Music Emotions Classification Using Artificial Intelligence

Cristian Alexis Rojas Gonzalez

Sinclair Community College

Honors Project Research Paper

Python for Data Analysis 2266.104

Mike Libassi

Dayton, Ohio

04/03/2025

Alexis Rojas Honors Project

INTRODUCTION

In this research paper, I will present and explain my Python AI project where I intended to create a machine learning model capable of predicting the emotion a human person might feel when listening to a specific song. Unlike most systems used to recommend songs to the user, this project is focused purely on the technical audio features of the music such as the tempo, the brightness, or the rhythm rather than using lyrics analysis, popularity, genre, artist similarities or user preferences.

To grant familiarity with the songs and also emotional relatability among the users, I chose well-known songs that most people could recognize and identify. Additionally, I selected six specific emotions that the AI will predict to select in every song by different reasons that will be explored in this paper.

When listening to music, usually we come across recommendations of songs and playlist that don't match our mood in the exact moment, for example, trying to listen to happy pop music in shuffle may give you some unexpected songs that are quite the opposite of what you're looking for. With this project I aim to show an innovative approach to redefine how we interact with music and our feelings and make more intuitive recommendation engines and perhaps, even therapeutic uses for the future

I decided to create my Honors Project in this topic thanks to my knowledge and deep interest in Artificial Intelligence, music and psychology. I consider that this project, and its development will take us a step further to enjoy in a more intense way the music and the songs of our favorite artists.

ABSTRACT

For a lot of us, music is something particularly important in our lives, and it is deeply connected with our feelings. This project uses artificial intelligence to classify songs based on the human emotion that we feel listening to them. A custom AI model was trained with a specific dataset of selected famous songs. With this project we could be a step closer to connecting even more with music through our emotions and apply it to different fields and the results open new possibilities for mood-based music recommendations and applications.

The goal is to explore how songs, without lyrics analysis, can be interpreted emotionally by analyzing technical audio features such as rhythm, speed, notes, harmonic content, etc.

PRE-DEVELOPMENT

Before starting with the coding and deployment, I wanted to plan specifically how the Artificial Intelligence were going to work, so I began by selecting the emotions I wanted to use in my model, I decided to use only six main emotions that are easy to identify with, I decided this exact number because the bigger the number of emotions, the lower the accuracy gets on the predictions, also, those six emotions are quite easy to relate to when listening a song, that will make the users feel more comfortable about the results provided by the model. The selected emotions were:

- Romantic (Ballads or romantic songs per se)
- Happy (The type of songs that improve your day and give you a smile)
- Energetic (Songs that make you feel active and feel like dancing)
- Sad (Slow songs that you listen to when you're not feeling quite great)
- Calm (Relaxing songs that are not related to sadness)
- Angry (Active songs that are usually related to Metal or Rock genres)

After classifying the emotions, I helped myself with a lot of musicians, musical producers and psychological studies related to the topic [1], to select the most useful technical features inside a song that could be used to interpret human feelings, some of those are used to measure very similar characteristics in order to have more accurate results. The decided features were:

- Mel-Frequency Cepstral Coefficients (MFCCs) => Reads the frequencies of Decibels across the song
- Chroma => Represents the tonal content of the relative strength in pitch
- Spectral Centroid => Reads the brightness or, perceived loudness
- Zero Crossing Rates => Reads when the music signal passes from positive to negative or vice versa, used to analyze percussion
- Root Mean Square (RMS) Energy =>Arithmetic operation to calculate average power or perceived intensity of a song
- Tempo => Represents the speed of the song in Beats per Minute (BPMs)

For extracting those features out of the songs, I used the Python library *"Librosa"* as it has a lot of functionalities and documentation that allow you to extract and manage those technical characteristics very easily without having to create code to do so from scratch. The only problem with using this library is the documentation and understanding how to use every function inside of it as it might be very confusing. I also used *"NumPy"* to manage the features along the whole song, for example, with *np.mean()* I will use the average of every technical characteristic as the reference for the measurements.

---

[1] Studies focused on how music affect human emotions, and which characteristics of the music do so

AI MODEL

For the Artificial Intelligence, I used the pre-made and supervised machine learning algorithm *"Random Forest Classifier"* Created by Scikit Learn. It is a model that is widely known and recommended by its robustness, ease of use, implementation and mainly, its ability to manage and handle classification tasks with large datasets and structures, an ideal case for classifying and analyzing multiple technical features of music files, as they explain: "Random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting." (SciKit Learn, 2025) [2]

The random forest works creating a group of decision trees, each of those decision trees are trained on different random sets of the data that's used to train the model (In this case, the songs) and also uses subsets of the features inside a song to generate decisions on the emotion inside of it. Once all of the decision trees are finished with the classification and the decision making process, the forest combines the results and determines which emotion is in the song by the quantity of trees, it's like a democracy vote, if ten trees voted for "Happy" emotion and eight trees voted for "Energetic" then, the forest will consider the right answer to be "Happy".

To improve the performance of the model and reduce potential problems in the future, I used different processing techniques of the data, such as, using the mean value of every technical feature in the first 60 seconds of the songs; normalizing values to ensure consistency in the tracks; and using data augmentation [3] by pitch-shifting every song to increase the volume and the variety of the songs that will be used to train the AI.

After completing the training, the model is able to make the classification of new songs that aren't labeled by analyzing the technical features and also by comparing them to the learned patterns. Although the accuracy reached in the Honors version of the model is 49%, this is considered to be a strong starting point considering the subjectiveness nature of the emotions and the complexity of interpreting them from music audio files without using the lyrics of the songs.

In the future of the project, more sophisticated models will be used, models like complex neural networks that can make better predictions, are faster and more reliable. However, Random Forest was an efficient and very easy to implement baseline for my first time creating AI models and for this Honors project.

---

[2] (SciKit Learn. 2025. *Random Forest Classifier documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html).

[3] Artificially increasing the size of the dataset, in this case, creating more songs with altered pitch

DATA CHARACTERISTICS

Using *yt-dlp* downloader [4], I created a .txt file that contained the URLs to ~150 YouTube songs and downloaded them to use them as a dataset to train the model. Specifying the emotions of each feeling and giving the Random Forest model the technical features, it will take its own guesses and compare them to the emotions I specified, then based on that comparison it will throw an accuracy and precision estimate.

For the honors project the estimation of precision in the overall emotions reached 49% using 60 seconds of the songs and ~450 songs. Going a bit deeper into the data augmentation, I used the pitch shift of *Librosa* to create two additional versions of every song, one version is two semitones up and the other one is two semitones down.

I selected the songs based on recommendations made by guitarists, singers, musicians, music producers, music concert judges and friends, they had to recommend one or two songs for the six emotions taking in consideration how much a song made them experience the desired feeling, meaning, they had to choose the saddest song for them, the one that made them feel more happy, the song that made them want to dance or sing immediately, etc.

WEB PAGE

Thanks to the well-known Python library *"Streamlit"* I created a web page that could be used to interact with the AI through a good,3 enjoyable and easy to use interface, I also added some JavaScript and HTML to the page so it can have more customization and have another feature that I wanted to include.

Streamlit has multiple built-in input methods, one of them is a file submission section, which can be used to manage files from the user's computer when submitted. I used this feature to read the song files and pass them to the AI, once the page reads the file, sends the bytes of information to the AI and gets back the emotion detected, then displays it on the screen and plays the first 15 seconds of the song so the user can re-listen to it and think if the AI was right about the estimation of the emotion. Additionally, a dropdown section is included and has all of the technical information used to analyze the song.

My father gave me the amazing idea to add a functionality to record your own audio and give it to the AI, this way you could analyze more personalized songs or recordings without needing to use one that's already created or without having to search for the file, for example, just playing any music streaming platform near the microphone to reproduce your favorite song, without downloading it and searching for the file in your computer.

---

[4] Command-line tool that's used to download videos from the web

Using HTML and JavaScript I created a button that looks as similar as possible to the Streamlit incorporated buttons that records audio when pressed, shows a message of feedback and a message to ask for at least 15 seconds of recording (There isn't a way to check that in the code right now, it's mostly trust in the user) and download the song when pressed the button to stop the recording. For the Honors Program version, the web page cannot read .WAV files yet, meaning that the audio recording is missing one last step, even though the rest of it works perfectly.

The web page had to go through multiple design changes due to technical issues, aesthetics or other distinct reasons, for the Honors Project has a faded background that includes a purple color chosen with an UI builder tool taking in consideration the colors of *Streamlit* and how they look together and, the buttons also had to be changed.

All of those changes can be visualized in the pictures section of this paper.

CONCLUSION AND FUTURE OF THE PROJECT

With the first phase of the project completed (The Honors version), we could evidence some interesting results. Most importantly, it is possible. It is completely possible to classify our favorite songs by the emotions we feel when we listen to them, it isn't a theory or a crazy idea anymore. It is a reality that we can improve the way we enjoy music and the recommendations we get from streaming services thanks to the AI.

Although the actual model has an accuracy of 49%, it is a solid foundation for getting to a bigger, better, faster and more capable model taking in consideration that this one does not rely at all on features like lyrics, artists, popularity, collaborations, genres associates but only, on raw audio.

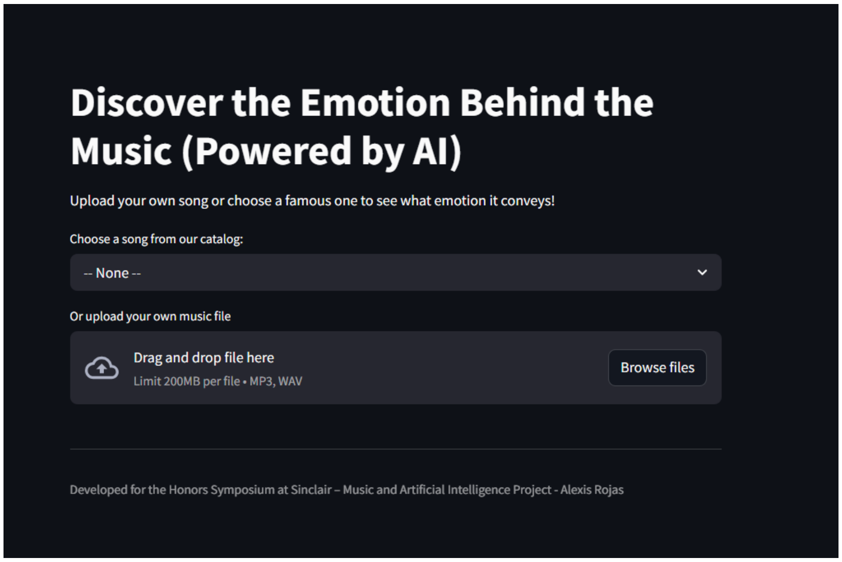
Looking into the probable future of this project, a lot of possibilities cross my mind, from the technical point of view, completing the audio recording tool has immense potential to enhance the user experience of the web page and give more possibilities of use to the project. Thinking about the possibilities of the project, it's application in music streaming platforms would change completely how those services provide us recommendations, for example, phrases like "I want to listen to only the sad songs of Nirvana", "I want relaxing music but only in Pop genre" could be possible now without having to skip songs that don't match what you're looking for and, uses in different fields like emotional education, therapy sessions, entertainment, marketing or even music production are completely possible, focusing on how a song will engage to the feelings of the public definitely changes what you want to reproduce to your patient, include in your marketing campaign or put inside your new album.

In the end, we can see now a further advance in how we interact with Artificial Intelligence, it isn't possible to just know what we enjoy, it is possible to know how we feel.

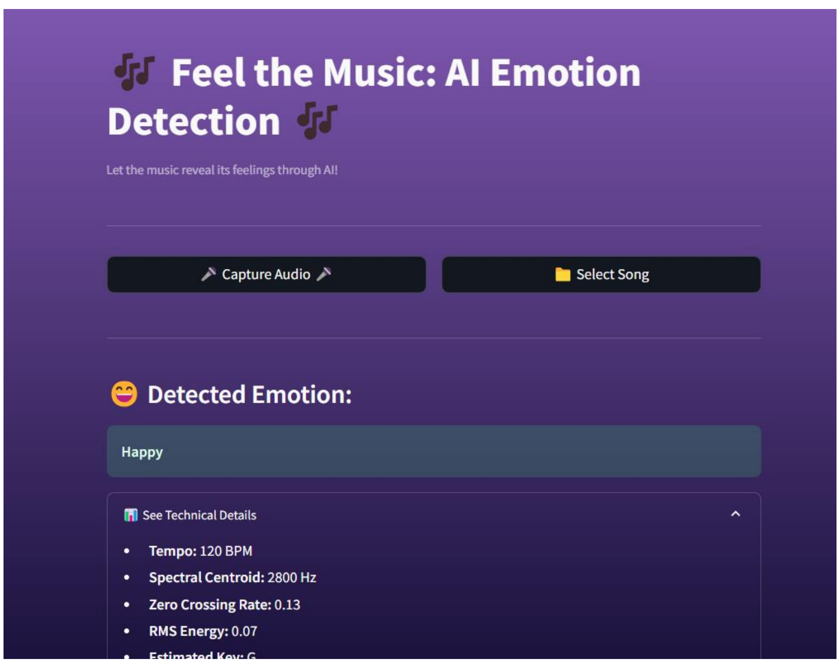
The project can be downloaded and tested by yourself with my GitHub repository.[5]

[5] Music Emotion Classification Online Repository - https://github.com/AL3XI5R/Music_emotion_clasification

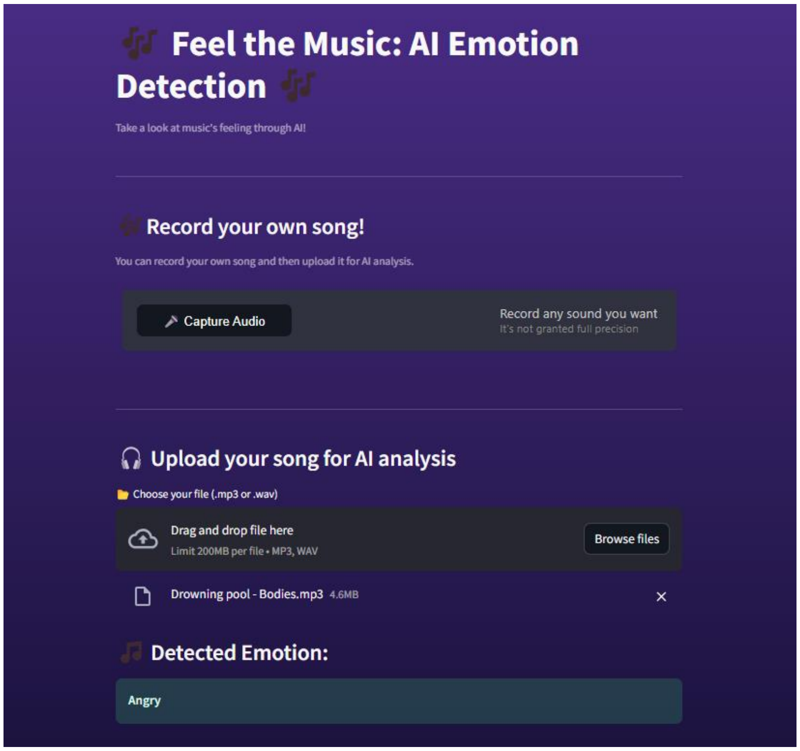
PICTURES SECTION



1-First version of the Web Page



2-Second version of the Web Page



3-Honors version of the Web Page

	precision	recall	f1-score	support
Angry	0.00	0.00	0.00	3
Energetic	0.17	0.25	0.20	4
Happy	0.00	0.00	0.00	2
Romantic	0.00	0.00	0.00	5
Sad	0.25	0.50	0.33	2
accuracy			0.12	16
macro avg	0.08	0.15	0.11	16
weighted avg	0.07	0.12	0.09	16

4-First version of the AI model with 12% accuracy

=== Classification Report ===

	precision	recall	f1-score	support
Angry	0.70	0.64	0.67	11
Calm	0.50	0.23	0.32	13
Energetic	0.27	0.27	0.27	11
Happy	0.40	0.60	0.48	10
Romantic	0.50	0.20	0.29	10
Sad	0.56	0.88	0.68	16
accuracy			0.49	71
macro avg	0.49	0.47	0.45	71
weighted avg	0.50	0.49	0.47	71

5-Honors version of the AI model with 49% accuracy

REFERENCES PAGE

- "Random forest classifier is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting." (SciKit Learn. 2025. *Random Forest Classifier documentation*. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)

- Rosabel Yu Ling Tay, Bee Chin Ng. PlosOne Journal. (2019). *Effects of affective priming through music on the use of emotion words*. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0214482#:~:text=select%20emotion%20words%20after%20exposure,us%20to%20better%20understand%20our

- Yihe Jiang, Maoping Zheng. Frontiers in Psychology. (2024). *The inductive effect of musical mode types on emotions and physiological responses*. https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2024.1414014/full

- Peters, V., Bissonnette, J., Nadeau, D., Gauthier-Légaré, A., & Noël, M. A. (2024). *The impact of musicking on emotion regulation: A systematic review and meta-analysis. Psychology of Music*. https://doi.org/10.1177/03057356231212362

- Alexis Rojas. (2025). *Music emotions classification public GitHub repository. https://github.com/AL3XI5R/Music_emotion_clasification*