Music Genre Classification

Duties of Group Members

Chen Anqi focuses on business scenario analysis. Wu Jun is responsible for data collection, preprocessing and explorative data analysis (EDA). Huang Zhijun is responsible for machine learning part. Wei Yuhan performs data training on CNN models ResNet18, ResNet34, and ResNe50, Liu Jingwen performs LSTM (Time Series), MLP and MLP with dropout layer, Yan Ge performs xxx model and Guo Xinglin performs xxx model.

Section 1: Executive Summary

As a streaming media, we are faced with a serious problem that misclassification on music genre has harmed user experience and therefore customer churn and profit loss. Worse more, doing music labeling by human labor is time-consuming and expensive. To mitigate this issue, we design an automatic music labeling solution by music feature extraction and deep learning algorithms application. The multilayer perceptron (MLP) with dropout method outperformed all other models in term of processing time and highest accuracy at 93%. Compared to manual labelling, our solution lessens the time for classification by about 6,000 times and reduce the variable cost to almost 0. By using this solution, we finally reduce the cost and improve the efficiency of music genre classification, which is expected to benefit user experience and make profits for our company. These findings also offer practical implications to technology and streaming media practitioners.

Section 2: Introduction

Music genre classification is an important step of music search and recommendation (jiqizhixin, 2021). Misclassification of music genres will result in the low accuracy of search and recommendation and therefore leading to a bad user experience.

There are three kinds of standard practice in real business. The first one is simply classifying music genres according to singers (Sun, 2022). The second approach is to label the music by the music publisher. However, some music publishers deliberately label the audio with a popular but unmatched genre to get more exposure (DeepHub, 2020). Both these two approaches will lead to low accuracy of music labeling. The last approach is to check the music genre classification results manually. Unlike the other data annotation tasks, music genre classification is more challenging. Hence, the common AI-labeling crowdsourcing platform does not have this manual labeling service. And our company must spend a lot of money on training its own data labelers.

Our team aims to develop a deep learning method for solving these two problems mentioned above. Thus, we review some previous published articles. The previous study has applied some deep learning methods to deal with the music genre classification problem. Although popular models like Convolutional Neural Networks (CNNs), ResNet, transfer learning and multi-model approach have been utilized, the model performance is not good, where the best model merely has about 75% accuracy (Khatana, 2020; Rai, 2021; Saint-Felix, 2021). The present study reuses some of these models and make attempts to improve the model performance. Finally, we propose an auto-tagging method using deep learning, which could not only improve the efficiency of music genre labeling but also save time and money for the company.

The rest of this report is organized as follows. In Section 3, we present data collection and preprocessing; In Section 4, we introduce a detailed analysis of three different data transformation methods and diverse algorithms we use to improve the accuracy. In Section 5, we show how our solution benefits our company from the efficiency increasing and cost-reducing aspects. We also present how will we improve our solution by using data enrichment and classification dimension enrichment in the next 6 months.

Section 3: Data

The raw data is GTZAN Genre Classification dataset, which consists of 1,000 audio tracks, each 30 seconds long. It contains 10 genres, each represented by 100 tracks (Tzanetakis & Cook, 2002). The raw data is first transformed into Mel Spectrograms, which is in image form. Afterward, by extracting the music features from original audio tracks and computing the corresponding mean and variance, we get a dataset of 1,000 samples with 57 numerical features. Since most machine learning models and deep learning models prefer larger dataset, each audio track is first to split into 3-second audio files based on the time horizon and Mel Frequency Cepstrum Coefficient (MFCC) are extracted for each segment. In that case, we get the time series data. In addition, after split audio tracks, we further extracted music features and computed the mean and variance to get the well-transformed data (Olteanu, 2020).

The well-transformed data consists of 9990 samples with 57 numerical inputs and 1 label with 10 levels. By performing explorative data analysis, we have 3 findings. First, the means of music features all follow normal distribution. Besides, the variances of most music features are small. Furthermore, the scales of variables are different. For these reasons, we only apply standard scaling on inputs (Appendix 1).

We split the data into 70% training and 30% testing set with stratified shuffle to guarantee they have the same proportion for each level of output.

Section 4: Analysis

The model selection stage is divided into three steps, three different forms of data generated from raw audio and diverse algorithms are used to ensure our models' accuracy.

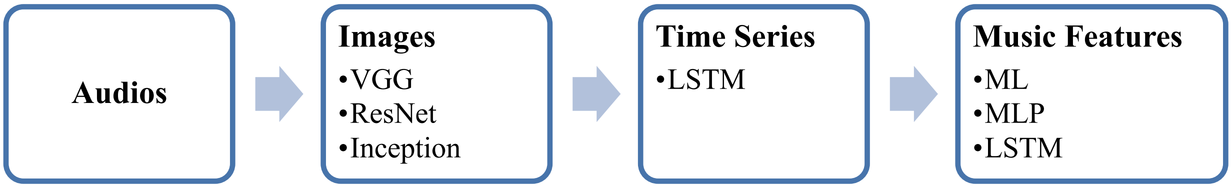

Figure 1 Three Different Forms of Data

As the flow chart shows, firstly, several pre-trained CNNs like VGG and ResNet are utilized to classify the Mel Spectrograms, and ResNet18 achieves the best classification accuracy (0.78) on this Images dataset. However, long short-term memory (LSTM), a specific Recurrent Neural Network (RNN), achieves a slightly higher accuracy (0.80) when trained on the time series dataset, which is transferred from audio data and segmented into 10 sub-arrays.

To further improve the models' accuracy, Music Features extracted from raw audio are exploited and trained with different machine learning and deep learning algorithms. The multilayer perceptron (MLP) with four hidden layers and dropout method outperformed all other models, especially in terms of runtime and accuracy (0.93). Detailed structure and performance of best performed MLP model is shown in table 3 and 4 (Appendix 2, 3).

Table 1 Model Comparison

Dataset	Model	Accuracy

Images	Inception-v3	0.57
	VGG16	0.66
	ResNet18	0.78
Time Series	LSTM	0.80
Music Features	LSTM	0.87
	Light GBM	0.90
	XGBoost	0.90
	MLP (4 layers)	0.91
	MLP (4 layers & dropout)	**0.93**

The model comparison result (Table 1) indicates that compared to images and time series, the music features are a better representation of raw audio data since the classification accuracy of the models trained on music features dataset is much higher than other models, e.g., the accuracy of LSTM on music features is 0.87 whereas on Time Series is 0.80. Besides, given the same music features dataset, both Light Gradient Boosting Machine (Light GBM) and XGBoost give the best result of machine learning methods with the accuracy of 0.90, while our deep learning model, MLP is better than most machine learning models, highlighting the high efficiency, high accuracy, and scalability of our model(Appendix 4).

Section 5: Conclusion

We have demonstrated the applications of multiple deep learning models along with benchmark machine learning models on the music genre classification problem. Although the model we derived is not sophisticated, we believe that such models do have the potential to do more precise classification tasks. Compared to data labelling by human work, an automatic labelling algorithm is significantly superior in time cost and human resources, despite the additional cost of training models and maintaining devices. According to our assumptions (Appendix 5), a mature labelling model could be 15000 times faster than labelling by human, while it is expected to save 2000000 CNY per year for a company. In industries, data labelling is considered an important step, as the quality of data directly influences the quality of its related projects, yet companies tend to rely on interns or outsourcing companies in order to lower the total cost, where mistakes made by human are generally inevitable. A classification model with high accuracy, on the other hand, will be an attractive choice for those in need of such labelling, which will generate more revenue for us. The growing number of clients will also flatten our fixed costs of maintaining computation resources.

To improve the classification quality, in the next 6 months we can mainly focus on two approaches, data enrichment and classification dimension enrichment.

For data enrichment, we already learn from our analysis that, compared to directly dealing with the original music data, extracting and refining representative features from the original data as inputs will give better performance. Currently, we extract 57 features, and we can extract more music features to form a more comprehensive summary of the original data. To scale down the computation and rule out the noise, we can also do feature importance analysis in advance to select the most important and representative features. Meanwhile, on the output side, since currently we only classify music into 10 genres, we can also consider more minor genres to make further elaboration of our classification. For example, rock can be further divided into funk, garage rock, hard rock, etc. One other case we need to consider is that many times one song can belong to multiple genres. To deal with this, we can apply a multi-label deep learning approach.

As for classification dimension, what we've done now is the genre-based classification. However, genre is only one of the many characteristics of a song. Therefore, we can also attach labels to a song according to its other dimensions. For example, by analyzing the lyrics to tell whether it's romantic, inspiring, or it's calling for peace, we can achieve the content-based classification for music.

These are all what we can work on later to enrich and elaborate our job.

Reference

Khatana, A. (2020). *Music genre classification using Transfer learning(pytorch).* Medium. Retrieved from https://medium.com/swlh/music-genre-classification-using-transfer-learning-pytorch-ea1c23e36eb8

Olteanu, A. (2020). *GTZAN dataset - music genre classification*. Kaggle. Retrieved from https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

Rai, S. (2021). *Music genres classification using Deep Learning Techniques.* Analytics Vidhya. Retrieved from https://www.analyticsvidhya.com/blog/2021/06/music-genres-classification-using-deep-learning-techniques/

Saint-Felix, M. (2021). *Music genre detection with Deep Learning.* Medium. Retrieved from https://towardsdatascience.com/music-genre-detection-with-deep-learning-cf89e4cb2ecc

Tzanetakis, G., & Cook, P. (2002). Musical genre classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing, 10*(5), 293–302. https://doi.org/10.1109/tsa.2002.800560

抖音「神曲」那么多，字节跳动是如何玩转亿级曲库的？. 机器之心. (n.d.). Retrieved from https://www.jiqizhixin.com/articles/2021-08-12-5

使用tensorflow进行音乐类型的分类 - 云+社区 - 腾讯云. (n.d.). Retrieved from https://cloud.tencent.com/developer/article/1680670

Appendix

1. Dataset and Explorative Data Analysis

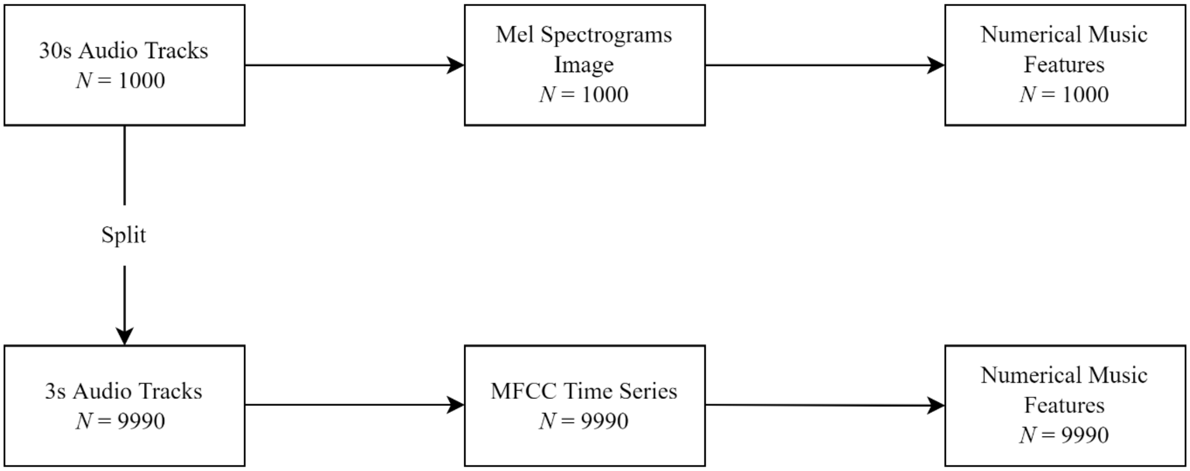

Figure 2 Dataset

2. Descriptive statistics of music features

Table 2 Descriptive Statistics of Music Features

Features	**Mean**	**Std**	**Min**	**Max**	**Range**
chroma_stft_mean	0.38	0.09	0.11	0.75	0.64
chroma_stft_var	0.08	0.01	0.02	0.12	0.11
rms_mean	0.13	0.07	0.00	0.44	0.44
rms_var	0.00	0.00	0.00	0.03	0.03
spectral_centroid_mean	2199.22	751.86	472.74	5432.53	4959.79
spectral_centroid_var	416672.70	434964.44	811.88	4794118.60	4793306.72
spectral_bandwidth_mean	2241.39	543.85	499.16	3708.15	3208.98
spectral_bandwidth_var	118271.11	101350.46	1183.52	1235142.51	1233958.99
rolloff_mean	4566.08	1642.07	658.34	9487.45	8829.11
rolloff_var	1628789.97	1489398.21	1145.10	12983203.77	12982058.67

zero_crossing_rate_mean	0.10	0.05	0.01	0.35	0.33
zero_crossing_rate_var	0.00	0.00	0.00	0.07	0.07
harmony_mean	0.00	0.00	-0.03	0.02	0.04
harmony_var	0.01	0.01	0.00	0.13	0.13
perceptr_mean	0.00	0.00	-0.01	0.01	0.02
perceptr_var	0.01	0.01	0.00	0.06	0.06
tempo	124.89	32.91	24.38	287.11	262.73
mfcc1_mean	-145.42	106.46	-662.17	107.94	770.11
mfcc1_var	2808.90	2596.26	25.19	45027.38	45002.18
mfcc2_mean	100.99	34.67	-12.05	251.21	263.27
mfcc2_var	588.80	459.71	9.67	5131.99	5122.32
mfcc3_mean	-10.00	23.97	-104.25	80.85	185.10
mfcc3_var	374.14	294.47	2.06	4147.79	4145.73
mfcc4_mean	37.24	17.80	-35.14	89.72	124.86
mfcc4_var	183.91	133.16	3.54	2303.75	2300.21
mfcc5_mean	-2.01	13.57	-47.89	46.83	94.72
mfcc5_var	143.82	109.27	9.75	1558.96	1549.21
mfcc6_mean	15.40	12.65	-34.89	54.43	89.32
mfcc6_var	107.78	75.90	5.27	885.97	880.70
mfcc7_mean	-5.82	11.09	-45.19	27.36	72.55
mfcc7_var	98.51	65.54	7.56	672.27	664.70
mfcc8_mean	10.77	11.12	-40.32	65.70	106.03
mfcc8_var	74.80	45.88	6.90	545.36	538.46
mfcc9_mean	-7.57	9.37	-39.45	32.16	71.61

mfcc9_var	74.31	44.73	8.25	421.21	412.96
mfcc10_mean	8.28	8.84	-32.83	58.59	91.42
mfcc10_var	68.80	41.86	7.58	481.92	474.33
mfcc11_mean	-6.50	7.82	-40.01	46.63	86.64
mfcc11_var	63.81	40.22	5.00	691.87	686.87
mfcc12_mean	4.94	7.56	-23.76	51.13	74.89
mfcc12_var	57.79	37.48	2.35	574.70	572.35
mfcc13_mean	-5.19	7.13	-29.35	36.17	65.52
mfcc13_var	57.13	35.75	7.81	571.87	564.06
mfcc14_mean	2.16	6.08	-23.39	34.73	58.12
mfcc14_var	54.07	37.72	3.23	897.63	894.40
mfcc15_mean	-4.18	5.93	-30.47	27.74	58.21
mfcc15_var	52.68	37.25	1.48	621.10	619.61
mfcc16_mean	1.45	5.74	-26.85	39.14	65.99
mfcc16_var	49.99	34.44	1.33	683.93	682.61
mfcc17_mean	-4.20	5.68	-27.81	34.05	61.86
mfcc17_var	51.96	36.40	1.62	529.36	527.74
mfcc18_mean	0.74	5.18	-20.73	36.97	57.70
mfcc18_var	52.49	38.18	3.44	629.73	626.29
mfcc19_mean	-2.50	5.11	-27.45	31.37	58.81
mfcc19_var	54.97	41.59	3.07	1143.23	1140.17
mfcc20_mean	-0.92	5.25	-35.64	34.21	69.85
mfcc20_var	57.32	46.44	0.28	910.47	910.19

3. Structure of best-performed MLP

Table 3 Structure of best-performed MLP

Layer (type)	Output Shape	Parameter
Dense layer (ReLU activation)	(None, 256)	14848
Drop out layer (rate = 0.2)	(None, 256)	0
Dense layer (ReLU activation)	(None, 128)	32896
Drop out layer (rate = 0.2)	(None, 128)	0
Dense layer (ReLU activation)	(None, 64)	8267
Drop out layer (rate = 0.2)	(None, 64)	0
Dense layer (Softmax)	(None, 10)	650

4. MLP model accuracy

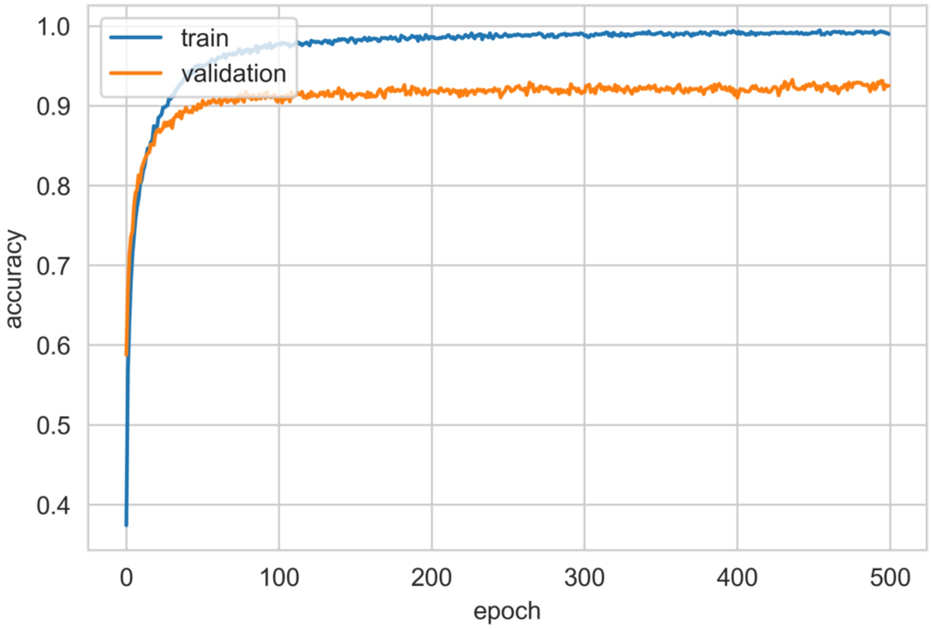

Figure 3 MLP model accuracy

5. Assumptions in Section 5 Conclusion

Table 4 Variable Values of Assumption in Section 5

Relative Variables	Human	Algorithm

Time Cost	**Labeling Rate**	30 seconds per song	500 songs per second
Capital Cost	**Intern or outsourcing company payment**	160 CNY per person per day	--
	Running and maintaining cost	--	800 CNY per day
	Working hours	8 hours per day	--
	Working days	250 per year	250 per year
	Algorithm price	--	1,000,000 CNY
	Total demand	20,000,000 songs	20,000,000 songs

To calculate the time cost, we assume that the algorithm is already deployed, which means that we ignore the time essential for model training. Algorithm is 15000 times faster than human.

To calculate the capital cost, we assume that all labels are classified correctly. The total demand per year is 20,000,000 songs (estimated by Spotify data).

For human labeling, according to the table above, human labeling rate is 30 seconds per song and working hours is 8 hours per day. We can calculate the work load is approximately 1000 songs per day. Hence, the total cost of human labeling can be calculated as follows:

$$20{,}000{,}000 \div (1000 \times 250) \times 160 \times 250 = 3{,}200{,}000 \text{ CNY per year}$$

For algorithm labeling, according to the running and maintaining cost and working days in the table 4, the total cost of algorithm labeling can be calculated as follows:

$$1{,}000{,}000 + 250 \times 800 = 1{,}200{,}000 \text{ CNY per year}$$

Hence, we can find that algorithm labeling saves around 2,000,000 CNY per year.