# Machine Learning Approaches to Classifying Music into Moods

Eric Han Yang Chen

Under the Supervision of Professor Yue Li

Department of Computer Science

McGill University

April 28, 2023

## 0. Abstract

*Music has a strong effect on people's mood. We believe machine learning could help classify music into moods. In this research project we leveraged Spotify's API to obtain audio features and used K-means clustering and neural networks in both unsupervised and supervised learning to classify them into four mood categories: happy, sad, energetic, and calm. We found that with an accuracy score of 74%, the neural network outperformed K-means clustering. This study contributes to the ongoing research on mood-based music classification and has potential applications in personalized music recommendation systems.*


## 1. Introduction

### 1.1 Background information

In recent years, advancements in machine learning have aimed to enhance our problem-solving capabilities, enabling us to tackle new challenges. While machine learning has demonstrated remarkable accuracy in tasks involving objective and impersonal data, it has also shown promising results in tasks related to human emotions. One such domain is music. The widespread adoption of streaming platforms like Spotify has significantly increased the volume and variety of music available, leading to a growing demand for more sophisticated music recommendation and classification algorithms.

Traditionally, recommender systems have relied on collaborative filtering, which compares users' tastes and recommends similar music to users with overlapping listening histories. However, in recent years, Spotify and other streaming platforms have begun to leverage machine learning tools, such as TensorFlow, to build AI-powered music recommendation systems. These systems incorporate a deeper understanding of music properties to provide more tailored listening experiences.

In 2020, the Spotify API started providing more detailed audio features, such as danceability, energy, and liveliness. These features offer valuable insights into the intrinsic qualities of a song, allowing for the development of algorithms that can analyze and classify music based on various dimensions, including mood. By incorporating these features into machine learning models, researchers and developers can create more nuanced recommendation systems that cater to listeners' emotional states and preferences, enhancing the overall music discovery experience for users.

### 1.2 Research project overview

This research project aims to develop a mood classification model for songs available on Spotify by employing machine learning techniques, such as neural networks and K-means clustering. We will utilize the Spotify API to access audio features and metadata to perform this task.

The primary objective is to categorize songs based on low-level audio features into four mood categories: happy, energetic, sad, and calm. The success of the proposed model will be evaluated by its ability to accurately classify songs into these mood categories and its potential application in creating mood-based playlists. The unique challenge in this project arises from the

subjective nature of music. Identifying suitable evaluation metrics for the model will also be a challenge, as there is currently no universally accepted method for categorizing music by mood.

The remainder of this report is structured as follows: Section 2 provides a brief overview of related work in the field of music mood classification and machine learning techniques that will be used in this project. Section 3 outlines the methodology used in this project, detailing the data collection process, preprocessing steps, and the implementation of neural networks and K-means clustering algorithms. Section 4 presents the results of the mood classification model and evaluates its performance using various metrics, including K-fold validation accuracy and Adjusted Rand Int. Finally, Section 5 summarizes the key findings and discusses the possible applications and future directions for the work.

## 2. Related work and tools

### 2.1 Prior publications about mood

Mood classification has emerged as a prominent area of research within the Music Information Retrieval (MIR) community, garnering significant interest among scholars and practitioners alike. A key development in this field is the Valence-Arousal model, a two-dimensional framework for mood representation proposed by Thayer in 1989. This model delineates mood along two orthogonal axes: valence, which ranges from negative to positive, and arousal, which spans from calm to energetic. The intersection of these axes creates four quadrants that correspond to distinct mood categories, providing a foundation for mood-based analysis and classification.

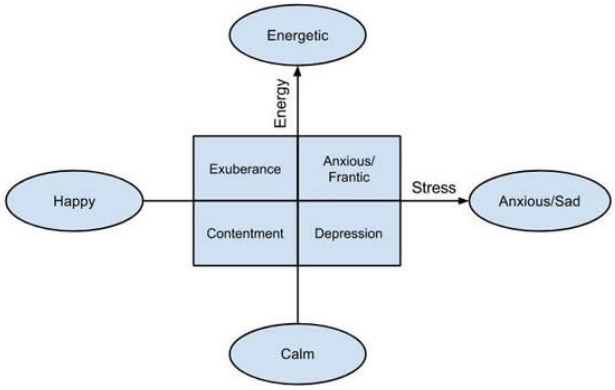


*Figure 1 The Thayer model of music to mood relationship (Nuzzolo 2015).*

Building on Thayer's Valence-Arousal model, numerous studies have employed this framework to categorize songs into the four quadrants, which closely align with the mood categories of the present project: happy, energetic, sad, and calm. For instance, Bhat (2014) utilized this approach to investigate the relationships between musical features and mood, shedding light on the potential of machine learning algorithms for mood-based song classification.

By incorporating the Valence-Arousal model into the analysis, this undergraduate research project seeks to advance the understanding of mood classification in the context of MIR, while also exploring the efficacy of various machine learning techniques in identifying and categorizing songs based on their mood-related attributes. The findings of this study could contribute to the development of more sophisticated mood-based playlist generation systems.

## 2.2 Feature selection and extraction

Feature extraction is a crucial step in the mood classification process. Acoustic analysis of audio signals has demonstrated in previous studies that low-level audio features, such as tempo, energy, valence, and acousticness, can significantly influence our perception of a song's mood (Nuzzolo, 2015). These features, derived from various aspects of the audio signal, help capture the underlying characteristics that contribute to the emotional impact of the music.

Tempo, for instance, refers to the speed or pace of a song, typically measured in beats per minute (BPM), and has been linked to emotions such as excitement and relaxation. Energy, on the other hand, represents the overall intensity and activity level of a song, with higher values indicating more energetic or upbeat tracks. Valence is a measure of the music's positivity or negativity, where higher valence corresponds to happier or more positive songs, while lower valence is associated with sadder or more negative tracks. Acousticness quantifies the degree to which a song sounds acoustic, as opposed to electronic.

In this research project, we will utilize the Spotify API to access a comprehensive set of audio features, encompassing, but not limited to, the aforementioned attributes. The Spotify API provides a rich source of data for a wide variety of songs, making it an ideal choice for obtaining and analyzing these audio features for mood classification tasks. By leveraging these features, we aim to develop more accurate and effective mood classification models, ultimately contributing to a better understanding of the relationship between music and emotions.

## 2.3 Machine learning approaches for mood classification

Various machine learning techniques have been employed for music mood classification, encompassing both supervised and unsupervised methods. Unsupervised approaches, such as K-means clustering, are utilized to explore the underlying structure of music data and identify mood-related patterns. In the unsupervised learning portion of this project, we have chosen K-means clustering because of its primary advantage: the ability to discover patterns in data without requiring labeled information.

Among supervised methods, Support Vector Machines (SVMs), Random Forests (RFs), and traditional Neural Networks (NNs) have demonstrated promising results. In recent years, deep learning models such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have gained popularity due to their capacity to capture complex patterns and structures in data. To maintain a manageable project scope, we will implement both a neural network-based model and a K-means clustering algorithm to classify songs into the four mood categories: happy, energetic, sad, and calm.

For this project, we have selected TensorFlow and its built-in Keras library as the primary machine learning tools. TensorFlow, an open-source machine learning library developed by Google, enables users to build, train, and deploy a wide variety of neural networks and machine learning models. It is highly flexible, offering support for multiple languages, including Python, which is the language used in this project. Keras, initially developed as a user-friendly interface for building deep learning models, is a high-level neural networks API. Since TensorFlow 2.0, Keras has been integrated

as its official high-level API, providing an intuitive and simplified way to design, train, and evaluate neural networks. This has made it a popular choice among developers and researchers for rapid prototyping and experimentation.

# 3. Methodology

## 3.1 Research design

The research design of this project comprises two primary components: unsupervised learning and supervised learning. Acquiring labeled data poses a challenge for this project, as datasets with mood annotations are scarce. Typically, song libraries are categorized by genre, and the few playlists labeled by mood either contain a limited number of songs or lack up-to-date music. As this challenge is likely to persist in future work, we will investigate the advantages and disadvantages of both unsupervised and supervised learning approaches.

In this study, we employ K-means clustering for both unsupervised and supervised learning, enabling an intriguing comparison between the two techniques. We have chosen to implement K-means clustering for both learning approaches to maintain consistency in our comparison. We have also incorporated a neural network model for supervised learning to benefit from its advanced capabilities and to provide a more comprehensive analysis of the potential of different machine learning methods in mood-based classification of songs.

## 3.2 Data collection

### 3.2.1 Unsupervised learning data

One of the benefits of utilizing unsupervised learning in this project is the ability to amass a vast collection of unlabeled songs directly from the Spotify API, eliminating the need for manual labeling. To facilitate this process, the Spotipy library was employed to import songs from the top 100 albums across various genres on Spotify. This approach resulted in a comprehensive dataset comprising over 5,300 relevant and popular songs.

The dataset was subsequently loaded into a Pandas DataFrame for easy manipulation and analysis. A CSV file containing this DataFrame is included as a supplementary material to this report. During the preprocessing phase, the audio feature values were scaled (normalized) to ensure that all features contributed equally to the model's learning process, preventing any feature from dominating the others due to its scale.

Dimensionality reduction was performed using an autoencoder, which is a type of artificial neural network capable of learning lower-dimensional representations of high-dimensional input data. This process helps reduce noise and computational complexity while retaining the essential information from the original data. The reduced dimension was one of the parameters varied in this study to evaluate the impact on the performance of the clustering algorithms.

By conducting a thorough investigation of unsupervised learning techniques, this undergraduate research project aims to provide valuable insights into the potential of machine learning for music mood classification, ultimately contributing to the development of mood-based playlist generation systems.

### 3.2.2 Supervised learning data

To obtain a labeled dataset of songs, we utilized mood-labeled playlists from Michael Moschitto's GitHub repository, which also dealt with a project involving AI and music. The playlist consists of over 1,700 songs, providing a sufficiently large sample to achieve satisfactory results. However, it is important to note that the size and popularity of the songs in this dataset cannot be compared to that of the unsupervised dataset used in this study.

The preprocessing stage involved normalizing the audio features to ensure all features have the same scale, which is crucial for the efficient training of machine learning models. Additionally, the mood labels were one-hot encoded, a process that converts categorical variables into a binary matrix representation, making it suitable for machine learning algorithms.

To split the dataset into training and testing sets, we employed the train_test_split method from the sklearn.model_selection library. Based on our experimentation, an 80% training and 20% testing split provided the best balance between the amount of data available for training the model and the data reserved for evaluating its performance.

## 3.3 Data analysis

## 3.3.1 K-means clustering

For both the unsupervised and supervised learning components of this research project, the KMeans algorithm from the sklearn.cluster library was employed to perform the clustering tasks.

To assess the performance of the K-means clustering in the unsupervised learning phase, three evaluation metrics were utilized: primarily the Silhouette score, Davies-Bouldin index, and Calinski-Harabasz index. The Silhouette score measures the similarity between instances within a cluster compared to instances in neighboring clusters, with higher values indicating better clustering. The silhouette score is calculated for each data point by considering the following:

$a(i)$: The average distance between the data point i and all other data points within its assigned cluster. This measures the cohesion or how well the point fits within its cluster.

$b(i)$: The minimum average distance between the data point i and all data points in any other cluster, excluding the cluster to which the point is assigned. This measures the separation or how well the point is separated from other clusters.

The silhouette score for data point i, s(i), is calculated as:

$$s(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))}$$

The overall silhouette score for the entire clustering is calculated by averaging the silhouette scores of all data points.

Interpretation of the silhouette score:

Values close to 1: The data point fits well within its cluster and is well separated from other clusters. This indicates a good clustering.

Values close to 0: The data point lies on or near the boundary between two clusters. This suggests that the clustering may be ambiguous or not well-defined.

Values close to -1: The data point may have been assigned to the wrong cluster, as it is closer to data points in a different cluster than to those in its own cluster. This indicates a poor clustering.

The Davies-Bouldin index evaluates the ratio between within-cluster dispersion and between-cluster separation, with lower values suggesting better clustering and lastly, the Calinski-Harabasz index measures the ratio of between-cluster dispersion to within-cluster dispersion, where higher values indicate better-defined clusters.

In the supervised learning phase, the performance of the K-means clustering was evaluated using the Adjusted Rand Index (ARI) metric. The ARI measures the similarity between the predicted clustering and the true labels, taking into account both the pairs of points that are in the same cluster and the pairs that are in different clusters, while adjusting for chance. The ARI score can be interpreted in this manner:

ARI close to 1: This indicates a high level of agreement between the true and predicted labels, suggesting that the clustering algorithm has performed well.

ARI close to 0: This implies that the agreement between the true and predicted labels is no better than what would be expected by chance, indicating a poor clustering performance.

ARI close to -1: This indicates that the predicted labels disagree with the true labels, which is even worse than what would be expected by chance.

The calculation of ARI is done following these steps:

- Create a contingency table that shows the number of data points for each combination of true and predicted labels.
- Calculate the Rand Index (RI) by summing the number of pairs of data points that are either in the same cluster in both the true and predicted labels or in different clusters in both the true and predicted labels, and then dividing by the total number of pairs of data points.
- Calculate the expected RI (ERI) by considering the number of pairs of data points in each cluster for both the true and predicted labels and then dividing by the total number of pairs of data points.

The formula calculating the ARI score is the following:

$$ARI = \frac{(RI - ERI)}{\max(RI) - ERI}$$

### 3.3.2 Neural network

In the supervised learning phase, the neural network model was constructed using the Keras library integrated with TensorFlow. The model consists of a fully connected feedforward neural network, also known as a Multi-Layer Perceptron (MLP). This neural network includes the following layers:

1. An input layer with 64 neurons and a ReLU (Rectified Linear Unit) activation function, designed to accept 10-dimensional input features.

2. A dropout layer with a 0.2 rate, which mitigates overfitting by randomly setting a fraction of input units to 0 during training.

3. A dense hidden layer with 32 neurons and a ReLU activation function, responsible for learning more complex representations from the input data.

4. Another dropout layer with a 0.2 rate.

5. An additional dense hidden layer with 16 neurons and a ReLU activation function, which learns even more complex representations from the data.

6. An output layer with 4 neurons and a SoftMax activation function. The SoftMax function transforms the output into probabilities, with each neuron representing one of the four mood categories ("calm", "energetic", "happy", and "sad"). The class with the highest probability is selected as the final prediction.

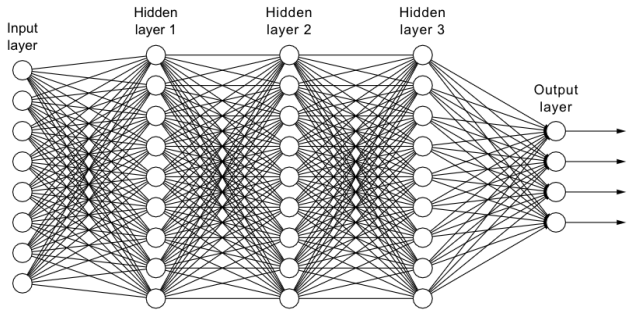


*Figure 2 Example of a fully connected neural network with dimension 8 input layer, 3 hidden layers and a dimension 4 output layer. The number of nodes is different, but the NN is similar to the one used in this project (FreeContent by Manning, n.d.).*

The model is compiled using the sparse categorical cross-entropy loss function, suitable for multi-class classification problems with integer labels. The 'adam' optimizer, a widely used efficient optimization algorithm, is employed in the model. To evaluate the neural network's performance, we utilized 10-fold cross-validation, implemented using the 'cross_val_score' method from 'sklearn.model_selection'. K-fold cross-validation is a popular technique for assessing machine learning model performance, particularly when data is limited. Its primary goal is to obtain a reliable and unbiased estimate of the model's generalization performance, or how well it performs on unseen data.

The importance of each input feature was investigated by permuting each feature individually. The classification accuracy for each randomized feature was then compared to the original non-randomized accuracy, and an importance score, ranging from 0 to 1, was calculated using the formula:

$$Importance = original\ accuracy - \frac{permutation\ accuracy}{3}$$

| Feature | Importance |
|---|---|
| danceability | 0.257 |
| energy | 0.258 |
| key | 0.257 |
| loudness | 0.258 |
| speechiness | 0.256 |
| acousticness | 0.257 |
| instrumentalness | 0.363 |
| liveness | 0.257 |
| valence | 0.257 |
| tempo | 0.259 |

*Table 1 The importance of each feature. The original accuracy used for calculation is 70%.*

Through this analysis, we concluded that all features exhibited similar importance, and thus,

all features were retained as input for the experiment.

# 4. Results

## 4.1 Data analysis outcomes

| Accuracy metric for K-means In unsupervised learning | Score |
|---|---|
| Silhouette score (With autoencoding) | 0.56 |
| Davies-Boulding (With autoencoding) | 0.49 |
| Calinski-Harabasz (With autoencoding) | 18265 |
| Silhouette score (No autoencoding) | 0.23 |
| Davies-Boulding (No autoencoding) | 1.38 |
| Calinski-Harabasz (No autoencoding) | 1605 |

*Table 2 The accuracy metrics for K-means clustering for unsupervised learning.*

The accuracy metrics scores for K-means clustering for unsupervised learning show that the data points fit moderately well within its clusters meaning that it is a decent clustering. The higher scores after using the autoencoder to reduce the dimensionality to 1 show that by doing this process, a significant amount of noise reduction in the audio features was done.

| Accuracy for NN Across 10-fold cross-validation | Score |
|---|---|
| Average | 0.744 |
| Best | 0.760 |

*Table 3 The 10-fold cross-validation accuracy for the neural network model in supervised learning.*

For supervised learning, the average accuracy across the 10-fold cross-validation at 74% shows that most of the time, the model will accurately categorize the mood of a song.

| Silhouette score metric for K-means | Score |
|---|---|
| Unsupervised and autoencoding | 0.56 |
| Unsupervised and no autoencoding | 0.23 |
| Supervised and autoencoding | 0.32 |
| Supervised and no autoencoding | 0.25 |

*Table 4 The Silhouette scores for K-means clustering with and without autoencoding in both unsupervised and supervised learning.*

[table with ARI score for k-means supervised]

| ARI score for K-means Supervised learning | Score |
|---|---|
| With autoencoding | 0.390 |
| No autoencoding | 0.388 |

*Table 5 The ARI scores for K-means clustering with and without autoencoding in supervised learning.*

The results obtained from these experiments show that the performance of K-means clustering is much lower with a labeled dataset than with an unlabeled dataset.

# 5. Discussion

## 5.1 Interpretation of results

The results presented in the previous section led to the observation that the K-means clustering model is better suited for handling unlabeled datasets. On the other hand, fully connected neural networks demonstrate superior performance when working with labeled datasets. This finding is consistent with other similar research projects (Mischitto, 2021), which indicate that neural networks are currently the leading models for labeled music

classification, slightly outperforming random forest classifiers.

One possible reason for the enhanced performance of neural networks in classifying labeled data is their ability to learn complex, hierarchical representations of the input data through multiple layers of interconnected nodes. In the context of music mood classification, this allows neural networks to capture intricate relationships between various song features and their corresponding mood labels. Additionally, neural networks can leverage advanced optimization techniques, such as backpropagation and adaptive learning rates, which enable them to fine-tune their weights and biases to minimize classification errors effectively.

Conversely, K-means clustering, as an unsupervised learning algorithm, is primarily designed to partition data into clusters based on similarity, without considering any explicit target labels. While it can be adapted to supervised learning tasks, it may not be as effective at learning complex relationships between input features and target labels as neural networks. Furthermore, K-means clustering relies on distance-based similarity measures, which may not be optimal for capturing the nuances of music mood classification, particularly when dealing with high-dimensional data or features that have different scales or units.

We can also notice that in both unsupervised and supervised learning cases, using an autoencoder, even though the input dimension of 10 is not relatively high, can improve the accuracy metric scores. While dimensionality reduction has negligible effect in this experiment, the autoencoder provides noise reduction on the data.

## 5.2 Limitations and manual testing

Comparing the results of supervised and unsupervised learning models in this project poses a challenge, as the performance scores for the neural network and K-means clustering models are based on different metrics. However, it is possible to draw some conclusions by considering their respective scales. The neural network achieved an accuracy of 74%, while the K-means clustering model obtained a Silhouette score of 0.56, which indicates a moderately well-performing cluster. It is important to note that a perfect clustering does not necessarily guarantee flawless classification, as the mood labels would still need to be verified manually.

To further assess the performance of these models, we conducted a qualitative evaluation by listening to the music and comparing the predicted mood labels with our subjective impressions. Based on this analysis, the neural network's classification appeared to be slightly more accurate and consistent. As a result, we conclude that neural networks are more effective in achieving the research objective of this project, which is to accurately classify music by mood using audio features.

## 5.3 Sources of errors and potential improvements

The main factors influencing the results in this study is the inherent uncertainty associated with the data obtained from Spotify. The top charts are subject to frequent fluctuations, making the results vary throughout the semester based on the trending songs at the time. Moreover, the Spotify API imposes a usage limit, which was reached each time we imported a new dataset, potentially affecting the consistency and quality of the data.

To enhance the reliability and robustness of the study, several improvements can be made. First, utilizing larger and more diverse datasets can help ensure that the models are trained on a comprehensive representation of the musical landscape, reducing the impact of temporal fluctuations in popular songs. Second, employing datasets that have been reviewed can provide more accurate mood labels, leading to better ground truth information for model training and evaluation. Lastly, exploring alternative data sources or supplementing the Spotify API data with additional features from other music databases could contribute to a more robust and generalizable analysis, ultimately improving the performance and applicability of the mood classification models.

## 6. Conclusion

The primary objective of this undergraduate research project was to identify an appropriate machine learning model for classifying songs into distinct mood categories based on audio features. By comparing the accuracy scores of K-means clustering in both unsupervised and supervised learning settings with that of a neural network, we discovered that the neural network, with a 74% accuracy, outperformed the other approaches. These findings have potential applications in music recommendation systems and may be related the methods employed by Spotify's AI DJ feature. Taking a mood-based perspective on music has the possibility to enhance user experiences and provide more personalized recommendations.

## Link to colab code

MusicMoodAI.ipynb - Colaboratory (google.com)

## Bibliography


Bhat, A. S., V. S., A., S. Prasad, N., & Mohan D., M. (2014). An efficient classification algorithm for music mood detection in western and hindi music using audio feature extraction. 2014 Fifth International Conference on Signal and Image Processing, pp. 359-364. DOI: 10.1109/ICSIP.2014.63

Nuzzolo, Michael. "Music Mood Classification." Tufts University, 2015, https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/.

AISongRecommender/Data/Training at main · michaelmoschitto/AISongRecommender (github.com)

Moschitto, M. (2019, August 14). Deep learning and music mood classification of Spotify songs. Medium. https://mikemoschitto.medium.com/deep-learning-and-music-mood-classification-of-spotify-songs-b2dda2bf455

FreeContent by Manning. (n.d.). Neural Network Architectures. Retrieved from https://freecontent.manning.com/neural-network-architectures/