	0.51	0.51	124
9	0.18	0.38	0.25	56
accuracy			0.55	1199
macro avg	0.55	0.53	0.53	1199
weighted avg	0.62	0.55	0.57	1199

	precision	recall	f1-score	support
0	0.78	0.85	0.81	117
1	0.91	0.91	0.91	120
2	0.70	0.75	0.73	124
3	0.67	0.71	0.69	114
4	0.84	0.77	0.80	127
5	0.79	0.83	0.81	108
6	0.87	0.90	0.88	114
7	0.89	0.87	0.88	126
8	0.77	0.75	0.76	125
9	0.76	0.63	0.69	124
accuracy			0.80	1199
macro avg	0.80	0.80	0.80	1199
weighted avg	0.80	0.80	0.80	1199

Figure 12: Classification Report for Random Forest (left) and SVM (right).

As it was consistently observed, genres with distinguished rhythms and beats were easier to classify. We can cluster genres in sets such as ‘similar genres’ and ‘dissimilar genres’ based on this observation. Hence, ‘classical’, ‘metal’ and ‘pop’ would belong to ‘dissimilar genres’ while ‘rock’, ‘disco’, and ‘reggae’ would belong to ‘similar genres’.

As a result of the comparison of different classifiers, particularly a Neural Network, K-NN, Random Forest and SVM, the later performs well in the task of genre classification.

In the future, we might explore a CNN or a RNN. Also, we may want to delve more deeper into similarity measures among different genres as boundaries between genres can be fuzzy at times. For example, a particular genre might be mostly ”rock” but could have elements of ”hip hop.” We would also like to examine additional datasets, perhaps the Spotify dataset. Finally, we might look at additional song features, such as lyrics, sentiments, or album covers as additional ways to aid in music genre classification.

8 Lessons learned

We learned how to use online resources like Librosa and TensorFlow. We learned how to use Librosa to extract features from a dataset, rather than relying on a dataset with previously extracted features. This will help us as we may need to perform both features selection and extraction when analyzing datasets in the future. Further, we learned how to use the popular library TensorFlow to help us with our neural network. Since we did not get a chance to do any labs with neural networks, learning how to use an outside package to implement a neural network will be quite helpful to us in the future.

By using different classifiers we learn how to set and use them for a real application. Moreover, we learned different approaches used for different models such as using an image representation of frequencies to feed the NN. Although, models like random forest do not perform as well as others, it provides others advantages such as an explainable human understandable description of the importance of features. It was interesting to compare them and conclude that older algorithms are still pretty good when compared to more advanced ones like neural networks. Finally, we faced and learned how to deal with over-fitting, one of the most common issues in machine learning.

References

[1] Tzanetakis, G., and P. Cook (2002) “Musical Genre Classification of Audio Signals.” IEEE Transactions on Speech and Audio Processing, vol. 10, no. 5, pp. 293–302. IEEE Xplore, doi:10.1109/TSA.2002.800560.

[2] Ellis, Daniel (2007) “Classifying Music Audio with Timbral and Chroma Features”, Proc. ISMIR-07, pp. 339-340, Vienna, Austria.

[3] Briot, Jean-Pierre, Gaetan Hadjeres, and Francois-David Pachet (2017) “Deep Learning Techniques for Music Generation - A Survey”, arXiv:1709.01620v4 [cs.SD]

[4] Ji, Shulei, et al. (2020) “A Comprehensive Survey on Deep Music Generation: Multi-Level Representations, Algorithms, Evaluations, and Future Directions.” ArXiv E-Prints, vol. 2011, arXiv:2011.06801

[5] Mandel, Michael I., and Daniel P. W. Ellis. (2005) "Song-Level Features and Support Vector Machines for Music Classification, https://www.ee.columbia.edu/ dpwe/pubs/ismir05-svm.pdf

[6] Setiadi, Ignatius Moses, De Rosal, et al. (2020) "Comparison of SVM, KNN, and NB Classifier for Genre Music Classification Based on Metadata." 2020 International Seminar on Application for Technology of Information and Communication (ISemantic), pp. 12–16. IEEE Xplore, doi:10.1109/iSemantic50169.2020.9234199

[7] Wyse, L. (2017) "Audio Spectrogram Representations for Processing with Convolutional Neural Networks", Proceedings of the First International Workshop on Deep Learning and Music Joint with IJCNN, Anchorage, US. 1(1). pp 37-41

[8] Bisharad, Dipjyoti, and Rabul Hussain Laskar (2019) "Music Genre Recognition Using Convolutional Recurrent Neural Network Architecture." Expert Systems, vol. 36, no. 4, 2019, p. e12429. Wiley Online Library, doi:https://doi.org/10.1111/exsy.12429.

[9] Nasrullah, Zain and Yue Zhao (2019) "Music Artist Classification with Convolutional Recurrent Neural Networks", arXiv:1901.04555v2 [cs.SD]

[10] TensorFlow Keras, https://keras.io/guides/

[11] Librosa, https://librosa.org/doc/latest/index.html

[12] Defferrard, Michael, Irell Benzi, Pierre Vandergheynst, and Xavier Bresson (2017) "FMA: A Dataset for Music Analysis," 18th International Society for Music Information Retrieval Conference, Suzhou, China.

[13] GZTAN Dataset, https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification

[14] Alpaydin, Ethem (2020) Introduction to Machine Learning, 4th edition, MIT Press.

[15] Muller, Meinard (2015) "Short-Time Fourier Transform and Chroma Features", International Audio Laboratories Erlangen, https://www.audiolabs-erlangen.de/content/05-fau/professor/00-mueller/02-teaching/2016s$_a pl/LabCourse_S TFT.pdf$