

Thanh Vinh Nguyen, 301634356

Matin Meraj Mohammadi, 301635545

Nadine Gunawan, 301539671

Jim Saraza, 301462218

Project idea

We will build a simple AI system that can classify songs based on mood. The problem we are solving is that instead of manually tagging songs, our system will take numeric audio features such as tempo/BPM and loudness/energy, along with other features that describe the overall intensity of a track, and then predict the mood. Additionally, we will independently predict the mood of a song from its lyrics to compare how well text-only signals align with audio-only signals. We will train an audio-based model using features and classic classifiers, as well as a text-based model. The goal is to design a model that takes a single song as input and outputs one of the four mood categories (happy, chill, sad, or hyped).

We plan to use supervised classification techniques, since we will be training the system on a dataset of songs where the moods are already labelled. For the classification algorithm, we are considering K-Nearest Neighbours and Logistic Regression, both available in Python through the `scikit-learn` library. For the text-based model, we are planning to use a simple API baseline (OpenAI). The system will follow a clear pipeline where we first collect song feature data from a dataset, then clean the data using Python libraries such as Pandas and NumPy, then train/evaluate both models (classifier using `scikit-learn`), compare predictions, and finally use the trained model to predict the mood of new songs.

Tools and Resources

Programming Language: Python

Libraries:

- Pandas and NumPy for data handling and cleaning
- Scikit learn for classification models (KNN and LogReg)
- PyTorch for training neural models
- Matplotlib for visualizations

Dataset:

- Real-world dataset of songs with audio features and mood labels
- Source: Kaggle dataset, OpenAI API (for text-based model)
- Start with a CSV containing tempo, loudness, valence, and mood labels

Data Handling: Use existing datasets, no simulator or self-generated data

Workflow: Import dataset and lyrics-> clean and preprocess -> train audio model with `scikit-learn` and lyrics model (API) -> evaluate and compare -> choose the best model.

Timeline/Project Plan

Milestone 1 - Oct 19

Main goal: Basic end-to-end classifier working on a small labelled dataset

- We will have the initial data ready (one labelled dataset has been chosen), where the labels are mapped to the four mood categories.
- Have the repo set up in GitLab.

- We will have the feature extractor script ready (the scripts that output the tempo/BPM and loudness/energy).
- We will test to get lyrics for a small set of songs and run a quick text model with the API.
- Baseline model:
 - Example: Sad/Chill: BPM < 90 and a low energy (calculated by the power or the energy of the song); BPM; Hyped: BPM > 120 and a high energy
- We will have the models trained with a small sample dataset
- After training the models on a small sample, we will evaluate the accuracy of the model we have built (can use a confusion matrix to show comparison examples vs lyrics prediction)

Milestone 2 - Nov 16

Main goal: a usable and stronger model with better evaluation that can run locally

- Expand to the complete dataset for both (audio and text) models and evaluate the best working model.
- Visualizations: confidence bar chart, mood map, confusion matrix
- Able to handle uncertainty (songs with uncertain predictions) based on the probability
- If possible, create a playlist based on the four mood categories, sorted by the confidence resulting from the model.
- Documentation: user cases/study summary.

Minimum Viable System

Simplest working version of a Python system that:

- Loads a labelled dataset of songs (maybe clips of the song/the full song)
- Extract basic features (numeric), such as BPM, loudness, and a few other stats.
 - Lyrics-based: get the lyrics text for a small subset
- Trains a single classifier using KNN
 - Lyrics: train a LogReg with API
- Compare audio-based vs lyrics-based mood
- Given one NEW audio file, able to output one of the four moods with its confidence score

What needs to be prioritized first:

- Feature Engineering: the ability to extract the feature using Python libraries from a CSV file
- Building a baseline model: majority class and the KNN, Logistic Regression
- Prediction result: able to print out the mood and confidence score

This MVS is straightforward and can be used immediately, as it automatically categorizes a folder into one of four moods and provides an audio vs. lyrics comparison with clear metrics, while also illustrating an end-to-end supervised machine learning approach.