CSE343/ECE343 — Machine Learning Monsoon 2021

Moo-Sic (Mood-Based Personalized Song Recommender System)

Mudit Gupta 2020315
Siya Garg 2020577

Srishti Jain 2020543
Sumit Soni 2020136

Abstract

Music is a great stress buster, helps in relaxation, and elevates our mood. Music recommendations can be applied in different areas, such as support of intellectual and physical work, studying,stress and tiredness removal, and many others. Nowadays, music platforms provide easy access to a wide variety of music along with personalized music recommendations based on recently played songs. Additionally, research shows that humans use facial expressions to express what they want to say and the context in which they mean their words. In this project, we aim to recommend songs to users by recognizing their moods using facial expressions while also considering the genres of their interest. All codes can be found here.

1. Introduction

Songs are the best way to cheer someone up. Due to the increasing trend of listening to music, there has been a huge rise in music streaming platforms providing their users access to millions of songs online, with new artists and genres popping up every now and then. However, choosing the appropriate song depending on our mood often becomes tedious and annoying. Hence, we plan to predict the user's current mood using facial recognition and then suggest a song based on the types of music they usually listen to. The input for user mood detection input will be the user's webcam picture. One of the top music streaming platforms currently is Spotify. Thus, we have used it as the basis for our project because of its immense popularity and its publicly accessible data. Using the Spotify data of each user, we aim to create a recommender system that can recommend songs that best suit their interests. Spotify provides us with lots of resources, but because of its rate-limiting feature, it becomes difficult to send multiple requests to Spotify's server since it starts returning error code 429 for our HTTP requests. Thus, we have used the Spotify library[1] to extract information on multiple songs belonging to different genres. Using this data, we aim to give some predefined emotion scores to all the songs under each genre. Once we get the user's mood score and the genre they listen to, we will recommend a matching song emotion score from that genre.

2. Literature Survey

In this section, we go through some of the work done in facial recognition and emotion-based music recommendation system fields.

The paper on the *transfer learning approach for Face Recognition using Average Pooling and MobileNetV2* [2] discusses improvising facial recognition technology. The research focuses on the implementation of 2 different facial recognition model classifiers: Average Pooling and MobileNetV2, and also draws a comparison between the results of both. They employ deep convolutional neural network layers. The model concluded that MobileNetV2 has a high accuracy rate compared to CNN average pooling. Hence we used MobileNetV2 for the classification of the FER-2013 dataset.

The paper *Emotion-Based Music Recommender System* [3] discusses personalized emotion-driven music recommendation systems. It emphasizes the fact that to change or maintain an emotional state of a user, the main function of the system is to search for the nearest music tracks, which are defined by a certain set of music-related attributes. It talks about the models: K-nearest neighbor and random forests (have better accuracy), LSTM to move towards the desired point in emotional space. Reinforcement learning is a handy tool for real-time recommendations. The approach presented in this study is targeted to provide maximum user benefits from the music-listening experience.

3. Dataset

We have two datasets, FER-2013, and a custom-curated Spotify Dataset.

3.1. FER-2013 Dataset

3.1.1. Dataset Description

The data consists of 48x48 pixel grayscale images of faces. The facial expression falls into one of the seven categories (Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral).

3.1.2. Dataset Extraction

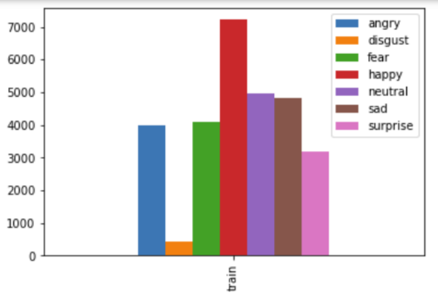
The dataset is publicly available on the Kaggle website [4]. However, we require only two classes, Happy and Sad, for our recommender system. An equal number of images (1000) of Sad and Happy are taken from this dataset.

3.1.3 Preprocessing

The input images are of size 48*48 pixels. As our models work on RGB images and require different input image sizes, we reshaped our images accordingly and extended them to 3 dimensions through duplicacy.

3.1.4 Data Visualization

To remove this class imbalance, we chose 1000 images from the two classes, happy and sad. We also performed some data augmentation to better train the model.



3.2 Spotify Dataset

3.2.1. Dataset Description

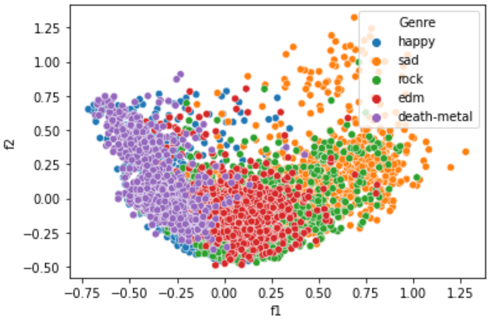
The dataset contains metadata and audio features about the songs. The Meta-Informative features include track id, track name, artist name, album name, Song duration, and Popularity. The Acoustic features include the measure of accousticness, danceability, song duration, energy, instrumentalness, key, liveliness, loudness, mode, speechiness, tempo, time signature, and valence.

3.1.2. Dataset Extraction

We used the Spotipy library to access the Spotify API for song information retrieval. First, we randomly chose 18 genres from a total of 126 unique genres available on Spotify. We obtained information on 18,000 songs using 1000 random songs of each genre. However, after further exploration, we reduced this dataset to just 5000 samples belonging to five genres, which were happy, sad, Rock, EDM, and death-metal, to avoid class imbalance during classification tasks.

We also extracted playlists of 20 users using the Spotipy library, which contains the same features as the above dataset, along with the song IDs.

The scatter plot for the 5k data genres is

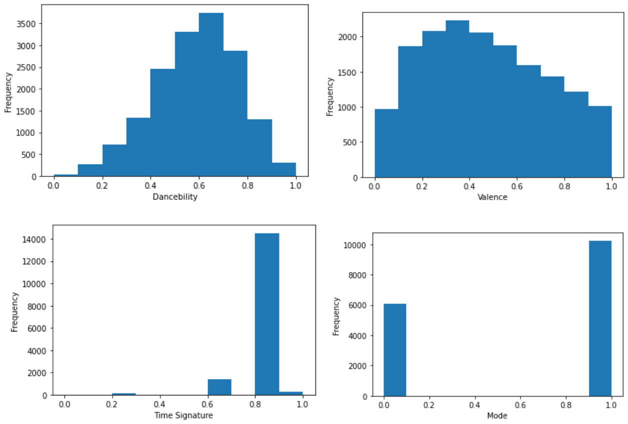


3.2.3 Preprocessing

For preparing the dataset for the model, we first dropped all the duplicates using pandas to avoid redundancy and were left with 16846 unique songs. We then removed all the features(fields) unique to the song (such as Song ID, Song Name, and Album name) and information related to the artist and its release. Further, we visualized the distribution in popularity of songs against all the features and removed all the extreme outliers like the song's duration, more than 10 minutes. We normalized the audio features using min-max normalization on the columns. This normalization technique is quite useful to avoid the biases of certain features with respect to others during model training. We then used this normalized data for feature extraction.

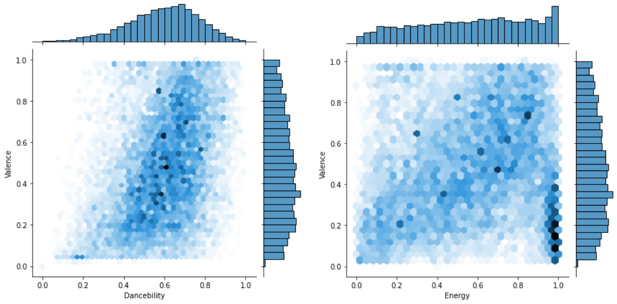
3.2.4 Data Visualization

We plotted the frequency of all the features using histogram plots of the matplot library. All our plots are plotted using matplotlib and seaborn library.



Frequency plots for a) Danceability and b) valence is distributed over a range. On the other hand, c) Time Signature and d) Mode contain only 4 unique values for the entire dataset.

Here, we can see that some features are distributed well over a range, while others mainly had the same value throughout the dataset. Thus, they perform a minimal role in our genre prediction and emotion score task and are mostly dropped during the feature selection task.



Correlation plot between a) Valence and Danceability show high correlation
b) Valence and Energy show high correlation. This is as expected based on the literature.

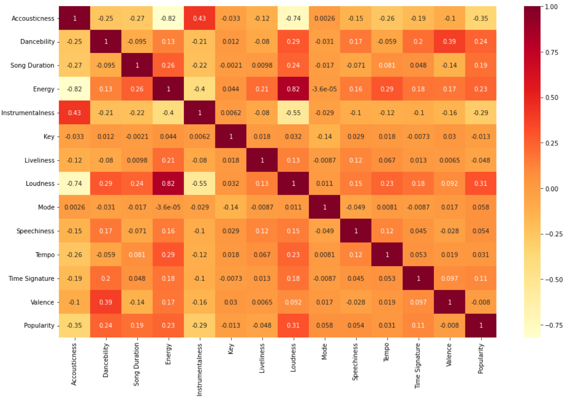
Through our literature review and exploration of the topic, we know that features like valence, danceability, and energy contribute highly to emotion score prediction tasks, which is verified by the correlation graph as shown above.

3.2.5 Feature Selection

As the dataset had a lower dimension, we decided to perform feature selection instead of feature extraction. We used the following methods to distinguish which features greatly affected the model's predictions and which features did not and could be dropped from the dataset.

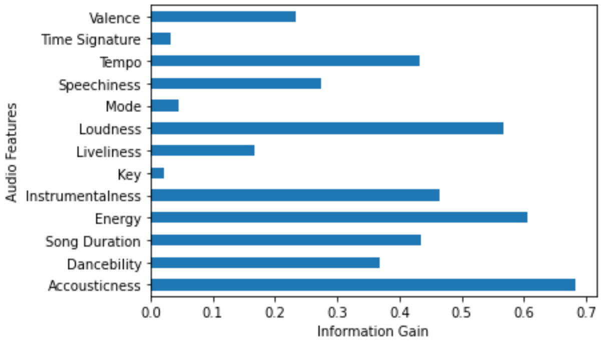
3.2.5.1 Correlation Coefficient

It measures the linear relationship between two or more variables. It helps in deciding the features which are largely correlated with each other and can be dropped after determining their correlation with the target variable and therefore help in feature selection.



3.2.5.2 Information Gain

Information gain for each variable is calculated in the context of the target variable and is used for feature selection. It is calculated by subtracting the weighted entropy for each variable from the original entropy. The higher the information gain, the greater the decrease in entropy. We used sklearn's inbuilt library to perform this task.



Looking at the information gains, we dropped the columns: Key, Mode, Song Duration, and Time Signature.

4. Methodology

Our project mainly contains three components: *Human Face Mood Score, Music Emotion Scores, and Playlist Genre Classification Task*. Merging this, we aim to create our Music Recommendation System.

4.1 Face Mood Score

The main objective of the facial recognition model is to correctly classify the images into 2 classes: *Happy and sad*. The input image sizes are changed as per model requirements, and data augmentation is applied. Different pre-trained CNN models are finetuned to our requirements. We have used Categorical cross-entropy loss and adam optimizer with a learning rate of 0.001.

VGG16 model: uses very small convolution filters and has approximately 138 trainable parameters.

Inception_V3 model: It has 42 layers and a lower error rate than its predecessors.

MobileNetV2 model: Contains two blocks. The first block has a stride of 1, while the next has a stride of 2.

We have defined the mood score of the user as a metric to help define the user's emotion and accordingly recommend a song based on the scoring. The mood score is the result of output generated by the softmax layer.

4.2 Music Emotion Score

Since no datasets available provide labels for emotions based on the above-discussed feature set, we have assumed the songs belonging to the 'happy' and 'sad' genres to also belong to 'happy' and 'sad' emotions, respectively. We trained different models to classify songs of other genres as happy or sad emotionally. The *Emotion Score* of a song has been defined as the softmax probability of the song belonging to the ‘happy’ genre. Multiple classification models like Logistic Regression, Multinomial Naive Bayes (MnB), SGD Classifier, K-Nearest Neighbor (KNN), Decision Tree Classifier, Random Forest Classifier (RFC), Support Vector Classifier (SVC), and Multi-Layer Perceptron (MLP) were used to classify the emotion of the songs. Of these seven, the best results were observed in *SVC* and *MLP*. Further, a grid search was performed on both models to get better hyperparameter values. The best-estimating hyperparameters for each came to be:

Support Vector Classifier (SVC): C = 0.5, degree = 2, probability=True, kernel='rbf', gamma='auto'

Multi-Layer Perceptron (MLP): max_iter = 1500, alpha = 0.005, learning_rate_init = 0.01, random_state = 42, solver = 'sgd'.

Finally, **MLP** was used since it gave better results than fine-tuned SVC.

4.3 Playlist Extraction and Genre Classification

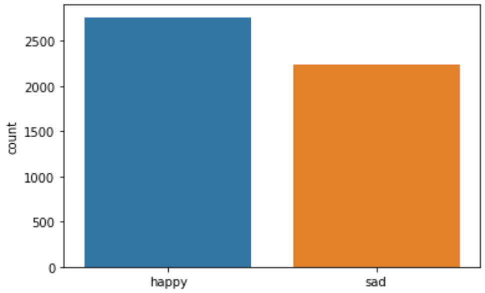
We used our custom Spotify Client API to access a user’s playlist and got a list of the audio features of all the songs in this playlist. Our next aim was to classify each of these newly retrieved songs into a genre. This task was to retrieve which genre of songs the user generally listens to.

We had the genres of the initial custom dataset and used them for training purposes. We used the same seven classification models as the previous task. Of these seven, the best results were observed in MLP and RFC. Further, a grid search was performed on both models to get better hyperparameter values. The best-estimating hyperparameters for each came to be

Random Forest Classifier (RFC): criterion = ‘gini’, max_features = 'sqrt', n_estimators = 160.

Multi-Layer Perceptron (MLP): max_iter = 700, alpha = 0.005, learning_rate_init = 0.01, random_state = 42, solver = 'lbfgs'.

Finally, **RFC** was used since it gave better results than fine-tuned MLP.



Almost Balanced Class distribution was achieved

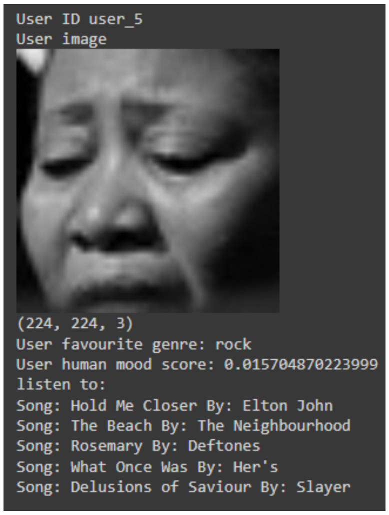
4.4 Recommendation Model

Using our music emotion score model. We gave the score to all the songs in our custom dataset. These songs are directly used in the recommendation model according to their scores. The recommendation model is a python script that selects the next song based on a pre-defined heuristic.

4.5 Pipelining

We have created a pipeline combining all the individual components into a final product. Input for this pipeline consists of capturing a live image of the user and getting their Spotify user ID. The output consists of a custom-curated playlist of recommended songs based on the user’s genre interests derived from their existing playlists and their current mood score.

Here’s an example of how the pipeline works for a user image and user ID:



```
User ID user_5
User image
(224, 224, 3)
User favourite genre: rock
User human mood score: 0.015704870223999
listen to:
Song: Hold Me Closer By: Elton John
Song: The Beach By: The Neighbourhood
Song: Rosemary By: Deftones
Song: What Once Was By: Her's
Song: Delusions of Saviour By: Slayer
```

5. Result

5.1 Face Mood Score

The accuracy of InceptionV3 came out to be greater than both Vgg16 and MobileNetV2 for the same transfer learning operations on all the models. Hence, we will proceed with using this model for the final pipeline.

The best accuracy came out for *InceptionV3,* so we are using this model for the final training and prediction.

Model Name	Training Accuracy	Testing Accuracy
InceptionV3	**0.96**	**0.91**
Vgg16	0.90	0.86
MobileNetV2	0.85	0.82

5.2 Music Emotion Score

Best accuracy came out for *Multi-Layer Perceptron* in the Music score prediction task.

Classification Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.9525	0.9535	0.9524	0.9525
Multinomial Naive Bayes	0.9275	0.9298	0.9277	0.9274
SGD Classifier	0.9500	0.9524	0.9498	0.9499
K-Nearest Neighbor	0.9475	0.9478	0.9474	0.9475
Decision Tree	0.9400	0.9402	0.9399	0.9400
Random Forest Classifier	0.9525	0.9540	0.9524	0.9524
SupportVector Classifier	0.9525	0.9540	0.9524	0.9524
Multi-Layer Perceptron	**0.9525**	**0.9535**	**0.9524**	**0.9525**

5.3 Playlist Genre Prediction

Best accuracy came out for *Random Forest,* so we are using this model for the Music Mood scoring task.

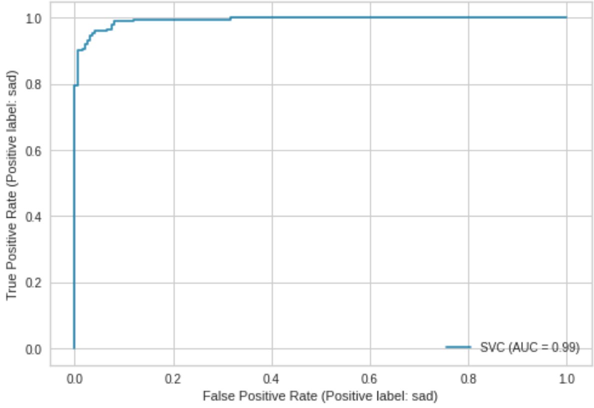
Classification Model	Accuracy	Precision	Recall	F1
Logistic Regression	0.7080	0.7112	0.7084	0.7094
Multinomial Naive Bayes	0.5480	0.5712	0.5600	0.5486
SGD Classifier	0.6573	0.6768	0.6658	0.6330
K-Nearest Neighbor	0.7120	0.7162	0.7111	0.7094
Decision Tree	0.6987	0.6983	0.6960	0.6969
Random Forest Classifier	**0.8093**	**0.8068**	**0.8091**	**0.8072**
Support Vector Classifier	0.7187	0.7199	0.7184	0.7186
Multi-Layer Perceptron	0.7547	0.7529	0.7549	0.7536

6. Analysis

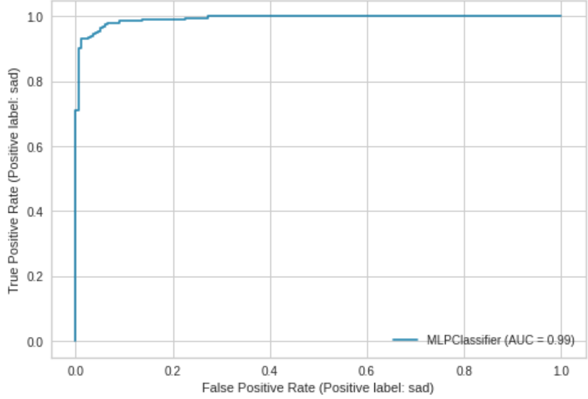
The Loss curve is a good representation of the training rate of the model. As we can see, MLP achieves very little loss in very few iterations. However, an even better metric is the AUC-ROC.

AUC - ROC curve is a performance measurement for classification problems at various threshold settings. The ROC curve is plotted with TPR against the FPR where TPR is on the y-axis and FPR is on the x-axis.

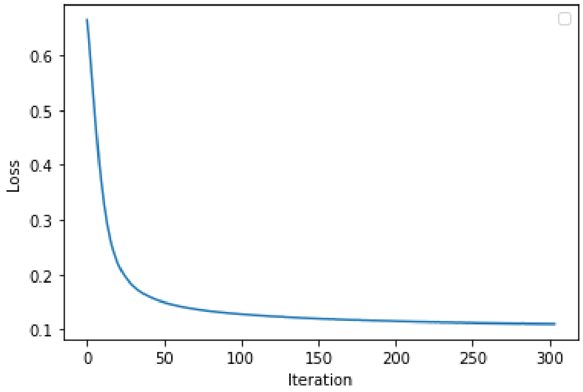
6.1 Music Emotion Scoring



AUC-ROC plot for SVC for Music Emotion Classification Task



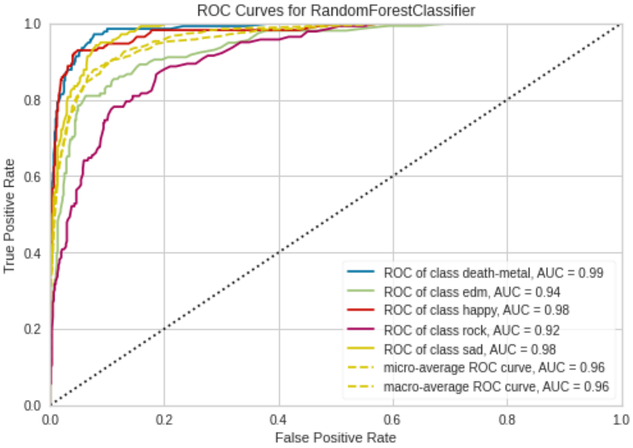
AUC-ROC plot for MLP for Music Emotion Classification Task



Loss Curve for MLP for Music Emotion Classification Task

As we can see from the AUC-ROC plots, MLP performs similar to SVC for the Music Emotion Classification task.

6.2 Playlist Genre Prediction



AUC-ROC plot for MLP for Genre Classification Task

We observe that the classification models perform much better during a binary classification task compared to multi-class one. Also, true positive rates are much higher when there are just two classes compared to five.

6.3 Recommender Model

As we haven't tested our pipeline model for mass users, we don't have a real metric on how well our recommendation system is working. We need more personalized and user-defined metrics.

7. Conclusion

7.1. Learnings from the Project

Through this project, we got hands-on experience in designing a pipeline for a machine learning project. We realized that collecting an accurate and adequate amount of data is equally important as developing and training the final machine learning model. We played around with the Spotify API, using which we extracted the data and prepared the dataset on our own. We became acquainted with different tools for data visualization, which helped us analyze different trends in our data. We learned how mutual information classification works and can help in feature selection. We trained various transfer learning and classification models for our different tasks.

7.2. Future Work

We have completed the project as proposed, following the tentative timeline we had given. We curated the datasets, trained and tested a model to predict human mood, stored song emotion scores for 5k songs, and classified the song's genre of different users' playlists. In the future, we plan to work on making this into a full-fledged deployable app or web extension. Also, we aim to test this pipeline with multiple users and improve our model.

7.3. Member Contribution

- **Mudit Gupta:** Literature review, Data Extraction and Collection, Extraction and EDA of Spotify dataset, Analysis and inference of the data and results, Extraction of user's Spotify playlist, and making Genre prediction classification model.
- **Siya Garg:** Literature review, Data Extraction and Collection, EDA, feature selection, and data augmentation of FER 2013 dataset. Analysis and inference of the data and results, song emotion score model, and final pipeline for recommender system.
- **Srishti Jain:** Literature review, Data Extraction, and Collection, Extraction and EDA of Spotify dataset, Analysis and inference of the data and results, song emotion score model, the final pipeline for recommender system.
- **Sumit Soni:** Literature review, Data Extraction, and Collection of working Facial recognition model, Analysis, and inference of the data and results, creating a pipeline for capturing images and predicting facial mood score.

References

[1] Paul Lamere, license for the private use of spotipy library.

*[2] F.M. Javed Mehedi Shamrat, Sovon Chakraborty, M. M. Imran, Abdulla, Ishtiak Ahmed, *Ankit Khater. A Transfer Learning Approach for Face Recognition using Average Pooling and MobileNetV2*

[3] Mikhail Rumiantcev, Oleksiy Khriyenko, University of Jyväskylä Jyväskylä, Finland. Emotion-Based Music Recommender System.

[4] https://www.kaggle.com/datasets/msambare/fer2013