

[Open in app](#)

[Sign up](#)

[Sign in](#)

Medium



Search



Write



Mood Based Music Recommendation System



Mayank, Meghavi, Pratyush, Shrishty, Vaibhav · [Follow](#)

17 min read · Dec 8, 2023

214

1

+

▶

↑



Music plays a crucial role in influencing and reflecting our moods, making mood-based music recommendation systems highly significant. By aligning music

choices with the listener's current emotional state, these systems enhance the overall listening experience. Music's profound impact on emotions and mental wellbeing means that the right selection can offer comfort, motivation, or relaxation. This personalization makes music more than just entertainment; it becomes a supportive companion, adapting to and improving the listener's mood.

Furthermore, mood-based recommendations aid in discovering new music and artists, tailored to the listener's emotional context, thus broadening their musical exposure. Such systems also foster user engagement and loyalty by consistently delivering relevant and enjoyable music experiences. From a commercial perspective, this leads to increased usage and potential revenue growth for streaming services. The adaptive learning of these systems ensures that recommendations evolve with changing preferences, making them an indispensable tool in today's digital music landscape.

Introduction and background

Why is the problem important?

The project addresses the challenge of enhancing the music listening experience by integrating emotional states into the recommendation process. This is significant as music is known to have a profound impact on emotions, and current recommendation systems do not adequately account for the listener's mood, potentially overlooking the deeper emotional connection that could be achieved with more personalized selections.

Related Work

The evolution of music recommendation systems has closely followed the rise of digital music. Initially, these systems relied primarily on two algorithms: collaborative filtering (CF) and content-based models (CBM). Collaborative filtering works by predicting a user's music preferences based

on the listening behaviours and ratings of similar users, essentially leveraging the power of collective preferences. In contrast, content-based models recommend music by analysing the acoustic features of songs, such as rhythm, pitch, and genre, focusing on the properties of the music itself.

As user needs became more complex, newer models emerged, notably the emotion-based and context-based models. These advanced models offer recommendations based on the user's current mood or situational context, adding a deeper level of personalization. Hybrid models, which combine various approaches, have demonstrated superior performance compared to individual methods. This ongoing development in music recommendation systems reflects a continuous effort to enhance user experience in music discovery, making these systems more adaptive and attuned to individual preferences and emotional states.

Outline of Approach and Rationale

The “Mood Based Music Recommendation” introduces an innovative approach to music recommendation, merging advanced image processing and mood recognition with personalized music selection. Utilizing a Convolutional Neural Network (CNN), the project focuses on identifying a listener's current emotional state — Angry, Happy, Sad, or Calm — through analysing facial expressions in images or videos. This mood recognition aspect is critical as it forms the basis for a tailored music recommendation system. The project's unique feature lies in its ability to align music recommendations with the detected emotional state, enhancing the relevance and impact of the suggested songs.

In the second phase of the project, songs are categorized into the emotional states based on their intrinsic features, employing content-based filtering for recommendations. This method ensures that the music selection is not only

aligned with the user's preferences but also resonates with their current mood. The system's aim is to enrich the music listening experience, offering songs that are not just liked but are emotionally fitting as well. By integrating emotional intelligence into music recommendation, the project intends to create responsive and empathetic AI systems, ensuring a more personalized and emotionally engaging user experience.

The rationale behind this dual-component strategy is to create a more intuitive and satisfying user experience by not just considering what users have enjoyed in the past but how they feel in the present moment.

Novel contribution

The novel contribution of this project lies in its use of real-time emotional feedback to inform music recommendations. Unlike traditional systems that rely solely on past behaviour or explicit user input, this project leverages advanced machine learning techniques to understand and predict the emotional impact of music, offering a more dynamic and responsive recommendation service

Data Collection and Preprocessing

Part 1: Mood Detection

Relevant Characteristics

Our emotion recognition model is trained on a subset of the FER2013 dataset, focusing on four primary emotions: Happy, Angry, Sad, and Calm (neutral expressions are converted as Calm). This curated dataset provides a balanced and diverse collection of facial expressions, enabling the model to accurately classify emotions in real-world scenarios.

The dataset after pre-processing has 26217 grayscale images of the format 48x48. The dataset has the following emotion classes

- **Happy:** Expressions of joy and positivity; **8,989** images
- **Angry:** Faces depicting anger and frustration; **4,953** images
- **Sad:** Representations of sadness and melancholy; **6,077** images
- **Calm:** Neutral expressions and calm states; **6,198** images



Example of different emotions in the dataset

Data Source

The FER2013 dataset was sourced from the Kaggle website. The dataset can be accessed through the following link: [Mood Dataset](#)

Data Preprocessing

The dataset was processed through the following steps:

1. Data selection: We refined the dataset to focus on the emotions most relevant to our application, namely Happy, Angry, Sad, and Calm. Neutral expressions were categorized as Calm, contributing to a more balanced emotional representation.

2. Exploratory Data Analysis: EDA was performed on the dataset to identify the proportion of emotions in the training, validation and test dataset.

Part 2: Song Recommendation

Relevant Characteristics

The dataset used for the mood-based music recommender system is sourced from Kaggle and contains a comprehensive collection of songs available on Spotify. Each entry in the dataset represents a unique song and includes a variety of features that capture different aspects of the music.

It has 42,000 Songs with 12 Features

- **11 Song features:** Energy, Valence, Loudness, Key, Tempo, etc.
- **5 Unique Identifiers and URI:** ID, uri, song_name, track_href, url
- **3 Additional Identifiers:** Genre, duration, timesignature

A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
1	danceability	key	loudness	mode	speechiness	acousticness	instrumentalness	valence	tempo	type	id	uri	track_href	analysis_url	duration_ms	time_signature	genre	song_name	Unnamed: 0	title					
2	0.831	0.814000000	2	-7.364	1	0.42	0.0598	0.0134	0.0556	0.389	156.985	audio_feature_2Vc6Nj9PWi.spotifytrack:https://api.s	https://api.s	124539	4	Dark Trap	Mercury: Retrograde								
3	0.719000000	0.493	8	-7.23	1	0.0794	0.401	0	0.11800000	0.124	115.08	audio_feature_7pgB1VrSVr.spotifytrack:https://api.s	https://api.s	224427	4	Dark Trap	Pathology								
4	0.85	0.893	5	-4.783	1	0.0623	0.0138	4.14e-06	0.37200000	0.0391	218.05	audio_feature_OvSVgAhIpxy.spotifytrack:https://api.s	https://api.s	98821	4	Dark Trap	Symbiot								
5	0.476000000	0.780999999	0	-4.71	1	0.10300000	0.0237	0	0.114	0.175	186.94799999	audio_feature_OvSxjdqJkw.spotifytrack:https://api.s	https://api.s	123661	3	Dark Trap	ProductOfDrugs (Prod. The Virus								
6	0.797999999	0.624	2	-7.668	1	0.293	0.217	0	0.166	0.591	147.988	audio_feature_4jCegeqHnM1.spotifytrack:https://api.s	https://api.s	123298	4	Dark Trap	Venom								
7	0.721	0.568	0	-11.295	1	0.414	0.0452	0.212	0.128	0.109	144.915	audio_feature_6SpypLhWvn.spotifytrack:https://api.s	https://api.s	112511	4	Dark Trap	Gattaka								
8	0.718	0.667999999	8	-4.162	1	0.13699999	0.0254	0.0078	0.124	0.038	130.826	audio_feature_0XfObqjDav.spotifytrack:https://api.s	https://api.s	77584	4	Dark Trap	kamikaze (+ pulse)								
9	0.694000000	0.711	8	-5.525	1	0.221	0.0397	0	0.11199999	0.28300000	138.049	audio_feature_0LLeuNBWPi.spotifytrack:https://api.s	https://api.s	127524	3	Dark Trap	T.R.U. (Totally Rotten Undergrou								
10	0.774	0.750999999	1	-2.445	1	0.198	0.0614	0	0.0728	0.18899999	219.96	audio_feature_37gqBnUA2e.spotifytrack:https://api.s	https://api.s	140326	4	Dark Trap	I Put My Dick in Your Mental								

Snapshot of the Spotify dataset

Data Source

The dataset was obtained from Kaggle, a platform for predictive modeling and analytics competitions. The specific dataset can be accessed through the following link: [Spotify Dataset](#)

Data Preprocessing

The dataset was processed through the following steps:

- 1. Kaggle Dataset Download:** The dataset was downloaded directly from Kaggle, where it was shared by the user “mrmorj.” Kaggle provides a platform for data science competitions, datasets, and code sharing.
- 2. Exploration and Understanding:** Before using the dataset, an exploration was conducted to understand its structure, contents, and the features available for analysis. This step helps in determining the suitability of the dataset for the intended project.
- 3. Data Preprocessing:** The dataset underwent preprocessing to handle missing values, outliers, and ensure consistency in the format. This involved cleaning up the data to make it suitable for analysis and modeling.
- 4. Feature Selection:** Relevant features for the mood-based music recommender system were selected. Features related to mood, such as valence and energy, were of particular interest for building the recommendation algorithm.
- 5. Exploratory Data Analysis (EDA):** Exploratory Data Analysis was performed to gain insights into the distribution of moods and genres in mood.

6. Normalization/Scaling: Numerical features were normalized or scaled to ensure that they are on a similar scale, preventing certain features from dominating the model training process.

Modelling

Part 1 — Mood Recognition

We use a Convolutional Neural Network (CNN) that predicts the emotion of a face in each frame by processing the region of interest (ROI) containing the face.

Once a face is detected, the algorithm utilizes a pre-trained CNN to classify the emotional state of the individual. The CNN model consists of multiple convolutional and pooling layers, which automatically learn hierarchical features from facial expressions. The model is trained to recognize four distinct emotions: Angry, Happy, Sad, and Calm.

Before we move into the specific of what our model specifics were, we would deep dive into some of the basics of what constitutes a **Neural Network**:

Convolution

- Images are a matrix of pixels.
- Grayscale images have single plane whereas RGB have three planes.
- A 3×3 filter/kernel is applied to the input image, producing a convolved feature.
- This convolved feature is then forwarded to the subsequent layer.

1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1 <small>$\times 1$</small>	0	0
0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1 <small>$\times 0$</small>	1	0
0 <small>$\times 1$</small>	0 <small>$\times 0$</small>	1 <small>$\times 1$</small>	1	1
0	0	1	1	0
0	1	1	0	0

Image

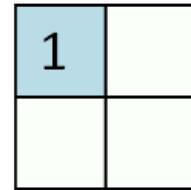
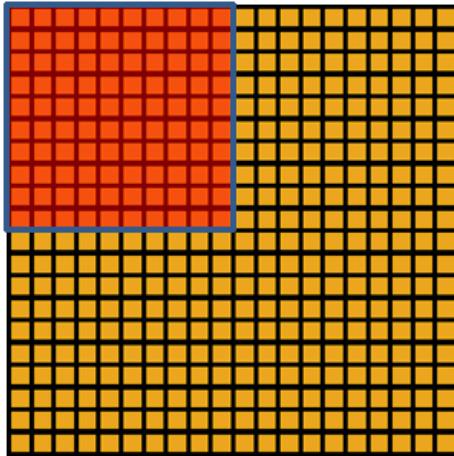
4		

Convolved Feature

Feature map creationg in a convolutional layer with a filter

Max pooling

- Identifying the highest pixel value within the area of the image covered by the kernel.
- Responsible for reducing spatial size of convolved feature.
- Decreases computational power required to process.



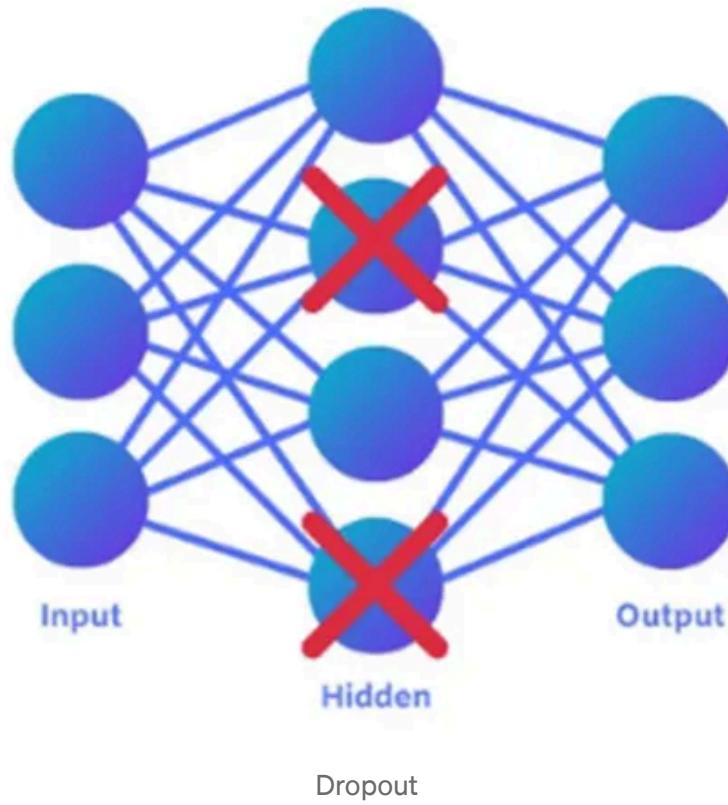
Convolved
feature

Pooled
feature

Downsampling operation of feature maps with a max pooling layer

Dropout

- Dropout randomly turns off neurons during training to avoid overfitting.
- It enhances generalisation to improve the network's ability to generalize by ignoring specific details.
- It acts like training multiple models by using different subsets of active neurons.



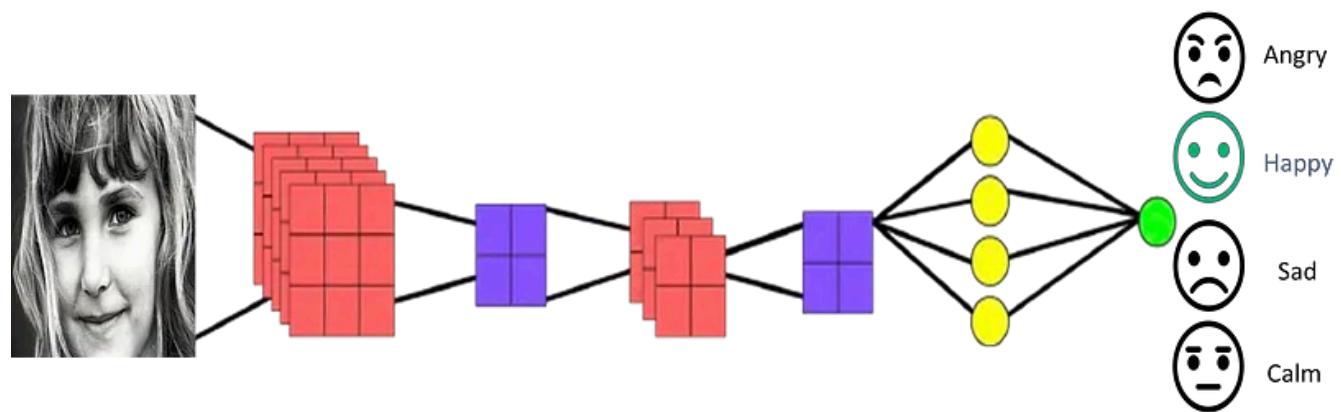
Mood Recognition CNN — Our Model

The CNN architecture includes convolutional layers to capture spatial features, max-pooling layers for downsampling, and fully connected layers for high-level reasoning. Dropout layers are incorporated to enhance model generalization and prevent overfitting. The final layer employs the softmax activation function to output probabilities for each emotion class.

- **Convolutional Layers:** The CNN begins with sequential convolutional layers configured with filter sizes escalating from 32 to 128, each with a 3x3 kernel size. These layers are designed to process grayscale images of 48x48 pixels, extracting features for classification.
- **Max Pooling:** Following each convolutional layer, max pooling with a 2x2 window is applied. This operation reduces the spatial dimensions of the feature maps, condensing the data and retaining the most significant features while reducing computational load.

- **Dropout:** To combat overfitting, dropout layers are interspersed after max pooling, with a dropout rate of 0.25 after convolutional layers and 0.5 after the dense layer. This technique randomly deactivates a subset of neurons, forcing the network to learn more robust features.
- **Dense Layers and Output:** Post feature extraction, the network employs a flattening layer to transform the data into a 1D array, followed by a dense layer with 1024 neurons. The architecture culminates in a dense output layer with 4 neurons and a softmax activation function, configuring the network for a 4-class classification problem.

For training, the model is compiled with the Adam optimizer and categorical crossentropy loss, suitable for multi-class problems. It employs a data generator for batch processing, essential for large datasets that may not fit into memory. The ‘fit_generator’ was used for an iterative training process over a number of epochs, with steps per epoch calculated to cover the entire training set in each epoch. Validation is performed similarly, ensuring the model is evaluated at the end of each epoch, though the specific number of epochs and batch size are not detailed in the snippet.



Mood Recognition Pipeline

```

import numpy as np
from google.colab.patches import cv2_imshow
import argparse
import matplotlib.pyplot as plt
import cv2
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Flatten
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
# os.environ['TF_CPP_MIN_LOG_LEVEL'] = '2'

# mode = "display"

# Create the model
model = Sequential()

model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(48,48,1))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

model.add(Flatten())
model.add(Dense(1024, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(4, activation='softmax'))

def emotion_recog(frame):
    model.load_weights('model_weights_training_optimal.h5')

    # prevents openCL usage and unnecessary logging messages
    cv2.ocl.setUseOpenCL(False)

    # dictionary which assigns each label an emotion (alphabetical order)
    emotion_dict = {0: "Angry", 1:"Happy", 2:"Sad", 3: "Calm"}

    # frame = cv2.imread("image1.jpg")
    # facecasc = cv2.CascadeClassifier('haarcascade_frontalface_default.xml') #
    facecasc = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_fronta

```

```

gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = facecasc.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=5)

for (x, y, w, h) in faces:
    cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 0, 255), 3)
    roi_gray = gray[y:y + h, x:x + w]
    cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray, (48, 48
prediction = model.predict(cropped_img)
maxindex = int(np.argmax(prediction))
cv2.putText(frame, emotion_dict[maxindex], (x+20, y-60), cv2.FONT_HERSHEY_SIMPLIFIED_ARIAL_9)

# cv2_imshow(frame)
return frame

```

Haar Cascade: Real — Time Image Processing

Face detection algorithm utilizes Haar Cascade classifier, a machine learning object detection algorithm used to identify faces in images or video frames. The Haar Cascade is adept at detecting facial features, allowing the system to isolate and extract regions of interest containing human faces.

```

facecasc = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
faces = facecasc.detectMultiScale(gray,scaleFactor=1.3, minNeighbors=5)

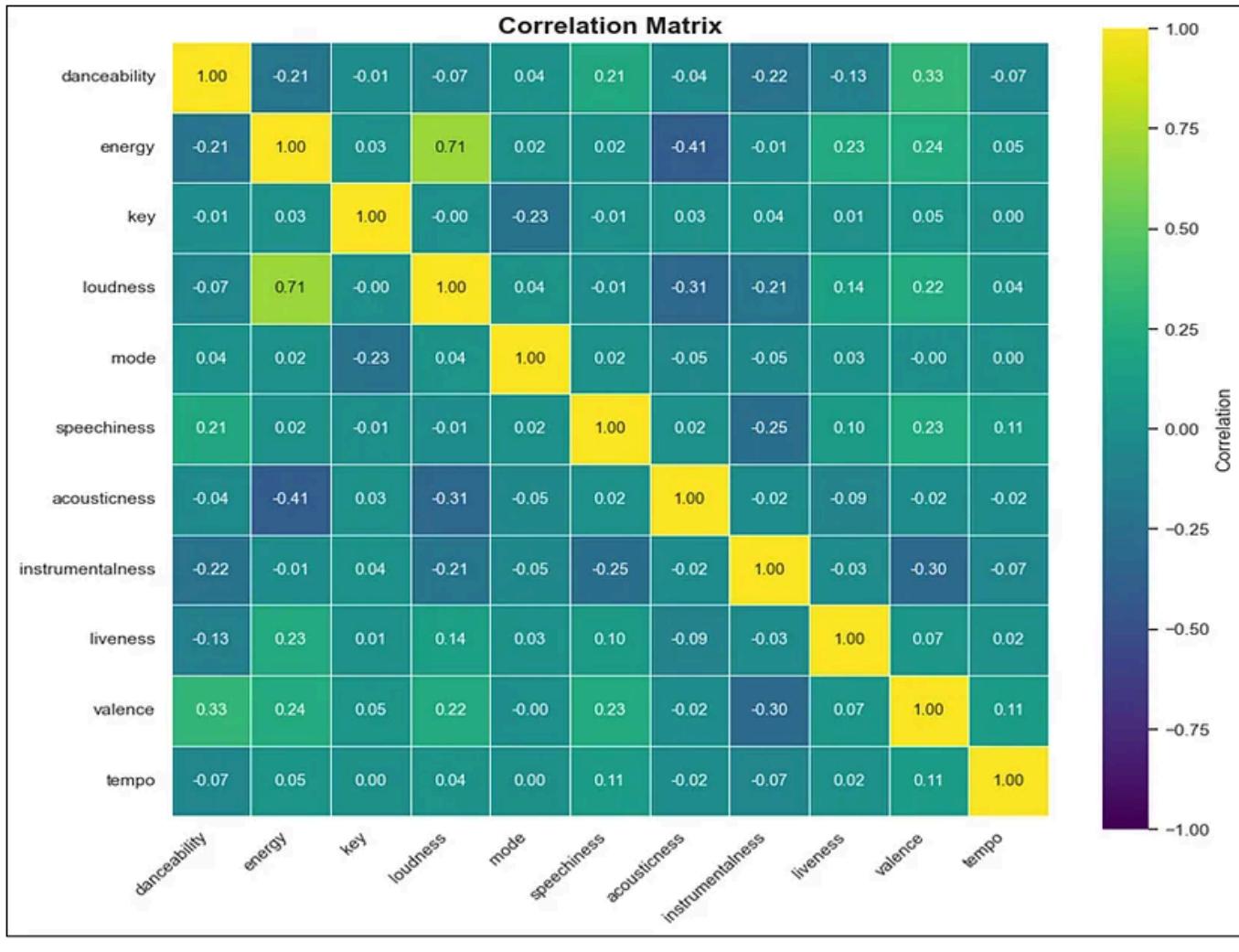
```

The CNN architecture includes convolutional layers to capture spatial features, max-pooling layers for downsampling, and fully connected layers for high-level reasoning. Dropout layers are incorporated to enhance model generalization and prevent overfitting. The final layer employs the softmax activation function to output probabilities for each emotion class.

Part 2 — Song Recommendation

Learning/Modelling

Generate song recommendations by clustering songs based on various features and categorizing them into four distinct moods. Utilizing content-based filtering, select and play the song from Spotify's API that best matches the user's preferences and current mood.

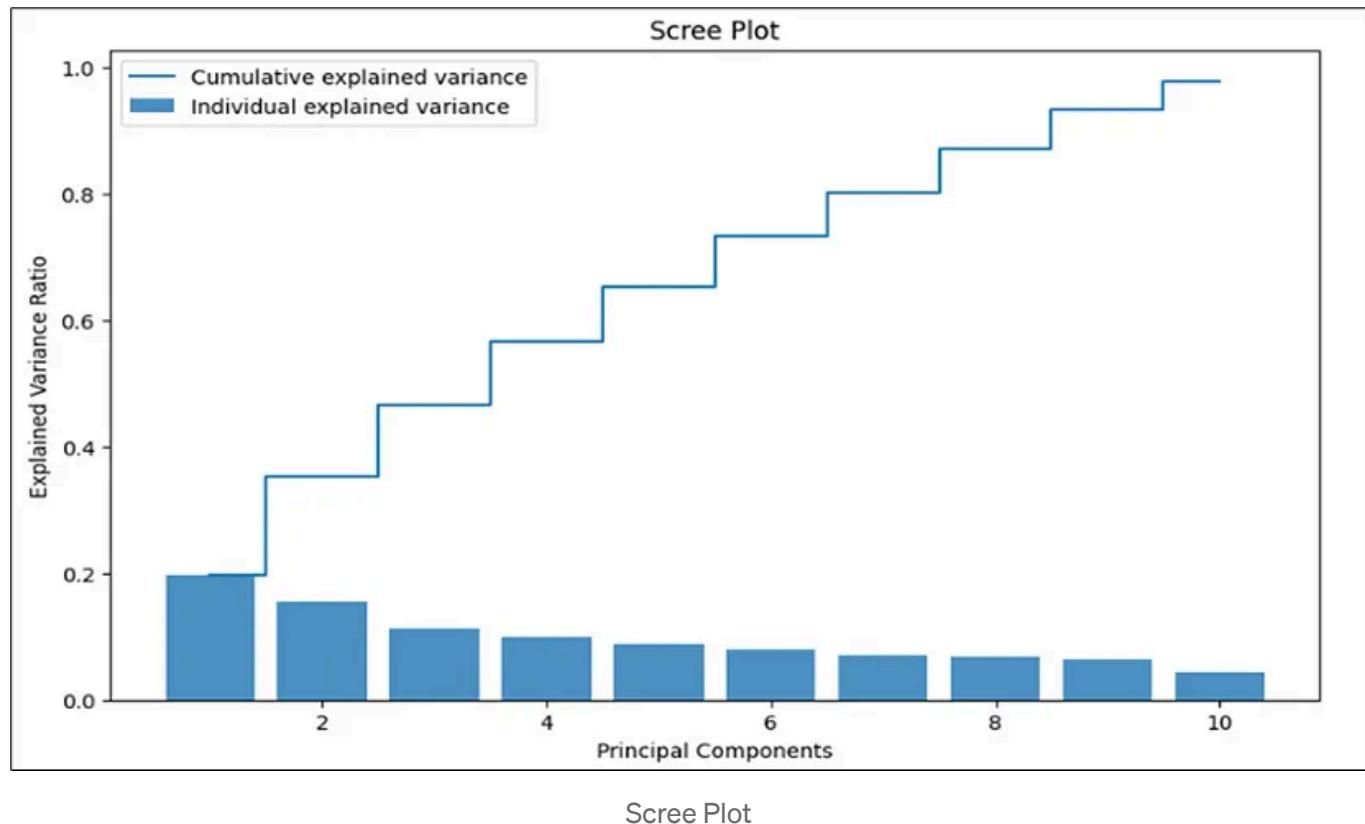


From the correlation matrix, we see that the song features are not correlated so we decide to move ahead with all of them starting with PCA followed by K-means Clustering Algorithm.

PCA

This is a dimensionality reduction technique which reduces the number of variables of a data set, while preserving as much information as possible.

From the scree plot, we can see that 4 Principal Components will capture most of the variance, So we take 4 PCs and proceed to create the clusters



K-means Clustering

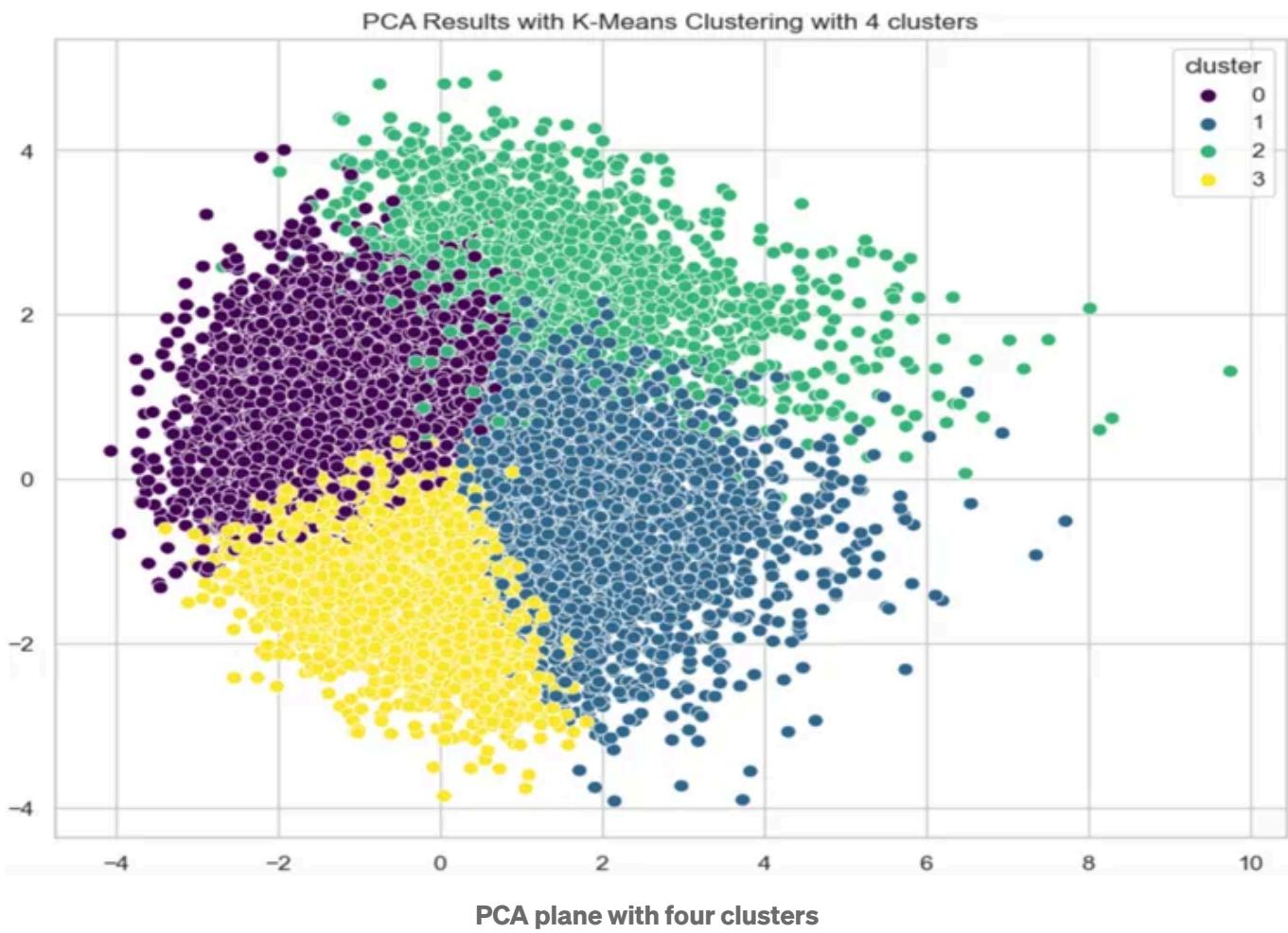
How does K-means work?

K-means aims to minimize the distance between data points and their cluster centroids, iteratively refining the clusters until convergence is achieved.

1. **Initialization:** Choose the number of clusters (K) and randomly position K centroids within the dataset.

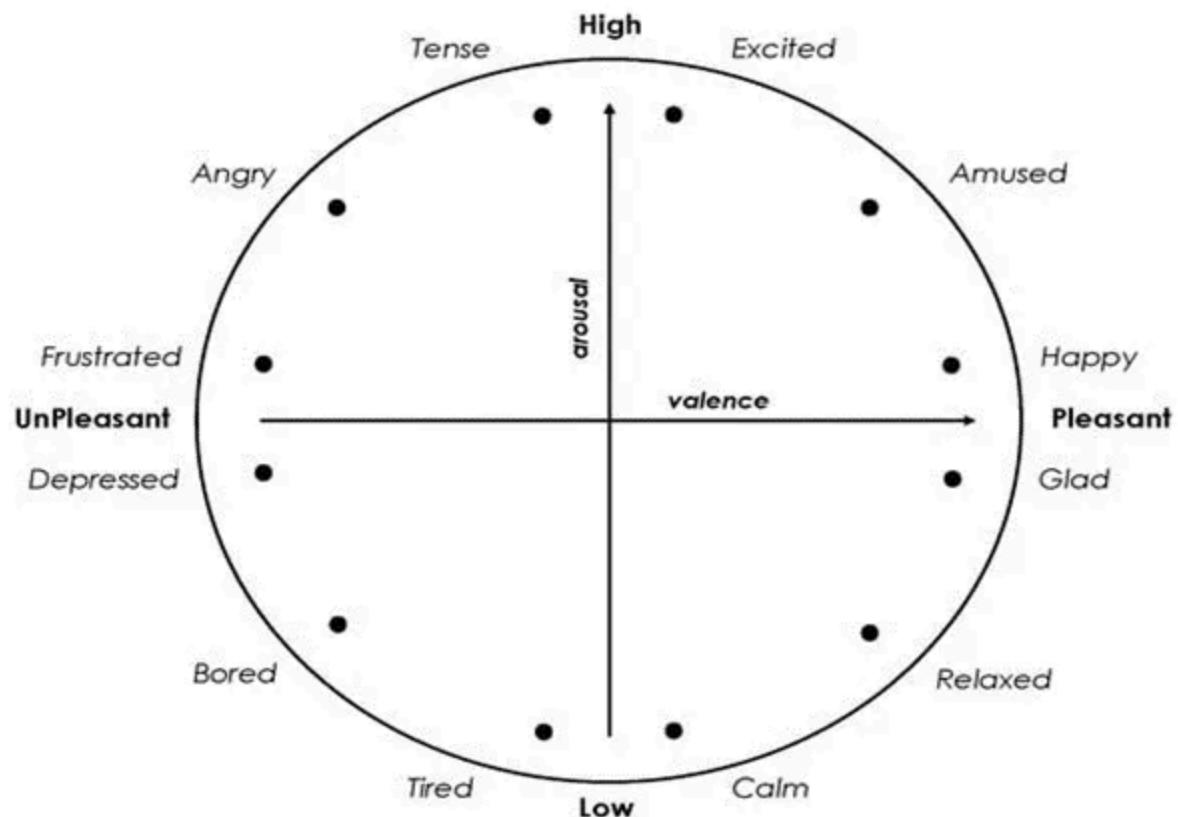
2. **Assignment:** Assign each data point to the nearest centroid, forming initial clusters.
3. **Update centroids:** Recalculate the centroids by taking the mean of all points in each cluster.
4. **Reassignment:** Reassign data points to the closest centroids based on the updated centroids.
5. **Repeat:** Iterate steps 3 and 4 until centroids stabilize or after a set number of iterations.
6. **Convergence:** The algorithm converges when centroids no longer change significantly.
7. **Final clusters:** Each data point belongs to the cluster with the nearest centroid, forming K distinct clusters.

We decide to create four clusters to map to each of the moods



Assigning a mood to the defined clusters

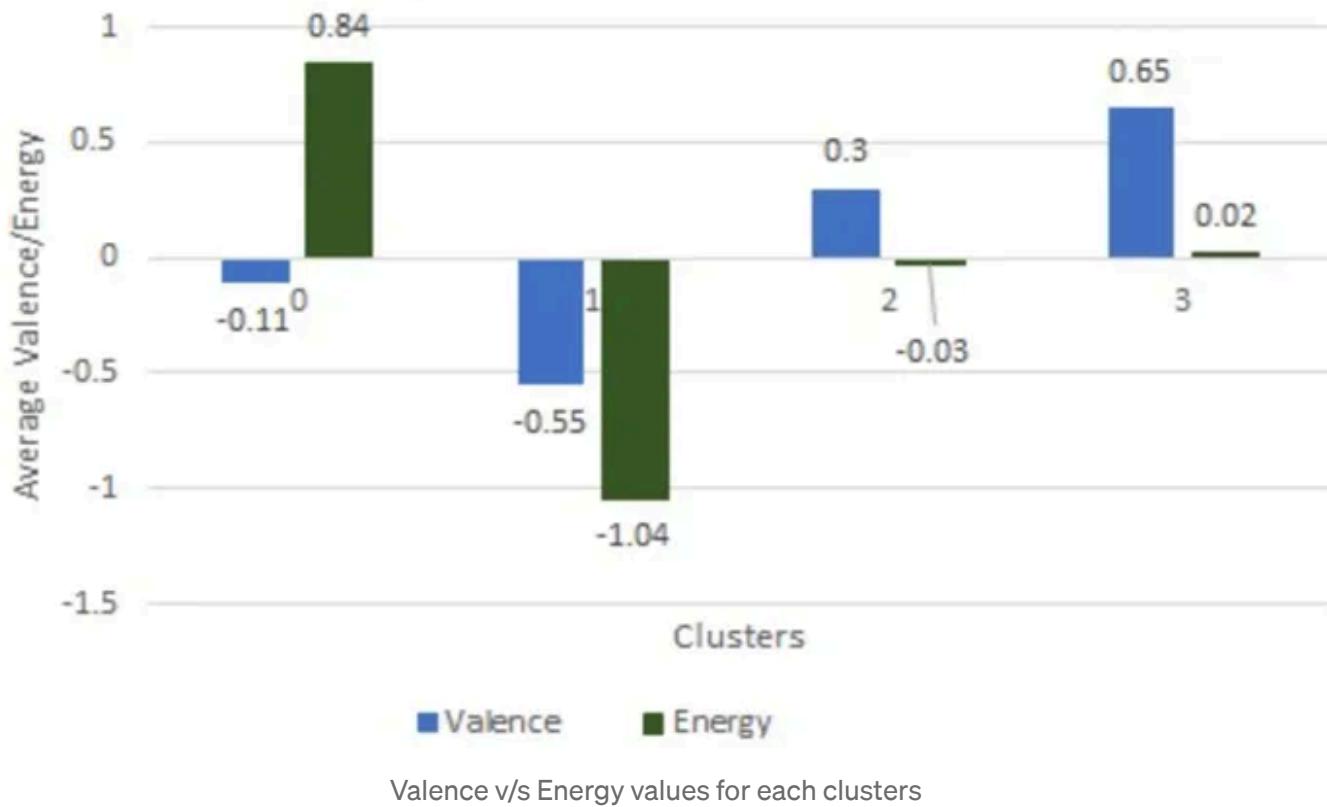
Among the several models of emotions proposed in the literature, one of the most used is the circumplex model defined by Russell. Such a model organizes the emotional states in terms of valence and arousal. The result is a two-dimensional space, where a pleasant-unpleasant (valence) value is represented by the horizontal axis and high-low arousal is represented by the vertical axis (see the below figure). We have used such a model by considering emotional states organized in the following groups: pleasant-high (excited, amused, happy), pleasant-low (glad, relaxed, calm), unpleasant-high (tired, bored, depressed), and unpleasant-low (frustrated, angry, tense).



Circumplex Model for emotion et al Russell

We map the clusters based on the above image using the average valence and arousal in each of the clusters.

Average Valence and Energy for each clusters



From the graph we can see that the clusters would be map in the following way:

- Cluster 0: low valence and very high energy -> Angry
- Cluster 1: very low valence and very low energy -> Sad
- Cluster 2: low valence and very low energy -> Calm
- Cluster 3: very high valence and high energy -> Happy

Song Recommendation based on user choice

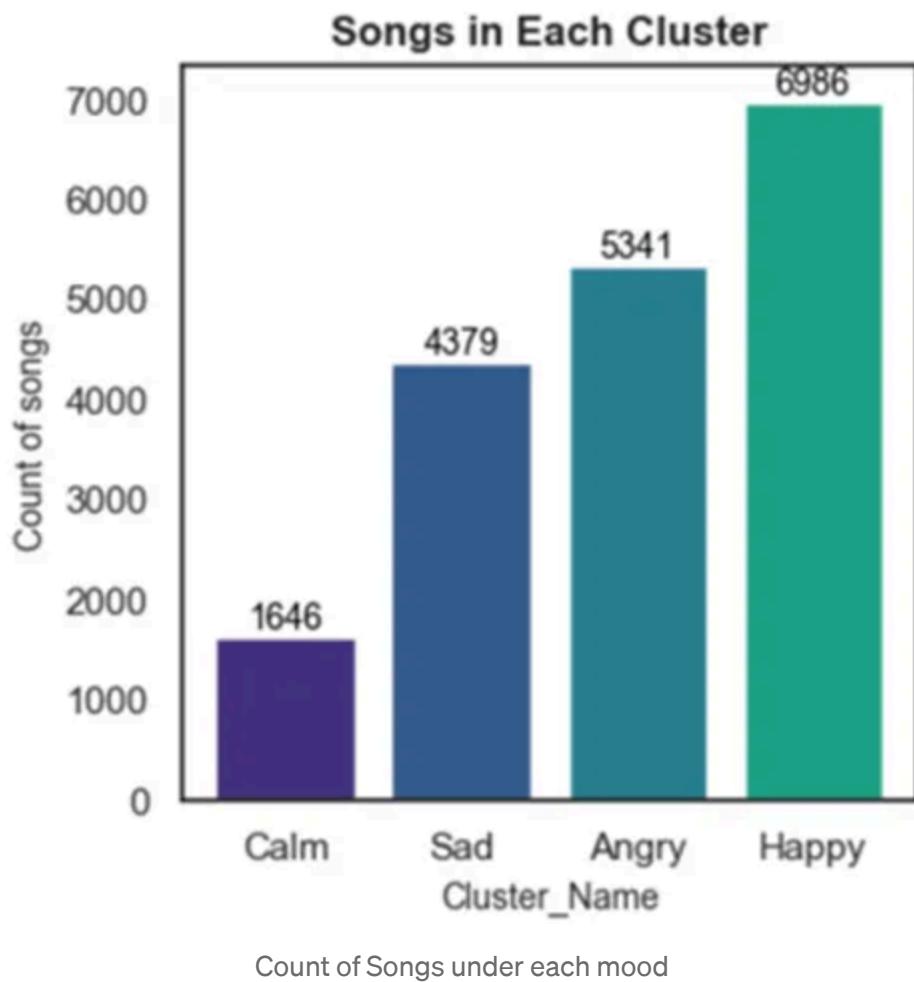
1. Five Random Songs: Based on the mood, we filter the song DB for that mood and suggest five random songs to the user.

2. Content-Based Filtering: User will now select one song out of the above five songs and then we run a cosine similarity based on the valence and energy with the detected mood songs in DB to finally recommend the top songs which can fix your mood.

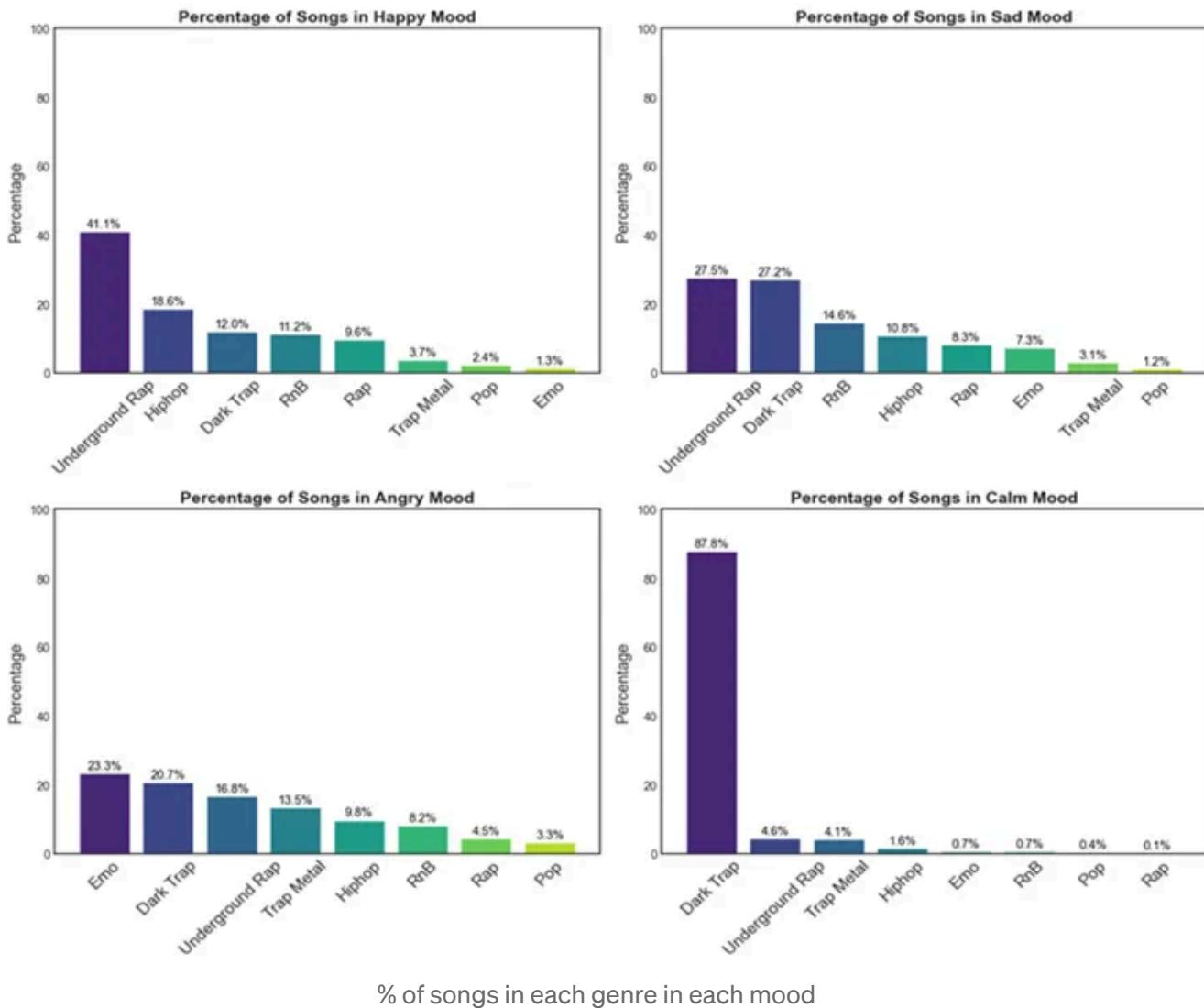
3. Hit the Music: Finally, we play the top song directly on the Spotify app or desktop using the API call.

Deep Dive into the Moods created

Distribution of moods



Distribution of genres in each of the moods



The graph illustrates mood distributions in music genres, highlighting that Underground Rap is predominantly Happy, Emo and Underground Rap share the lead for sadness, Emo is most associated with Anger, and Dark Trap is overwhelmingly calm. Pop consistently shows the least mood representation, especially in sad and calm categories.

Other models used

For Image Recognition: During the training phase on a large dataset, difficulties were encountered, likely due to the extensive computational demands of the model. To manage this, the model's state was saved periodically after training sessions. While the performance post-training

was satisfactory, the use of the saved models for subsequent tasks presented issues, with the model failing to function as expected when reloaded. This suggested potential problems in the saving or loading process, which could be due to issues like incompatibility of saved model formats or corruption of the model weights during the saving process.

For Music Recommendation: For clustering techniques, we decided to use other clustering techniques to get a better differentiation between the clusters on the valence arousal plane. Some of the techniques were:

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

Density-based clustering: DBSCAN is a popular clustering algorithm that groups together points in a dataset based on their density. It identifies clusters as regions in the data space where the density of points is higher than a specified threshold, allowing it to detect clusters of arbitrary shapes and handle noise effectively.

Birch (Balanced Iterative Reducing and Clustering using Hierarchies):

Hierarchical clustering approach: Birch is a hierarchical clustering algorithm designed for large datasets. It constructs a tree-like data structure (clustering feature tree) by recursively merging smaller subclusters to form larger clusters. Birch maintains compact representations of the dataset, making it memory-efficient and suitable for streaming data or applications with limited memory resources.

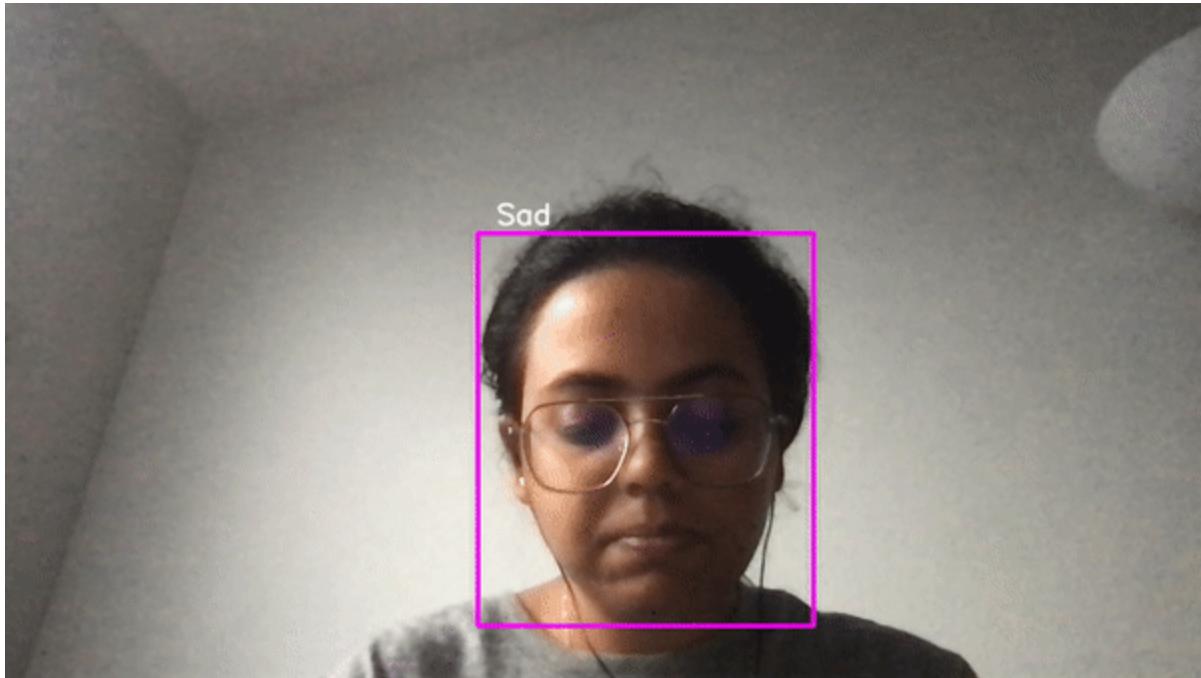
Both these algorithms led to oversized clusters and the later failed because of the absence of a large enough dataset.

There are some advanced neural network-based techniques that we wished to try but given the dataset limitations we had to look the other way, These are discussed

in the future steps section of the article.

Let's see how our recommendation system works!

1. Use OpenCV to take a real time image of user!



2. Our model tries to find the detected mood



3. To assess user preferences within a detected mood cluster, we present the user with five song choices. The user then selects the best song among these

options, helping us gauge their preferences based on the music presented. The second part of our model measures this choice and gives a recommendation along with similarity scores basis arousal and valence

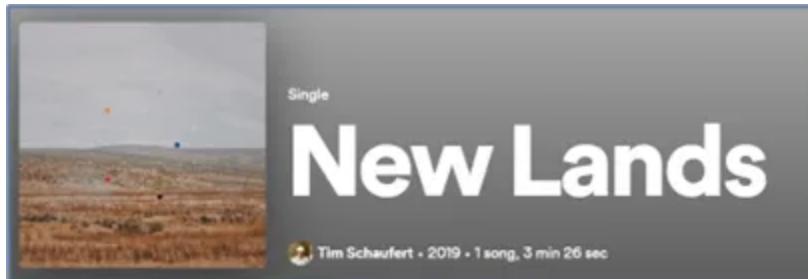
Select a so... Mw1 ▼

You selected: Purple Sky

Top 3 recommendations based on energy and valence:

	song_name	similarity
1	New Lands	1.000000
2	Blvck Rvin	0.999999
3	LinksysBroadbandRouter	0.999999

4. Play the top recommendation using Spotify via an API call to spotify



This code facilitates direct playback of the current top song on Spotify through API calls. By leveraging the Spotify API, it interacts with the Spotify app or desktop to initiate the playback of the most popular track. This streamlined process simplifies user interaction, allowing for seamless enjoyment of the latest trending music without manually searching or selecting songs.

```
import spotipy  
import os
```

```
import time
from spotipy.oauth2 import SpotifyOAuth

os.system("open /Applications/Spotify.app")
time.sleep(4)

# Replace 'YOUR_CLIENT_ID', 'YOUR_CLIENT_SECRET', and 'YOUR_REDIRECT_URI' with your own
client_id = 'e9c2ca587e9a49d68e0e7239facaf237'
client_secret = '92ab877118ba4e58859d6747f64e30cd'
redirect_uri = 'http://localhost:8888/callback' # Make sure this matches your Spacetime URL

# Set up the Spotify OAuth object with user authentication
scope = 'user-modify-playback-state user-read-playback-state'
sp = spotipy.Spotify(auth_manager=SpotifyOAuth(client_id=client_id, client_secret=client_secret, redirect_uri=redirect_uri))

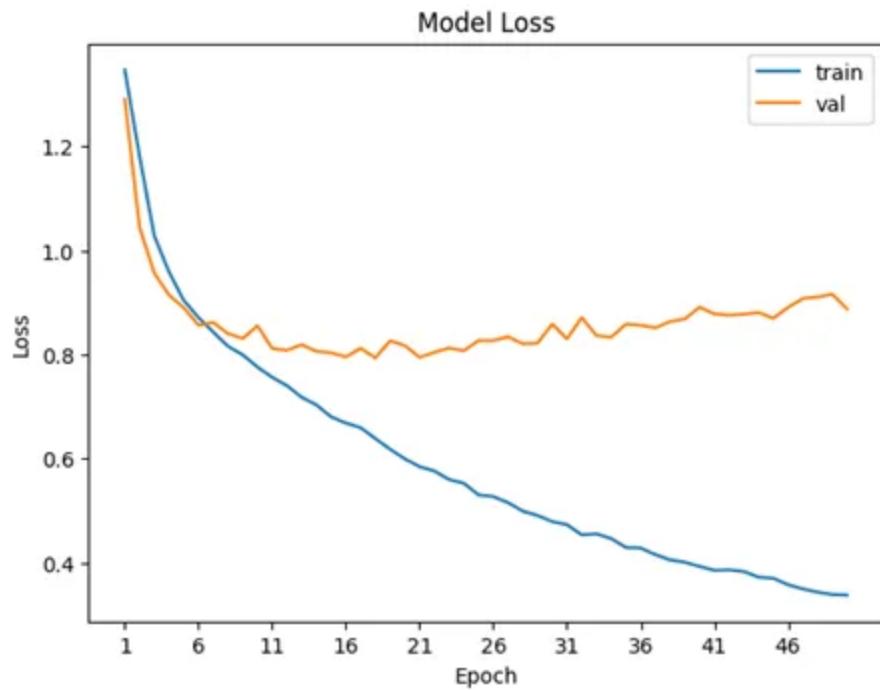
# Get the list of user's available devices
devices = sp.devices()
device_id = None

# Check if there are available devices
if devices['devices']:
    device_id = devices['devices'][0]['id'] # Use the first available device

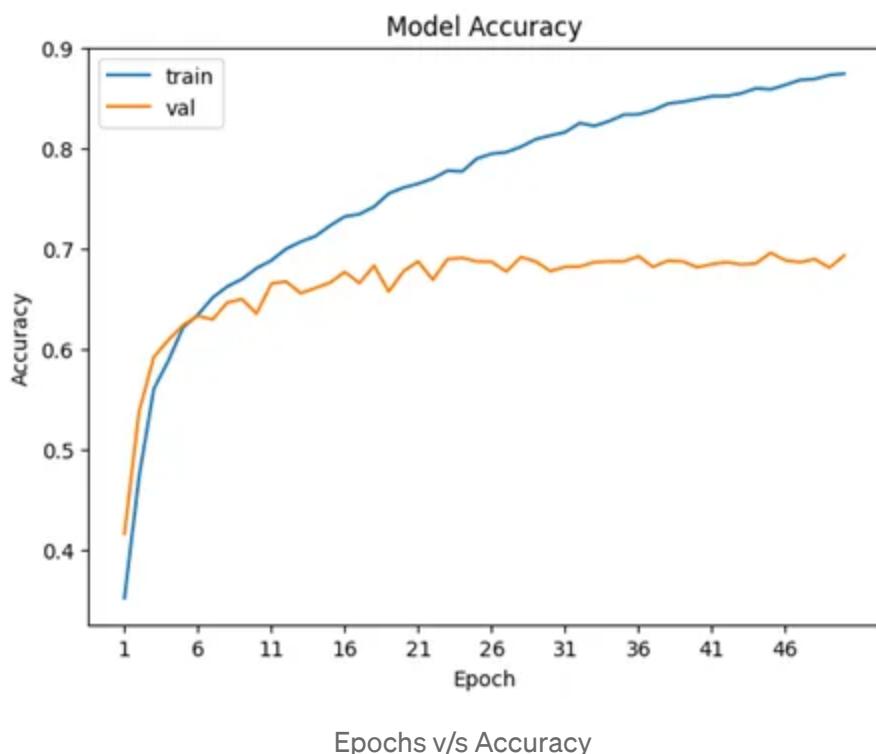
# Start playback of the specified track with the selected device
sp.start_playback(device_id=device_id, uris=[top_recommendation_uri])
```

Results

The model was trained for **50 epochs** and the validation loss and train loss were monitored. It could be observed that the lowest validation loss occurred at **16th epoch** beyond which the model started overfitting on the train data. Hence, the model was again trained for **16 epochs**. The trained model was evaluated on the test set which achieved a test set accuracy of about **70.28%**



Epochs v/s Loss



Epochs v/s Accuracy

Summary

The “Mood Based Music Recommendation System” enhances the music listening experience through real-time mood recognition and personalized

song selection. Using a Convolutional Neural Network (CNN) for facial expression analysis, it categorizes the user's emotional state (Angry, Happy, Sad, or Calm). Songs are then classified based on intrinsic features and emotional states, employing content-based filtering for recommendations.

The project's uniqueness lies in aligning music suggestions with the user's detected emotional state, creating a responsive and empathetic AI system. Learning involves CNN techniques like convolution, max pooling, and dropout. Song recommendation employs Principal Component Analysis (PCA) and K-means clustering, mapping clusters to emotional states using a circumplex model

The recommendation process offers five random songs based on mood, refining choices through content-based filtering and achieving a 70.28% test accuracy for image recognition. The Mood Based Music Recommendation System aims for a dynamic, personalized music experience by considering both user preferences and real-time emotional states

Lessons Learned

Developing the Mood Based Music Recommendation System has revealed crucial lessons, particularly in the context of the digital age where emotions are fundamental to ICT systems predicting social behaviours.

Mood Recognition Limitations

Incorporating emotion into the recommendation system involves dealing with computational demands, data requirements, and the risk of overfitting in Convolutional Neural Networks (CNNs). Effective regularization is essential to manage the complexities associated with mood recognition, requiring substantial computation and diverse datasets.

Song Recommender Limitations

The song recommender faces challenges in analyzing nuanced user preferences due to limited song features and minimal user preference information. This limitation hinders the depth of understanding, making personalized recommendations challenging and risking oversimplified generalized insights. Future improvements could explore ways to gather more comprehensive user data for a refined recommendation system.

In summary, the Mood Based Music Recommendation System acknowledges the significance of emotions in ICT systems, aiming to create a prototype that can evolve into a more user-centric experience. Addressing challenges in mood recognition computational demands and song recommender limitations is crucial for enhancing the system's accuracy and personalization capabilities in the future.

Future Work

Exploring emotion recognition, the project envisions potential advancements in accuracy by integrating diverse datasets. Additionally, deep learning-based recommender systems are employed to enhance personalization, providing context-aware music recommendations for a more tailored user experience.

Emotion Recognition Enhancements

- Integrate diverse and multi-modal data to increase accuracy in emotion recognition
- Utilize advanced CNN architectures, such as Inception and DenseNet, to improve result quality

- Include multi-modal data (facial expressions, voice data, textual context) for a more comprehensive mood analysis
- Leverage large datasets and pre-trained models, fine-tuned on mood-specific data to boost performance

Deep Learning-Based Recommender Systems

- Develop deep learning-based systems for personalized and context-aware recommendations
- Implement CNNs and RNNs for session-based recommendations, utilizing click history data
- Enhance traditional collaborative filtering with modern neural network approaches
- Employ Restricted Boltzmann Machines for scalable solutions in large dataset recommendation systems
- Apply neural attention-based models to filter out non-informative content and identify the most representative items, ensuring interpretability

Github Link: <https://github.com/pratyush335/Mood-Based-Song-Recommender/>

References

ChatGPT is used for syntax modifications

Analytics Vidya-Intro to CNN

Dr Furkan Kinli — Learning Emotions via Deep Learning

Roberto De Prisco-Emotion based music recommendation

[Deep Learning Based Recommender Systems](#)

[Mood-Based Music Recommendation](#)

[Spotify Music Data Analysis](#)

[Emotion Recognition Datasets Analysis](#)

AI

Machine

Machine Learning

Musicrecomendation



Written by Mayank, Meghavi, Pratyush, Shrishty, Vaibhav

31 Followers

Follow

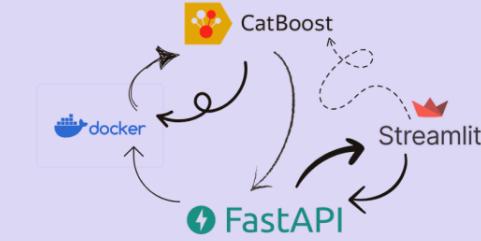


Recommended from Medium

```
*___, a, b, ___ = [1, 2, 3, 4, 5, 6]
print(___, ___)
```

What does this print?

- A) Syntax error
- B) [1] [4, 5, 6]
- C) [1, 2] [5, 6]
- D) [1, 2, 3] [6]
- E) <generator object <genexpr> at 0x1003847c0>



 Liu Zuo Lin

You're Decent At Python If You Can Answer These 7 Questions...

No cheating pls!!

 Mar 6  6.1K  30



 Ramazan Olmez

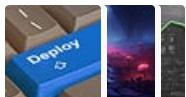
End-to-End Machine Learning Project: Churn Prediction

The main objective of this article is to develop an end-to-end machine learning project. For...

Feb 22  1.2K  15



Lists



Predictive Modeling w/ Python

20 stories · 1372 saves



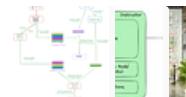
The New Chatbots: ChatGPT, Bard, and Beyond

12 stories · 422 saves



Practical Guides to Machine Learning

10 stories · 1660 saves



Natural Language Processing

1575 stories · 1127 saves

Amazon.com Software Development Engineer Mar. 2020 – May 2021
Developed Amazon checkout and payment services to handle traffic of 10 Million daily global transactions
• Integrated Iframes for credit cards and bank accounts to secure 80% of all consumer traffic and prevent CSRF, cross-site scripting, and cookie-jacking
• Led Your Transactions implementation for JavaScript front-end framework to showcase consumer transactions and reduce call center costs by \$25 Million
• Recovered Saudi Arabia checkout failure impacting 4000+ customers due to incorrect GET form redirection

Projects

NinjaPrep.io (React)

- Platform to offer coding problem practice with built in code editor and written + video solutions in React
- Utilized Nginx to reverse proxy IP address on Digital Ocean hosts
- Developed using Styled-Components for 95% CSS styling to ensure proper CSS scoping
- Implemented Docker with Seccomp to safely run user submitted code with < 2.2s runtime

HeatMap (JavaScript)

- Visualized Google Takeout location data of location history using Google Maps API and Google Maps heatmap code with React
- Included local file system storage to reliably handle 5mb of location history data
- Implemented Express to include routing between pages and jQuery to parse Google Map and implement heatmap overlay





Alexander Nguyen in Level Up Coding

The resume that got a software engineer a \$300,000 job at Google.

1-page. Well-formatted.

★ Jun 1 ⌘ 13K 🗣 173



Daniel García

How I'd learn ML in 2024 (If I Could Start Over)

I am a PhD candidate and the head of ML department in an automotive industry...

★ Apr 2 ⌘ 751 🗣 10



See more recommendations

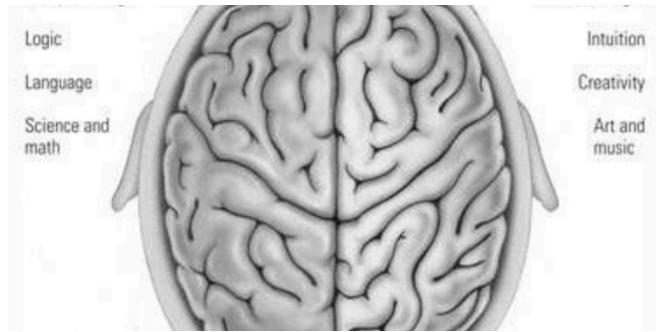


Jason Roell in Stackademic

Ultimate Python Cheat Sheet: Practical Python For Everyday...

(My Other Ultimate Guides)

Jan 29 ⌘ 5.7K 🗣 48



Sufyan Maan, M.Eng in ILLUMINATION

What Happens When You Start Reading Every Day

Think before you speak. Read before you think.—Fran Lebowitz

★ Mar 11 ⌘ 25K 🗣 550

