**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, relaxing, stress and tiredness removal, music therapy, 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.*

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](https://github.com/spotipy-dev/spotipy/blob/master/LICENSE.md)] 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.

1. **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](https://www.researchgate.net/publication/363634208_A_Transfer_Learning_Approach_for_Face_Recognition_using_Average_Pooling_and_MobileNetV2)] 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](https://fruct.org/publications/acm26/files/Rum.pdf)] 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.

1. **Dataset**

We have 2 datasets, FER-2013 and 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)

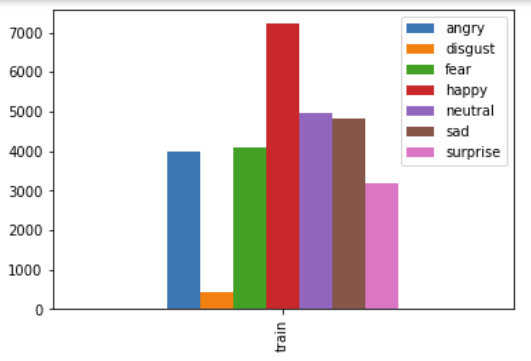
**3.1.2. Dataset Extraction**

The dataset is publicly available on the Kaggle website [[4](https://www.kaggle.com/datasets/msambare/fer2013)]. However, we require only 2 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.

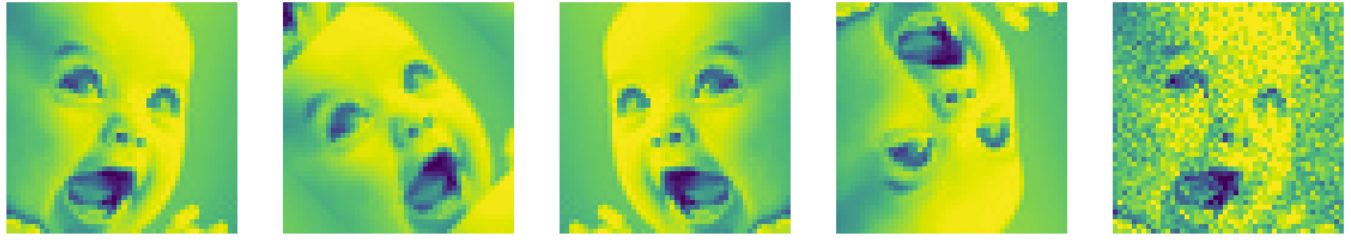
**3.1.4 Data Visualization**



This plot tells about the imbalanced class distribution in the FER dataset. To have a class balance, we chose 1000 images from the 2 classes, happy and sad.

Data augmentation increases the amount of data by adding slightly modified copies of already existing data. We did this to reduce [overfitting](https://en.wikipedia.org/wiki/Overfitting) when training a machine learning model.

****

****

**3.2 Spotify Dataset**

**3.2.1. Dataset Description**

The dataset contains some metadata and various 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.

| Feature Name | DataType |
| --- | --- |
| Song ID | String |
| Song Name | String |
| Artist Name | String |
| Album Name | String |
| Release Date | Integer |
| Acousticness | Float |
| Danceability | Float |
| Song Duration | Integer |
| Energy | Float |
| Instrumentalness | Float |
| Key | Integer |
| Liveliness | Float |
| Loudness | Float |
| Mode | Integer |
| Speechiness | Float |
| Tempo | Float |
| Time Signature | Integer |
| Valence | Float |
| Popularity | Integer |

**3.1.2. Dataset Extraction**

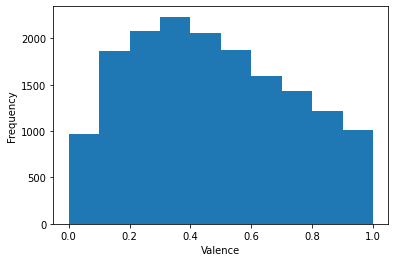
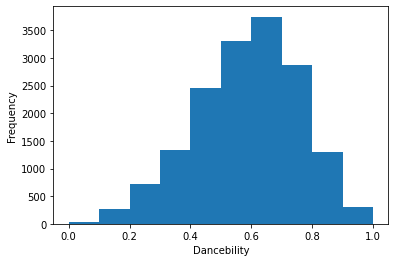
We used the Spotipy library to access the Spotify API for song information retrieval. First, we randomly chose 18 genres from a total of 126 unique genres available on Spotify. We obtained information on 18,000 songs using 1000 random songs of each genre. Since Spotify has recently restricted the offset limit to 1000, we could not extract more songs using our query.

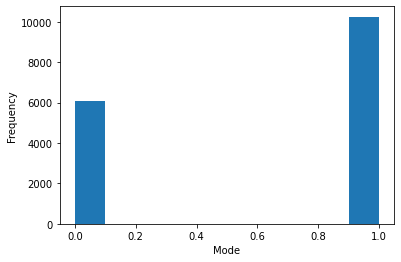
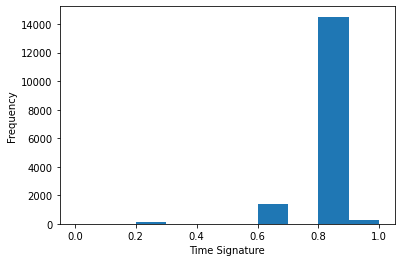
**3.2.3 Preprocessing**

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) which were 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 will normalize 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. This normalization technique is quite useful to avoid the biases of certain features with respect to others during model training.

**3.2.4 Data Visualization**

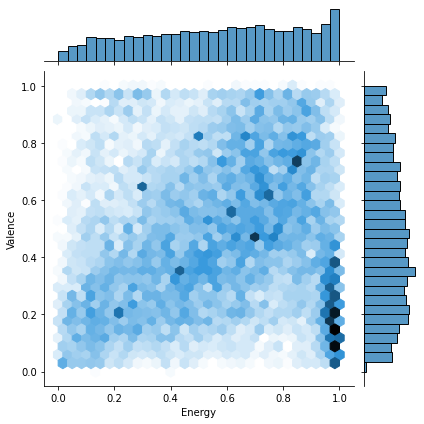
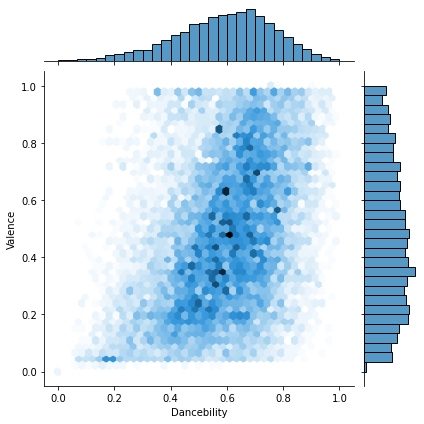
We plotted the frequency of all the features using histogram plots of the matplot library and got various plots. All our plots are plotted using matplotlib and seaborn library.





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.

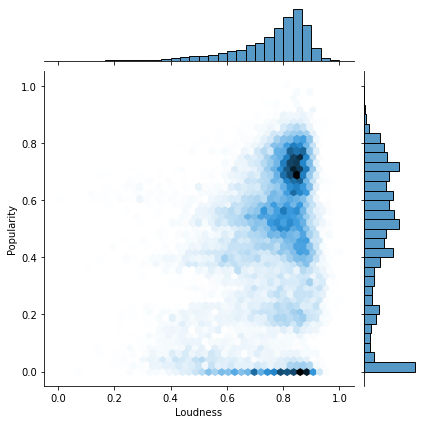
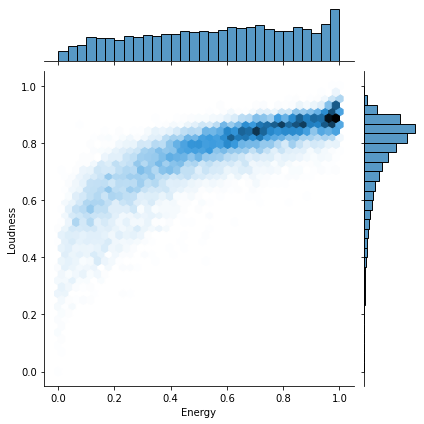
Here, we can see that some features are distributed well over a range, while others mainly had the same value throughout the dataset. Thus, they perform a minimal role with our genre prediction and emotion score task and hence are mostly dropped during the feature selection task.



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. Some other interesting observations between relevant features is seen in the below images.



a) We can see a log-like relationship between Loudness and Energy

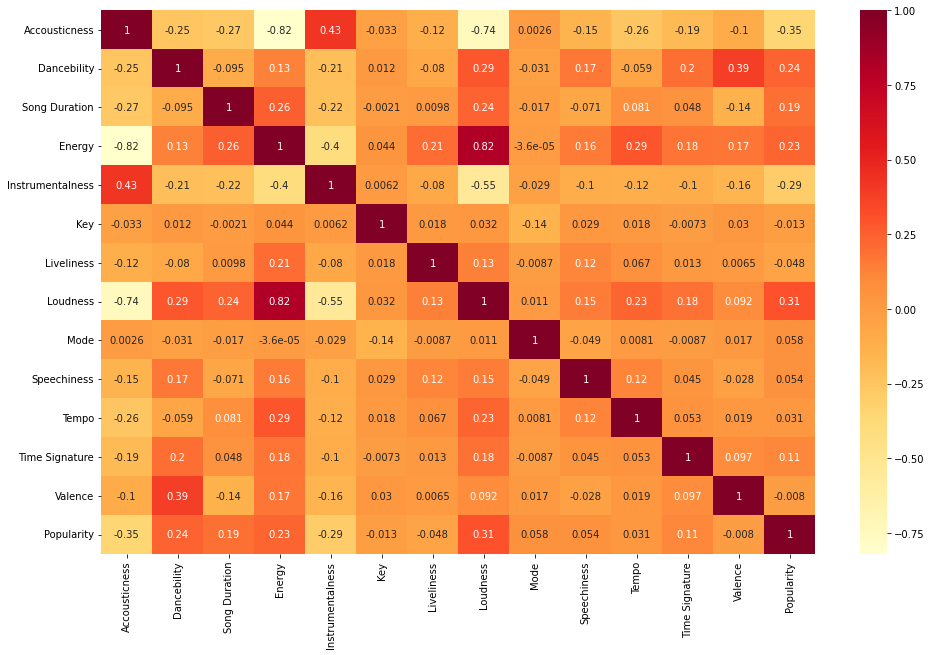
b) An unexpected relation between Popularity and Loudness since high loudness corresponds to high as well as low popularity. This will be useful in our song recommender model.

**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 play a great role and could be dropped from the dataset.

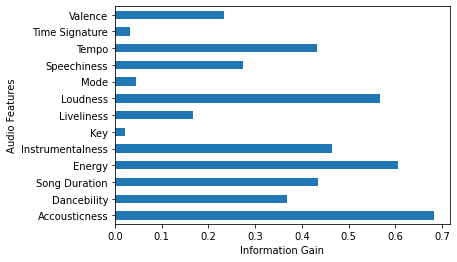
**3.2.5.1 Correlation Coefficient**

It is a measure of the linear relationship between two or more variables. It helps in predicting a variable based on the value of another variable. 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.



3.5.2 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 is the decrease in entropy. We used sklearn’s inbuilt library to perform this task.



1. **Methodology**

The main objective of the facial recognition model is to correctly classify the images into 2 classes: Happy and sad. This is achieved through CNN models. The input image sizes are changed as per model requirement, and data augmentation (rotation, shifting, flipping) is applied. The validation set is created in an 85:15 ratio to prevent overfitting. Different inbuilt CNN models are trained. This was downloaded and used for prediction. Categorical\_crossentropy loss is used for model training along with adam optimizer with a learning rate of 0.001. Due to computational power constraints, fine-tuning wasn’t implemented.

*Transfer learning:*A machine learning technique where a model trained on one task is re-purposed on a second related task, allowing rapid progress or improved performance when performed on another task.

*VGG16 model:* very small (3 × 3) convolution filters are used. It pushed the depth to 16–19 weight layers, making it approximately 138 trainable parameters.

*Inception\_V3 model:*It has a total of 42 layers and a lower error rate than its predecessors.

*MobileNetV2 model:* There are two types of blocks. One is a residual block with a stride of 1. Another one is a block with a stride of 2 for downsizing.

A python script is made, which takes 5 images of the user with a delay on each image of 3 sec.

1. **Result and Analysis**

The accuracy of VGG16 came out to be greater than both InceptionV3 and MobileNetV2. Having Vgg16 having the best accuracy while applying the same transfer learning operation on all the models, we will proceed with using this model later.

*Accuracy of Vgg16:* **88.5%** on Validation Set

**89%** on Testing Set

*Accuracy of Inception\_V3:* **84.5%** on Validation Set

**82%** on Testing Set

*Accuracy of MobileNetV2*: **79.5%** on Validation Set

**78%** on Testing Set

1. **Conclusion**

**6.1. 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.

**6.2. Future Work**

To date, we managed to follow the tentative timeline we had proposed. We have curated the datasets, trained and tested a model for mood prediction based on facial data input, and received a satisfying result. In the future, we plan to work on

1. Hyper-parameter tuning along with the comparison-based analysis for different algorithms for the music scores.
2. Creating a recommender system that can suggest songs based on their interests.
3. Creating a pipeline that can aggregate both results to develop a final list of recommended songs.

**6.3. Member Contribution**

* **Mudit Gupta:**  Literature review, Data Extraction and Collection, Extraction and EDA of Spotify dataset, Analysis and inference of the data and results.
* **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.
* **Srishti Jain:** Literature review, Data Extraction and Collection, Extraction and EDA of Spotify dataset, Analysis and inference of the data and results.
* **Sumit Soni:** Literature review, Data Extraction and Collection, working Facial recognition model, Analysis, and inference of the data and results.

**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*