Music Classification

Wangzitian
1023040922
Nanjing University of Posts and Telecommunications
School of Computer Science
Nanjing, China

***Abstract*—This paper preprocesses the original music dataset, performs feature extraction, and conducts feature engineering to obtain various music features. Simulation experiments are conducted using both traditional machine learning classification models and feedforward neural network models. The study utilizes a dataset comprising 1000 songs from ten different music genres for testing, and achieves favorable results on the test set. Among the selected traditional machine learning methods, XGBoost demonstrates the best performance. As for the feedforward neural network model, after training for 1500 epochs, it achieves an average accuracy of 92.66%. The experimental results affirm the effectiveness of the model.**

***Index Terms*—music classification, FNN, XGBoost**

I. Introduction

With the widespread availability of digital music and the continuous emergence of online music platforms, the demand for music resources from users has been growing steadily. Particularly, in the current era of rapid internet development, digital music has become an indispensable part of people's lives. Musicologists have been devoted to understanding the similarities and differences among various musical works. Music classification contributes to establishing an understanding of musical forms, styles, and genres, aiding scholars in more effectively researching and teaching music.In the field of music classification, the diversity of musical types and forms, along with their inherent differences, makes classifying music an exceedingly complex and challenging task. In 1997, Dannenberg and others attempted automated classification of music genres in WAV format[1]. In 2004, McKay conducted in-depth research on WAV format audio data using machine learning classifiers such as nearest neighbor algorithms and random forests[2]. Currently, mainstream classification methods include both machine learning[3,4] and deep learning[5,6] approaches.

This paper is divided into two parts: data processing and music classification. In the data processing section, we explore various methods used to characterize music features. In the music classification section, we investigate how the classification performance varies under different classification models, aiming to identify the models with better performance.

II. DATA PREPROCESSING

Songs can be analyzed based on their digital signatures for some factors, including tempo, acoustics, energy, etc. In this part, We want to understand what is an Audio file and what features we can visualize on this kind of data.We will use librosa to explore audio data. Librosa is a Python module designed for the analysis of audio signals, with a specific emphasis on music.

We can plot the audio array using librosa.display.waveshow. Here, we depict the amplitude envelope plot of a waveform.

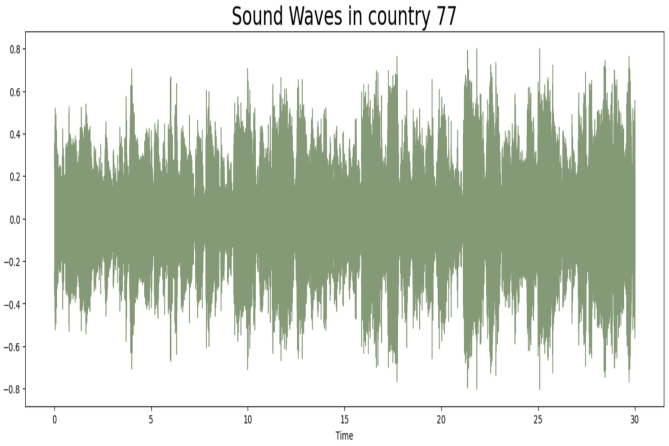

Fig. 1. Amplitude plot of sound.

In the field of audio signal processing, the analysis of acoustic features plays a crucial role in understanding the characteristics of sound. Nevertheless, it is essential to extract the relevant characteristics pertinent to the specific problem at hand. That is to say, we need to extract meaningful features from audio files. We will choose five features to classify our audio clips, i.e., Mel-Frequency Cepstral Coefficients, Spectral Centroid, Zero Crossing Rate, Chroma Frequencies, Spectral Roll-off. All the features are then appended into a .csv file so that classification algorithms can be used.

The first feature we can explore is the Mel frequency cepstral coefficients. A spectrogram is a visual representation of the spectrum of frequencies of a signal as it varies with time. The Mel frequency cepstral coefficients (MFCCs) of a signal constitute a compact set of features, typically numbering around 10 to 20, providing a concise representation of the overall spectral envelope shape. This modeling is designed to capture the characteristics inherent in the human voice.The figure below displays the Mel Frequency Cepstral Coefficients of a country genre song.

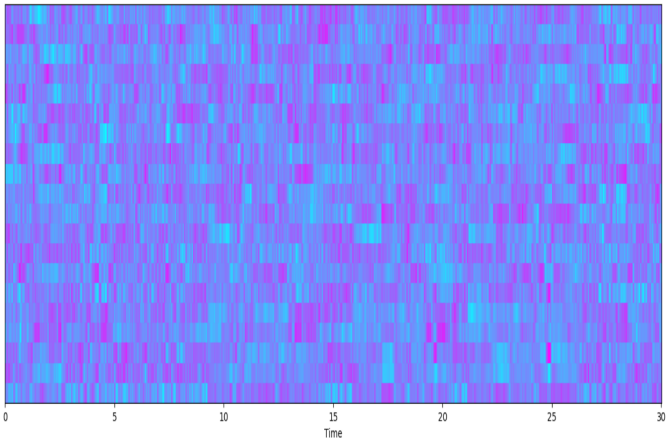

Fig. 2. Mel Frequency Cepstral Coefficients.

The Spectral Centroid is an essential physical parameter in describing timbral attributes and it indicates where the ”center of gravity” of the spectrum is located and provides information about the distribution of frequencies in a signal. It is the frequency at which the energy-weighted average occurs within a certain frequency range, measured in Hertz (Hz). The Spectral Centroid provides crucial information about the frequency and energy distribution of a sound signal. In the realm of subjective perception, it characterizes the brightness of a sound, with darker or deeper qualities associated with more low-frequency content and a relatively lower Spectral Centroid. Conversely, brighter and more cheerful qualities tend to concentrate in higher frequencies, resulting in a relatively higher Spectral Centroid. The figure below plots the Spectral Centroid along the waveform.

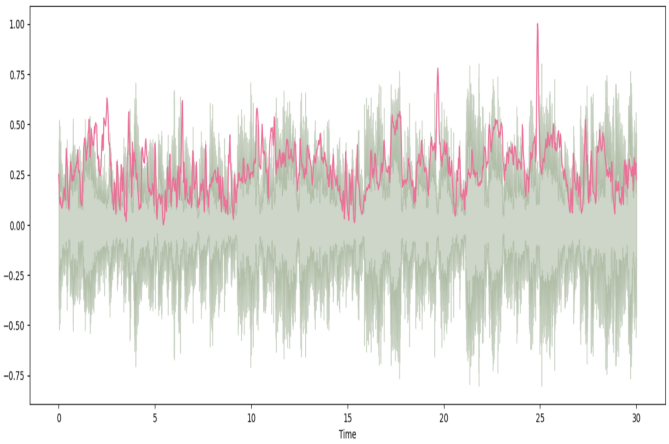

Fig. 3. Spectral Centroid along the waveform.

The Zero Crossing Rate (ZCR) refers to the number of times a speech signal crosses the zero point (changing from positive to negative or from negative to positive) within each frame. This feature has been widely utilized in the fields of speech recognition and music information retrieval, serving as a crucial characteristic for classifying percussive sounds. In the context of audio analysis, it is particularly relevant for understanding the noise or percussive quality of sound. As we can see in the picture below, there appear to be 7 zero crossings and we can verify it with librosa.zero_crossings.

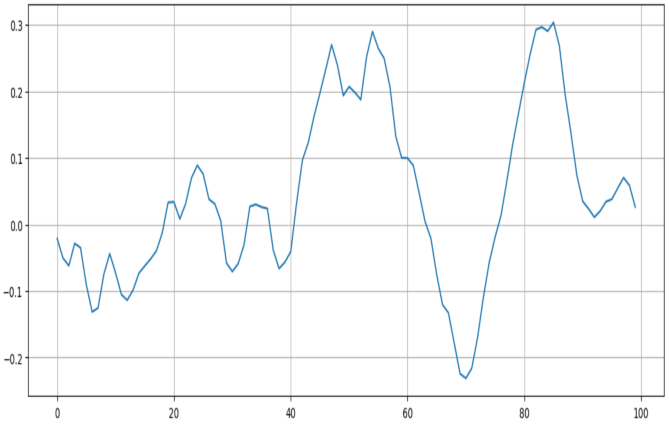

Fig. 4. Zero Crossings.

Chroma frequencies, also known as Chroma feature or Chromagram, are audio features that describe the tonal content of music. They are based on the idea of representing musical notes as colors on a two-dimensional plane, where the vertical axis represents pitch and the horizontal axis represents time.These pitch classes correspond to the twelve different notes in the chromatic scale, which includes all the semitones in an octave. Chroma features are commonly used in music analysis and audio signal processing. The figure below displays the Chroma frequencies of a country genre song.

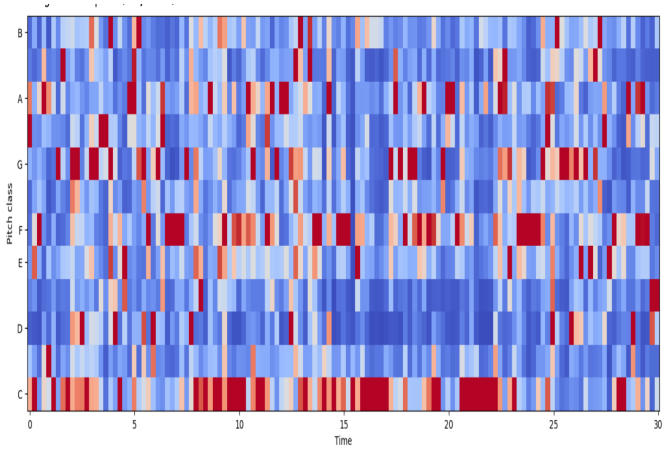

Fig. 5. Chroma frequencies.

Spectral Roll-off is a feature extraction method in audio signal processing. It signifies a specific point in the frequency spectrum of an audio signal, where the preceding frequency components accumulate to a certain proportion of the total energy (typically 85% or 90%). This feature reveals the distribution of spectral energy, holding significance in identifying

various sounds in audio signals, such as the timbre of different instruments. Spectral Roll-off is often combined with other features, such as MFCC, zero-crossing rate, and energy, to enhance the performance of audio processing tasks.The figure below plots the Spectral Roll-off along the waveform.

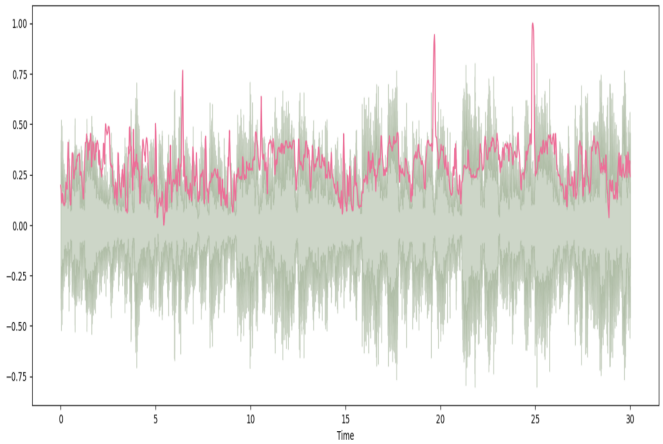

Fig. 6. Spectral Roll-off along the waveform.

III. Solutions

A. Naive Bayes

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem, assuming independence among features. The basic principle of the algorithm is to use Bayes' theorem to calculate the probability of a certain class under given characteristic conditions. For a given set of features, Naive Bayes calculates the probability of each possible category and chooses the category with the highest probability as the final prediction result.

B. k-Nearest Neighbour algorithm

The k-Nearest Neighbors (k-NN) algorithm is a versatile and intuitive approach used for both classification and regression tasks. It operates on the principle of proximity, assigning labels or predicting values based on the majority of k-nearest data points in the feature space. This algorithm is straightforward yet effective, making it suitable for various applications, especially in scenarios where local patterns are crucial. Key components include distance metrics, neighbor selection, and the choice of the parameter k. While k-NN adapts well to different data distributions, it has limitations, such as sensitivity to irrelevant features and computational demands for large datasets.

C. Decision Tree

The decision tree is a graphical model that makes decisions based on a series of sequential choices rooted in feature values, ultimately achieving classification or numerical prediction for instances. The structure of a decision tree includes nodes, branches, and leaf nodes. Each internal node represents a decision based on a feature or attribute, each branch signifies the outcome of the decision, and each leaf node represents a category label (in classification problems) or a numerical value (in regression problems). The construction of the model involves a recursive splitting process, where the optimal feature is chosen to enhance the purity of each branch or reduce the uncertainty of the data. This selection relies on metrics such as information gain, Gini index, measuring the purity or uncertainty of the dataset under a specific feature. Decision trees are characterized by their intuitive interpretability and ease of understanding, coupled with the ability to handle nonlinear relationships.

D. RandomForest

Random Forest is an Ensemble Learning algorithm used primarily for classification and regression tasks. It is based on the idea of decision trees to improve the performance of models by building multiple decision trees and integrating their predictions.The core idea of random forest is to construct multiple decision trees by Bootstrap Sampling of training data and random sampling of features. At each node of the decision tree, the algorithm selects a random subset of features to partition, rather than considering all features. Such randomness helps to reduce the overfitting of the model and improve the generalization ability.In the classification task, the random forest determines the final classification result by majority voting. In the regression task, it takes the average of multiple decision trees as the final prediction result.Random forests are robust, capable of handling high-dimensional data, and insensitive to outliers. In addition, because it can train multiple decision trees in parallel, it also performs well on large-scale data sets.

E. Support Vector Machine

Support Vector Machine(SVM), is a powerful supervised learning algorithm with the core concept of finding a maximum-margin hyperplane in a high-dimensional space to effectively separate data points of different classes. The algorithm's strength lies in emphasizing the maximization of the distance between support vectors on either side of the decision boundary and the hyperplane, thereby enhancing the model's generalization performance. Support Vector Machines can handle not only linear relationships but also model nonlinear relationships by employing various kernel functions, such as polynomial kernels and radial basis function (RBF) kernels. This flexibility allows SVM to effectively capture complex patterns in data.

F. XGBOOST

XGBoost is an efficient machine learning algorithm based on the gradient boosting framework. It iteratively trains multiple weak learners and combines them into a powerful ensemble model. Its key features include utilizing the gradient boosting algorithm to iteratively optimize the model's predictive performance. XGBoost introduces regularization terms such as L1 and L2 regularization to enhance the model's generalization capabilities. It can output feature importance

scores, aiding in the identification of critical features within the model. XGBoost can handle missing values, supports parallel processing to accelerate training, and excels in both classification and regression tasks.

G. Feedforward Neural Network

Artificial Neural Networks (ANNs) have emerged as powerful models for machine learning tasks, demonstrating remarkable success across various domains. Among the diverse architectures of ANNs, the Feedforward Neural Network (FNN) distinguishes itself as a foundational and extensively employed structure. A Feedforward Neural Network is a foundational type of artificial neural network where information flows in one direction — from the input layer to the output layer — without forming cycles. The architecture consists of layers of interconnected nodes, or neurons, organized into an input layer, one or more hidden layers, and an output layer. Each connection between neurons is associated with a weight, and the network learns by adjusting these weights during the training process.

The key characteristics of a Feedforward Neural Network can be described as follows:

- Input Layer: Each neuron in the input layer corresponds to a feature, and the number of neurons is determined by the dimensionality of the input data.
- Hidden Layers: Intermediate layers between the input and output layers are called hidden layers and each neuron in a hidden layer is connected to every neuron in the previous and subsequent layers.The inclusion of hidden layers allows FNNs to learn hierarchical and abstract representations of input data. Each layer captures increasingly complex features, enabling the network to model intricate relationships. The number of hidden layers and neurons in each layer is a critical design consideration, impacting the model's capacity to learn and generalize.
- Activation Function: Activation functions play a crucial role in shaping the behavior of artificial neural networks. They introduce non-linearity, enabling neural networks to learn complex patterns and relationships within data. ReLU is a common activation function, defined as f(x)=max(0,x), and is widely adopted in hidden layers. Its simplicity and ability to mitigate the vanishing gradient problem make it suitable for accelerating convergence in training.
- Weights and Biases: Connections between neurons are characterized by weights, which are adjusted during the training process to learn from data and each neuron has an associated bias term that helps control the neuron's output.
- Output Layer: Neurons in the output layer produce the final predictions or classifications.

The following is a simple feedforward neural network diagram.

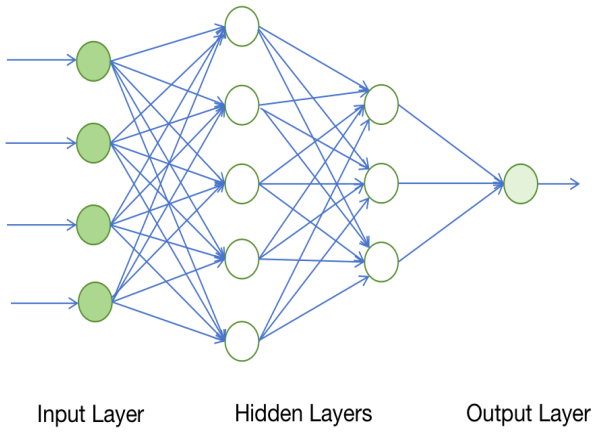

Fig. 7. A simple feedforward neural network diagram.

IV. Related Works

A. Librosa

Librosa is a Python module designed for the analysis of audio signals, with a specific emphasis on music.It provides a wide range of tools and utilities for tasks such as feature extraction, manipulation, and visualization of audio data. Librosa is particularly popular in the fields of signal processing, machine learning, and music information retrieval and it provides the fundamental components necessary for constructing a Music Information Retrieval (MIR) system, offering the essential elements for extracting meaningful information from music data.

B. Scikit Learn

Scikit-learn is an open-source machine learning library for Python, offering a simple and efficient set of tools for data mining and analysis. Built on top of NumPy, SciPy, and Matplotlib, it provides a consistent and user-friendly API, making it a preferred choice for a wide range of machine learning tasks. The library encompasses an extensive collection of machine learning algorithms, including supervised and unsupervised learning, dimensionality reduction, and model selection. With features for robust data preprocessing, feature engineering, and model evaluation, Scikit-learn supports the entire machine learning workflow.

V. MUSIC CLASSIFICATION

A. Datases

The GTZAN[7] dataset stands out as the predominantly utilized public dataset for assessing music genre recognition (MGR) in machine listening research. Compiled between 2000 and 2001, the dataset comprises files sourced from diverse outlets, including personal CDs, radio broadcasts, and microphone recordings. This paper conducts a classification experiment based on this dataset, which contains 10 genres, namely disco, rock-pop, country, metal, jazz, blues, classical, reggae, and hip hop, each with 100 tracks.

B. Experiment Setup

We employed six traditional machine learning methods for music classification, namely Naive Bayes, k-Nearest Neighbors, Decision Tree, Random Forest, Support Vector Machine, and XGBoost.For the Naive Bayes method, the default settings of the GaussianNB function in sklearn.naive_bayes are utilized.For the k-Nearest Neighbors algorithm, the KNeighborsClassifier function from sklearn.neighbors was employed, with the parameter n_neighbors set to 19. For the Decision Tree algorithm, the default settings of the DecisionTreeClassifier function from sklearn.tree were used. In the case of the Random Forest algorithm, the RandomForestClassifier function from sklearn.ensemble was utilized, with the parameters n_estimators set to 1000, max_depth set to 10, and random_state set to 0. For the Support Vector Machine algorithm, the default settings of the SVC function from sklearn.svm were employed. Regarding the XGBoost algorithm, the XGBClassifier function from the xgboost library was used, with the parameters n_estimators set to 1000 and learning_rate set to 0.05.

Additionally, we employed a feedforward neural network model from deep learning for music classification.The feedforward neural network model used in the experiment can be described as follows:

- Input Layer:The input layer consists of Xtrain.shape[1] neurons, corresponding to the number of input features.
- Hidden Layer 1:It is a fully connected layer with 512 neurons and is activated by the Rectified Linear Unit (ReLU) activation function.
- Dropout Layer 1:Randomly drops 20% of the input units to prevent overfitting.
- Hidden Layer 2:It is a fully connected layer with 256 neurons and is activated by the ReLU activation function.
- Dropout Layer 2:Randomly drops 20% of the input units.
- Hidden Layer 3:It is a fully connected layer with 128 neurons and is activated by the ReLU activation function.
- Dropout Layer 3:Randomly drops 20% of the input units.
- Hidden Layer 4:It is a fully connected layer with 64 neurons and activated by the ReLU activation function.
- Dropout Layer 4:Randomly drops 20% of the input units.
- Output Layer:It is a fully connected layer with 10 neurons, suitable for multi-class classification and is activated by the softmax activation function, providing a probability distribution over the output classes.

VI. Result

The classification accuracy of seven different classification methods is shown in the table below.

From the data in the table, we can observe that the classification performance of the Naive Bayes and Decision Tree methods is relatively poor on this dataset. Random Forest outperforms the Decision Tree method, possibly because music classification tasks often involve a large number of features, such as spectral and temporal features. Random Forest demonstrates better handling of high-dimensional data, effectively selecting useful features for the classification task.Among the traditional machine learning methods, XGBoost exhibits the best classification performance with an accuracy of 0.90. Surprisingly, the Feedforward Neural Network achieves an accuracy of 0.92, surpassing XGBoost. This improvement may be attributed to the automatic feature learning capability of deep learning models, allowing them to extract more advanced and abstract feature representations directly from raw data.

TABLE I
Classification Accuracy.

Algorithms	Accuracy
Naive Bayes	0.51952
k-Nearest Neighbors	0.80581
Decision Tree	0.6383
Random Forest	0.81415
Support Vector Machine	0.75409
XGBoost	0.9009
Feedforward Neural Network	0.9266

The following figure visualizes the confusion matrix of the results obtained from the XGBoost method using a heatmap.

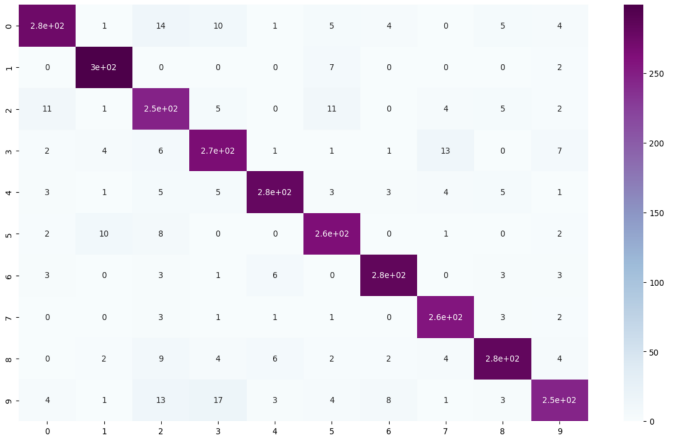

Fig. 8. A heatmap of the confusion matrix.

From the first column of the confusion matrix, it can be observed that there are 280 samples with label 0 that have been correctly predicted, and the number of incorrectly predicted samples is relatively low. In a confusion matrix, larger values on the diagonal elements are desirable, and smaller values elsewhere are preferable. Based on this analysis, it can be concluded that the XGBoost method performs well.

The classification performance of the feedforward neural network model for music is illustrated in Figure 9. The blue line represents the loss function values for the test set, the orange line depicts the accuracy of the test set, the green line reflects the loss function values for the training set, and the red line signifies the accuracy of the training set. As evident in the visual representation, the model demonstrates exceptional performance, showcasing its effectiveness. Notably, there is a notable absence of overfitting or underfitting issues.

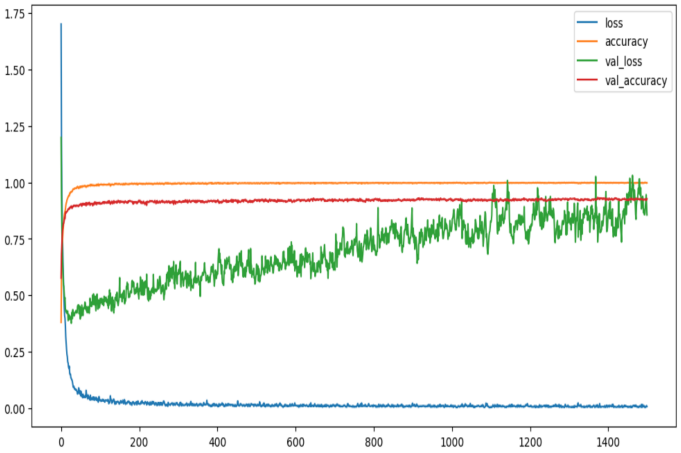

Fig. 9. Classification performance of the feedforward neural network.

VII. Conclusion

This study employed both traditional machine learning and deep learning methods for music classification, aiming to validate the diverse effects of these methods on music categorization. The findings revealed that the Feedforward Neural Network excels in capturing intricate non-linear relationships within music data, showcasing its enhanced flexibility compared to traditional machine learning methods.

Having studied the course on data mining, I have greatly benefited from it. Prof Zou's class is full of excellent experience. Through the course, I gained in-depth understanding of the fundamental principles and methods of data mining. I acquired knowledge about the techniques and tools for extracting valuable information from data, including aspects such as data cleaning, feature selection, and model building. This has provided me with a clearer comprehension of the entire data mining process. Additionally, the data mining course has strengthened my programming and data processing skills. Through assignments and major projects, I mastered some commonly used data mining tools. Most importantly, the course has cultivated my problem-solving and analytical abilities. By continually addressing real-world problems, I have learned how to formulate questions sensibly, choose appropriate methods, and analyze and interpret results.

References

[1] Dannenberg R. B., Thom B., Watson D. A machine learning approach to musical style recognition[C]. International Computer Music Conference, 1997, (1):344-347.

[2] Mc K. C. Automatic genre classification of midi recordings[D]. Canada: Mc Gill University, 2004

[3] Laurier C, Herrera P, Mandel M, et al. Audio music mood classification using support vector machine[J]. MIREX task on Audio Mood Classification, 2007: 2-4.

[4] Wang Q, Xiong Y, Su F. Semantic Music Annotation by Label-Specific Conditional Random Fields[C]//2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018: 2941-2946.

[5] Won M, Ferraro A, Bogdanov D, et al. Evaluation of cnn-based automatic music tagging models[J]. arXiv preprint arXiv:2006.00751, 2020.

[6] Dieleman S, Schrauwen B. End-to-end learning for music audio[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 6964-6968.

[7] Tzanetakis G, Cook P. Musical genre classification of audio signals[J]. IEEE Transactions on speech and audio processing, 2002, 10(5): 293-302.