

Machine Learning-Based Mood Classification in Music

Weiting Lin, Yuxi Jiang, Hungta Chen, Jaehyeon Park

Abstract

In this paper, our goal is to classify music based on four moods. The rest of the paper is organized as follows: we will first use the API to fetch data from Spotify and perform thorough data preprocessing, including data normalization, outlier removal, and data denoising. After cleaning the data, we will leverage several machine learning methods, such as random forest and deep neural networks, to build models for data classification

1 Introduction

Nowadays, music has become a critical part of our lives, and research indicated that it can be classified based on mood. Many music platforms already utilize mood classification to personalize playlists for their clients. This demonstrates the practicality and relevance of mood classification in the market. Inspired by these developments, we are driven by exploring the potential of efficiently categorizing music using machine learning techniques.

2 Data description

To achieve our goal of efficient music categorization based on mood, we begin by acquiring data through the Spotify API. Our aim is to identify music playlists that correspond to four distinct mood types: anger, relaxation(chill), happiness, and depression. The selection of these mood types is influenced by prior research such as "Music Mood Classification" by Michael Nuzzolo and "What music makes us feel: At least 13 dimensions organize subjective experiences associated with music across different cultures" by Alan S. Cowen.

However, it is important to acknowledge that playlist classifications can sometimes be subjective, with some playlists containing music from multiple mood types. Hence, we assume that each playlist is objective and that all the music within a given playlist belongs to the same mood type. We plan to collect 1000 songs for each mood category in this project and carefully label them accordingly.

There are two types of features that we will use in this project, the first type is continuous features and the second type is categorical features. We will have 9 continuous features (danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo) and 3 categorical

features (key, mode, time signature) leveraged in this project, and the description of features will be placed in the features description link of reference. We aim to construct three distinct datasets for our analysis. These datasets will consist of:

- The first dataset will include only continuous features.
- The second dataset will encompass all available features, regardless of their type (continuous or categorical).
- The third dataset will focus on including only the most relevant and significant features.

By creating these diverse datasets, we can explore the impact and effectiveness of different feature subsets on our analysis.

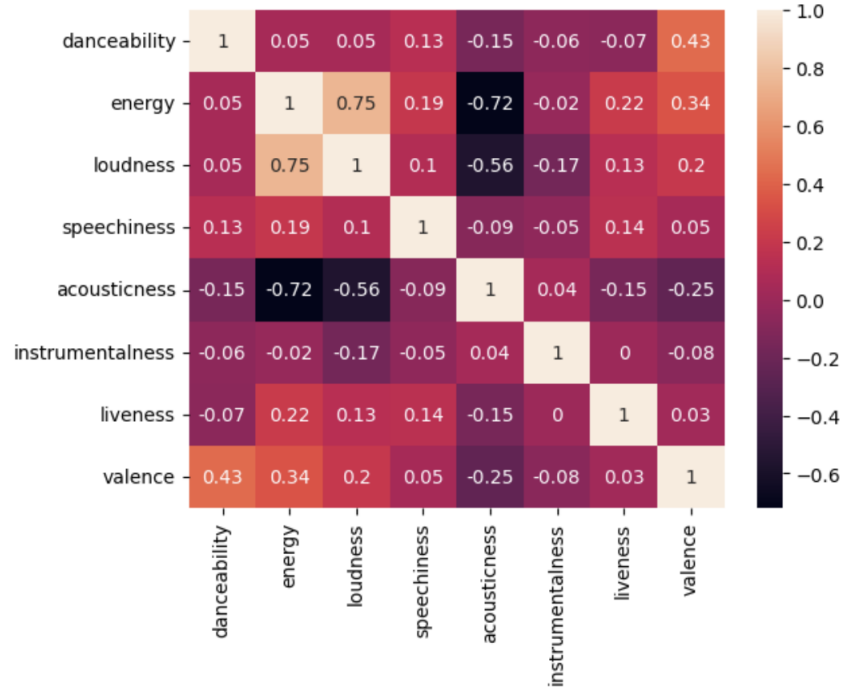


Figure 1: correlation between continuous variables

We will determine third dataset based on the heatmap, we would like to select danceability, energy, loudness, acousticness, valence as the relative critical features due to the higher corresponding correlation.

2.1 Exploratory Data Analysis

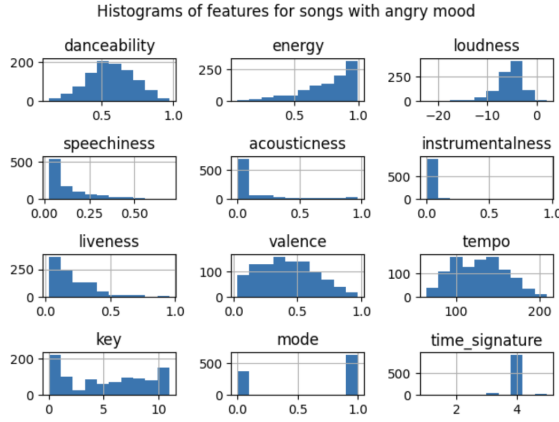


Figure 2: eatures histogram in angry songs

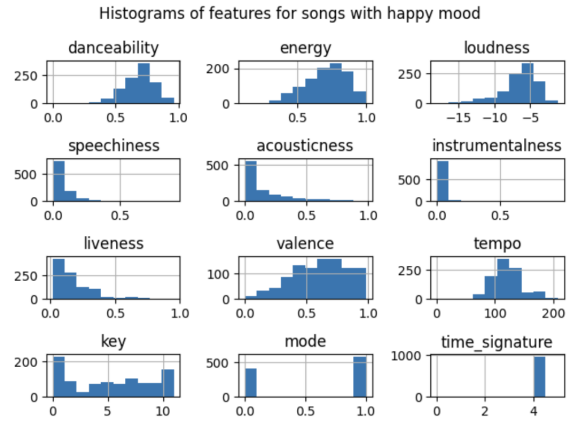


Figure 3: eatures histogram in happy songs

We found that the energy feature in angry songs receives a higher score, which is reasonable considering that this genre of music is often characterized by its emotional intensity. Happy songs exhibit a tendency for the valence feature to approach a value close to 1, signifying their ability to convey a strong sense of musical positivity. Furthermore, happy songs tend to have higher energy features, further contributing to their lively and upbeat nature.

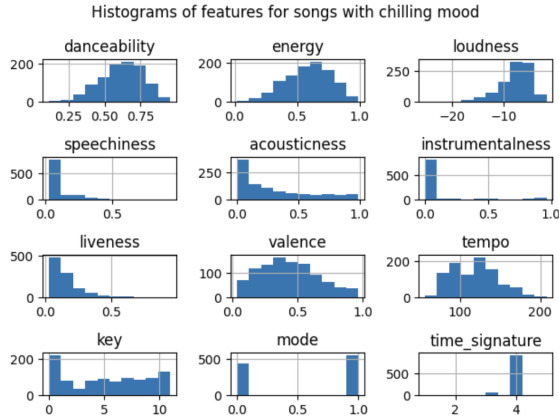


Figure 4: eatures histogram in angry songs

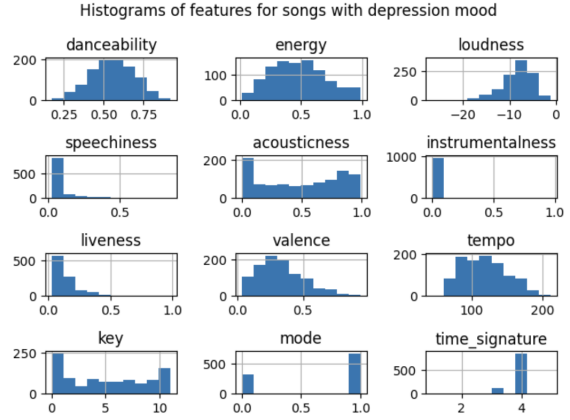


Figure 5: eatures histogram in happy songs

In chill songs, which are commonly known as relaxation songs, there is a higher emphasis on the acousticness feature. This can be attributed to the fact that most chill songs predominantly consist of instruments or white noise, resulting in a greater presence of acoustic elements in the music. In depression songs, the valence, extent of positivity in the track, is opposite to the happy songs. This result is reasonable.

3 Proposed Methods

3.1 Intuition

Our proposed methods for music mood classification, which include Principal Component Analysis (PCA), Support Vector Machines (SVM) with different kernels, decision trees, random forests, Linear Discriminant Analysis (LDA), and deep neural networks, may have several advantages on this task:

1. Feature Extraction with PCA:

PCA helps reduce the dimensionality of the music data while retaining the most relevant information related to mood. By effectively capturing the essential characteristics of the songs, PCA may improve the efficiency and effectiveness of the classification.

2. Ensemble Learning with Random Forest:

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It can handle high-dimensional data and capture complicated relationships between features. By utilizing the power of ensemble learning, we may be able to create a more robust music mood classification model that is less prone to overfitting.

3. Deep Neural Networks for Capturing Complex Patterns:

Deep neural networks have demonstrated exceptional success in many domains, including audio and music processing. They excel at capturing complex patterns and extracting hierarchical representations from the data. By incorporating deep neural networks into our proposed method, we can leverage their capability to capture intricate musical features and potentially enhance the accuracy of mood classification.

3.2 Description of the methods

1. **Principal Component Analysis**

PCA is applied with a variance retention threshold of 95% to reduce the dimensionality of the music data while preserving the most important mood-related features. The resulting transformed data is used for further analysis.

2. **Support Vector Machines**

The proposed method utilizes SVM with different kernels:

- **Linear Kernel:**

A linear kernel is employed to classify the music data into mood categories with a linear decision boundary.

- **Polynomial Kernel:**

The polynomial kernel is utilized to capture nonlinear relationships between features.

- **RBF (Radial Basis Function) Kernel:**

The RBF kernel is used to model complex and nonlinear decision boundaries.

- Sigmoid Kernel:

The sigmoid kernel is employed to capture nonlinear relationships with a sigmoid-shaped decision boundary.

Also, the OneVsOne decision function shape is utilized for multiclass classification, enabling SVM to handle multiple mood categories.

3. Decision Tree

The decision tree algorithm is used to partition the music data based on different features. The decision tree structure allows the model to make decisions at each internal node based on specific features, leading to the classification of music into different mood categories.

4. Random Forest

The random forest model is an ensemble learning method, combining multiple decision trees, where each tree is trained on a random subset of features. By aggregating the predictions of individual trees, the random forest produces the final prediction. This approach helps to improve the accuracy and robustness of the music mood classification by reducing overfitting and capturing complex relationships in the data.

5. Linear Discriminant Analysis

LDA is utilized to reduce the dimensionality of the music data while enhancing the separability between different mood categories. By finding a linear combination of features that maximizes the separation between classes, LDA contributes to improved accuracy in music mood classification.

6. Deep Neural Networks

We proposed a deep neural network with several hidden layers. Our model has the following architecture:

- Input Layer: The input layer is designed to have the same number of units as the features in the music data.
- Hidden Layers: We used Multiple hidden layers with 80, 20, 20, 10, and 10 units respectively. Dropout regularization with a rate of 0.2 is applied to prevent overfitting.
- Output Layer: The output layer consists of 4 units, corresponding to the number of mood categories. The softmax activation function is used to generate probabilities for each category.

As for the activation function, ReLU function is used in the hidden layers to introduce non-linearity and capture complex patterns in the music data.

Also, we explore various optimizers, including Adadelta, Adagrad, Adam, RMSprop, and SGD, to train the DNN model for better performance and convergence of the model. The model is trained using categorical cross-entropy loss and accuracy as the evaluation metric.

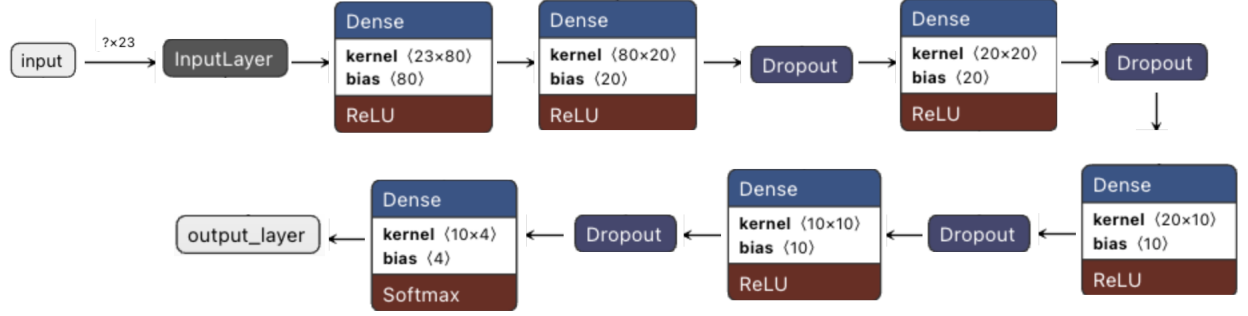


Figure 6: Architecture of DNN model

4 Results

In this project, we are mainly interested in the following aspects of this dataset: 1. The distribution and summary statistics of the song features('danceability', 'energy', 'key', 'loudness',...) in each playlist associated with one of the four moods(angry, chilling, happy, depression). We'd like to know whether there is a significant distinction among features of songs with different moods. 2. We will explore the possible correlations among numerical features of songs to detect any potential relationships between certain variables. 3. The last, and the most important task of this project is to implement models to classify songs with mood labels and find the model that has the best performance.

We first grouped the dataset based on the moods and computed the summary statistics of all the song features. The results are mostly as expected. 'Happy' songs have the highest danceability and valence on average. 'Angry' songs have the highest energy, loudness, speechiness, liveness, tempo and lowest acousticness on average. 'Chilling' songs have the highest instrumentalness on average. 'Depression' songs have the lowest danceability, energy, loudness, speechiness, instrumentalness, liveness, valence, tempo and highest acousticness and duration_ms. The key, mode and time_signature distributions of songs with all moods don't differ much. In addition, we found a strong negative correlation between the acousticness and energy feature, and a moderate negative correlation between the acousticness and loudness feature of songs.

We then ran our models on the training and test data. For the SVM model in particular, we split the features into continuous and categorical subsets and fit the model separately. The result in Table 1 shows that, for continuous data, using the rbf kernel gives us the highest accuracy rate(0.546667) while the sigmoid kernel gives us the lowest accuracy rate(0.396667). For categorical data, using linear kernel gives us the highest accuracy rate(0.2825) while the sigmoid kernel gives us the lowest accuracy rate(0.246667). We also ran the model on a selected smaller subset of the continuous features("danceability", "energy", "loudness", "acousticness", "valence"). Again, using the rbf kernel gives the highest accuracy rate(0.53) while the sigmoid kernel gives the lowest accuracy rate(0.405). Then, we fit Decision Tree, Random Forest and Linear Discriminant Analysis models

Table 1: Accuracy rates of SVM model with different kernal functions/datasets

	continuous features	categorical features	selected continuous features
linear	0.531667	0.282500	0.500000
ploy	0.517500	0.272500	0.505000
rbf	0.546667	0.267500	0.530000
sigmoid	0.396667	0.246667	0.405000

Table 2: Accuracy rates of DNN model with different optimizers/datasets

	Adadelta	Adagrad	Adam	RMSprop	SGD
continuous features with pca	0.515000	0.493800	0.503800	0.373800	0.513800
continuous features without pca	0.507500	0.522500	0.442500	0.420000	0.227500
selected combined features with pca	0.475000	0.503800	0.455000	0.515000	0.480000
selected combined features without pca	0.475000	0.503800	0.455000	0.515000	0.480000
combined features with pca	0.480000	0.496300	0.477500	0.451200	0.493800
combined features without pca	0.437500	0.261200	0.242500	0.392500	0.401300

on the combined training data. The result indicates that the Random Forest model performs the best(0.575). Decision Tree model has a slightly worse performance(0.433333)

Lastly, we implemented the Deep Neural Network model. We first split the song features into continuous and categorical subsets again, and performed principal component analysis to reduce the dimensionality of the input data and increase computation efficiency. Then, we implemented the Deep Neural Network model with 'ReLU' as activation function. According to the result in Table 2, for continuous data with PCA, 'Adadelta' optimizer gives us the highest accuracy rate(0.5150) while 'RMSprop' gives us the lowest accuracy rate(0.373800). We then repeat the procedure on the original continuous data. Again, 'Adadelta' optimizer gives us the highest accuracy rate(0.546200) while 'Adam' gives us the lowest accuracy rate(0.515000). We then test the model on the selected subset of features. 'RMSprop' optimizer gives us the highest accuracy rate(0.515000) while 'Adam' gives us the lowest accuracy rate(0.455000). For the combined data with pca, 'Adagrad' optimizer gives us the highest accuracy rate(0.496300) while 'RMSprop' gives us the lowest accuracy rate(0.451200). For the combined data without pca, 'Adadelta' optimizer gives us the highest accuracy rate(0.437500) while 'SGD' gives us the lowest accuracy rate(0.242500).

5 Conclusion

In this study, we investigated the classification of music based on mood using various machine learning techniques. Our goal was to explore the potential of efficiently categorizing music and contributing to developing personalized music recommendation systems.

Through data preprocessing, including normalization, outlier removal, and denoising, we prepared the dataset obtained from the Spotify API. Subsequently, we employed different classification methods, including kernel-based Support Vector Machines (SVM), Deep Neural Networks (DNN),

Decision Trees, Random Forests, and Linear Discriminant Analysis.

Our results revealed that while not all methods achieved high accuracy, specific approaches performed relatively better. The SVM model, specifically with the 'rbf' kernel, demonstrated the highest accuracy for continuous data, while the 'linear' kernel yielded the best results for categorical data. Additionally, the random forest model outperformed the decision tree model. In the case of the DNN model, the 'Adadelata' optimizer exhibited the highest accuracy for both PCA-reduced and original continuous data.

In conclusion, our study highlights the challenges and potential of music mood classification using machine learning techniques. The SVM model with the 'rbf' kernel and the random forest model emerged as promising approaches in accurately categorizing music based on mood. These findings contribute to the advancement of personalized music recommendation systems, benefiting users in discovering music that aligns with their desired mood. Future research can focus on further improving classification accuracy, exploring additional features, and considering the interpretability of the models for a more comprehensive understanding of music mood classification.

6 Discussion

One of the significant limitations of our study was the overall lower accuracy than anticipated, despite specific approaches demonstrating relatively higher accuracy. Upon reflection, the main factor that significantly impacted our results was the potential overlap in classification due to subjective interpretations when categorizing music based on mood. As music classification involves subjective elements, it becomes challenging to establish precise definitions. Although we utilized pre-classified mood playlists from Spotify, these playlists likely contained overlapping elements. Moreover, we proceeded with the project assuming perfect classification, contributing to the lower overall accuracy. We are confident that utilizing more accurately classified playlists would have yielded higher accuracy in our study.

Reference

1. Web API — Spotify for Developers. (n.d.-a). <https://developer.spotify.com/documentation/web-api>
2. Netron. (2023, June 10). <https://netron.app/>
3. Brownlee, J. (2022, August 2). Tensorflow 2 tutorial: Get started in deep learning with tf.keras. MachineLearningMastery.com. <https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/>
4. Features description. <https://developer.spotify.com/documentation/web-api/reference/get-several-audio-features>