# Classification of Modern Chinese Music

TianQiu
23009764
04/03/2024

## Introduction

This project collects chart-topping pop music from 1980 to 2020 and divides them into 4 categories, trying to create a machine learning model using different machine learning approaches to automatically classify the music into specific eras. The machine learning methods include KNN (K-Nearest Neighbors) and CNN (Convolutional Neural Network), and their classification accuracies were calculated to compare how good the models were at classifying music between them.

## Background

With the diversification and globalisation of popular music, the status of Chinese popular music seems to be facing challenges. On the one hand, Western pop music and Korean pop music have achieved great success globally, while the influence of Chinese pop music seems to be relatively weakened. And due to the popularity of TikTok, Chinese pop music in the last two years has often been mouth-watering songs originating from TikTok, which have been rated as 'rubbish' and 'hard to listen to' by musicians. I would like to explore whether there is a quantitative indicator for the 'getting worse' that Chinese pop music is said to be, and to explore whether it is possible to categorise pop music from different years well by means of machine learning.

## Method

Firstly, I got the Chinese music list of different years from the official spotify channel, then I got the playlist and the sharing address, I input the address and download the music through the spotify music downloader in python, and categorise them into different folders.

After that we will process the music files, firstly I will convert the mp3 file which is the most common music format to wav file which is better to handle in python, secondly I will split the music files, cut each music file into 10 segments of 5 seconds for subsequent processing. When selecting music segments, I will subtract the beginning and the end of 20s, because generally speaking the beginning and end of the music contains less music information, also more homogeneous, not conducive to classification and analysis. And randomly select ten 5-second clips in the middle. Then I will put half of the music files into another validation folder to verify the accuracy of the model working out.

Then I will convert the music to mono music files and reduce the sample rate a bit, because I found that the input channel of some models can only accept single-channel music, so I will convert the music to single-channel for subsequent processing. And lowering the sample rate will save some of the model calculation time and reduce the model difficulty.

Next I will create a datasheet for my music file with the music's [filename', 'length', 'chroma_stft_mean', 'chroma_stft_var', 'rms_mean', 'rms_var', 'spectral_centroid_mean', 'spectral_centroid_mean', 'spectral_centroid_mean', 'spectral_centroid_mean', 'spectral_centroid_mean', 'spectral_centroid_mean', 'spectral_centroid_var' 'spectral_centroid_var', 'spectral_bandwidth_mean', 'spectral_bandwidth_var', 'rolloff_mean', 'rolloff_var', 'zero_crossing_rate_mean', 'zero_crossing_rate_var', 'harmony_mean', 'harmony_var', 'tempo', 'label'] and export a CSV file for subsequent processing and import.

Next I will use 2 machine learning algorithms to classify the music, KNN and CNN. in CNN I will use two methods directly, one using the PyTorch framework to convolve and pool the data and output the model, and one using the Sequential model from the Keras library with a predefined optimiser (e.g. Adam) and loss functions (e.g. Cross Entropy Loss) and uses advanced APIs provided by Keras, such as the fit function for model training.

Finally I will compare the accuracy between them and draw conclusions.

# Results

## KNN

I imported 75% of the processed music folders into the code and trained them and used 25% of the music files to verify the accuracy of the model we can get a model accuracy of 58.37% for KNN.

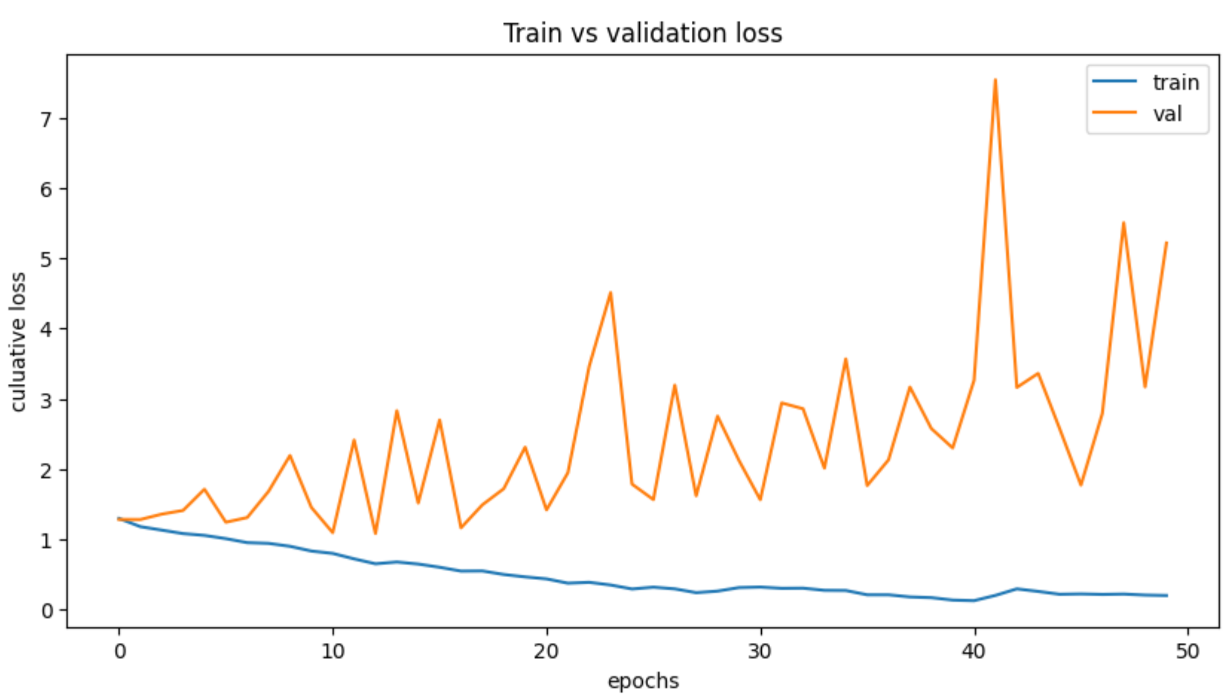
```
# Make the prediction using KNN(K nearest Neighbors)
length = len(testSet)
predictions = []
for x in range(length):
    predictions.append(nearestclass(
        getNeighbors(trainingSet, testSet[x], 5)))

accuracy1 = getAccuracy(testSet, predictions)
print(accuracy1)
```

```
0.5836653386454184
```

## CNN(PyTorch)

We import the processed music folder into the code and train it, and since he can export the current training best model and keep repeating the training, we can then readjust the Train Epoch at the end of each training cycle and observe the degree of change in the TRAIN LOSS and VALIDATION LOSS.

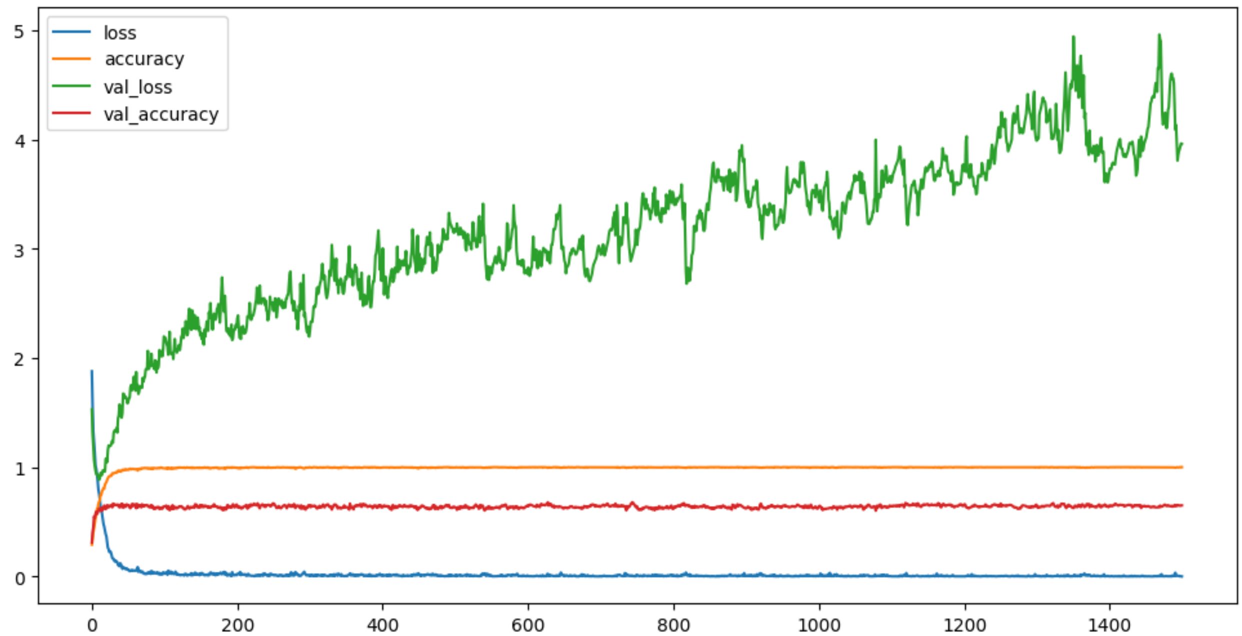


We can see from the figure that the training loss decreases as the epoch increases, but the validation loss fluctuates throughout the training process, especially in the later stages, where some epochs have much higher losses than others. This may indicate that the model is overfitting on the training data, leading to unstable performance on unseen validation data. And as the training progressed, the gap between training loss and validation loss gradually increased and the model may not have generalised well. So I readjusted the learning rate to train again.


We can see little difference in the re-learned images, so we may need to tweak the model and database to optimise. But the model training accuracy increased dramatically. We can find that our training accuracy via CNN (PyTorch) was 47.63% before and 62.41% the second time.

## CNN(Keras)

We imported the processed music datasheet into the code and trained it, in which we used XGBoost (eXtreme Gradient Boosting) for regression and classification of large datasets and also used Adam optimizer to train the model. we can get the plot of the training model's loss, val_ loss, val_accuracy and accuracy of the trained model. And we can finally get Validation Accuracy as 67.89%.




As the number of iterations increases, the loss decreases rapidly and levels off, the accuracy increases with the number of iterations and is close to 100% in the training set at 30 iterations, and we can see from val_loss and val_accuracy_loss that the model performs well on unseen data as well, and on the validation set the generalisation performs well. The model as a whole performs well and shows no obvious signs of overfitting.

## Discussion

In this project, we tried two different machine learning methods to chronologically classify Chinese pop music. The results show that both methods are able to complete the classification task to some extent, but with different levels of accuracy. the KNN model has an accuracy of 58.37%, while the CNN model, after parameter tuning, improves the accuracy to 62.41% using PyTorch and has an accuracy of 67.89% using Keras. This may be due to the fact that the CNN model is more effective in dealing with time and frequency domain features of audio data, especially when dealing with complex patterns regarding the melody and rhythm of the music. And we processed the music data in advance, analysing the multifaceted features of the music and transforming them into data tables may have helped in processing the music data.

Although CNNs show better performance, fluctuations in validation loss suggest that the model may be overfitting the training data. This is usually due to the model being too sensitive to specific noise in the training data, causing it to fail to make accurate predictions on unknown data. This may be due to the relative convergence of the Chinese pop music dataset, which can be addressed by employing diversity, more sophisticated data augmentation methods, or attempting more complex model structures.

We also observe that the validation accuracy stays at 67.89% despite the training accuracy being close to 100%. This is further evidence of overfitting and also hints at the inherent diversity of Chinese pop music, which may require our model to capture more nuanced musical features.

## Conclusion

By exploring and comparing the two machine learning models, KNN and CNN, we can conclude that although KNN is relatively simple and easy to implement when dealing with classification tasks, CNN is more advantageous when dealing with complex audio features.The CNN model demonstrates a high level of accuracy and generalisation ability, and although preliminary experimental results show that the model has a tendency to overfit to a certain extent, the accuracy is improved by adjusting the learning rate and increasing the model complexity, the accuracy was significantly improved. The accuracy was also partially improved by switching to the Keras model and using Adam's optimiser to adjust the learning rate. This provides a valuable reference for our future research on music classification tasks.

Our study also confirms the potential of machine learning models in analysing and understanding popular music trends, especially in analysing the historical lineage of Chinese popular music. The results of this study demonstrate the feasibility and potential of applying machine learning techniques to music chronological classification, which opens the way for intelligent processing of Chinese music and other cultural contents.

## Ethical considerations

1. Data Collection and Usage:
- Ensure appropriate permissions and rights when collecting and using music data from Spotify or other platforms.
- Respect the intellectual property rights of artists and creators.
- Transparently disclose the sources of data and the purposes of its usage.
2. Bias and Fairness:
- Guard against any biases present in the collected data, such as cultural biases or biases introduced by the selection criteria of music charts.
- Ensure fairness in the classification process to avoid misleading or marginalizing certain music genres or artists.
- Regularly assess and mitigate biases in the machine learning algorithms used for classification.
3. Privacy and Consent:
- Protect individual privacy, especially those whose music preferences are included in the dataset.
- Obtain informed consent if personal data is collected or if the study involves human participants.
4. Transparency and Accountability:
- Clearly document the methods used for data collection, preprocessing, and model training to ensure reproducibility.
- Provide explanations for decisions made during the research process, such as feature selection or model architecture.
- Publicly disclose limitations and uncertainties in classification results.
5. Data Security:
- Implement appropriate measures to safeguard the confidentiality and integrity of collected music data.
- Minimize unauthorized access, data breaches, or misuse of sensitive information.

Adhering to these ethical considerations allows for responsible and respectful research into the classification of modern Chinese music, contributing to the advancement of scientific knowledge and fostering cultural understanding.

## LLM disclaimer

In order to be transparent about how you use the Large Language Model (LLM) for code debugging and article translation, the following disclaimer is provided:

Disclaimer:

Tool Use: I use large-scale language models (LLMs), such as ChatGPT, as an aid during code debugging and article translation. These tools are used for inspiration, touch-ups and translations, but should not replace my independent thinking and creativity.

Code Debugging: LLMs were used to assist with code debugging, providing possible solutions, syntax suggestions and programming ideas. The final code is modified by me according to my own understanding and needs.

Article translation: LLM was used to translate article passages, providing initial translation and touch-up suggestions. I reviewed and revised the translations to ensure the accuracy and flow of the final text.

INDEPENDENT WORK: I understand that my independent work should be demonstrated in any submission. Although LLM provides valuable assistance, my creations and decisions are based on personal understanding and the need to make changes.

Avoiding Copying: I have taken care to avoid copying large chunks of LLM-generated code or text in my submissions to ensure my originality. Direct copying of LLM-generated content may result in a lower score or even failure.

EVIDENCE OF ORIGINALITY: I will provide sufficient evidence of originality to demonstrate my independent work in the code and article. This may include code comments, revision notes, independent thought processes, etc.

By providing this disclaimer, I aim to clearly explain how I use LLM and ensure that my work meets standards of integrity and independence.

# Bibliography

[1] Surenjanath Singh. Automating Spotify Playlist Music Download

[ Spotify Free Version ]. Available at:

https://surenjanath.medium.com/automating-spotify-playlist-music-download-spotify-free-version-3ca289bf59f7(Accessed: 14/03/2024)

[2] RAGHAV AGRAWAL. Music Genre Classification USING KNN(Begineers) . Available at: https://www.kaggle.com/code/rxsraghavagrawal/music-genre-classification-using-knn-begineers. (Accessed: 14/03/2024)

[3] PADMAVATHI D. Let's tune🎧the music🎵🎶with CNN🎼XGBoost🎷🎸🎻🎺. Available at: https://www.kaggle.com/code/aishwarya2210/let-s-tune-the-music-with-cnn-xgboost/notebook. (Accessed: 14/03/2024)