Moodify

ALIAKBAR RZAYEV
ABDULLAH ORZAN
GUNDA ESRA ALTINISIK
YIGIT DINCER

miuul

DATA SCIENCE& MACHINE LEARNING
BOOTCAMP

The Idea

The main idea of this project is to classify songs not only based on their lyrical and musical features, but also incorporating emotions, in order to provide users with more successful recommendation outputs. Moodify aims to categorize songs into four main emotions and utilize the similarities in musical features within these categories to offer users more effective recommendations, weighting them with emotions.

Creating Labelled Dataset

At this stage of the project, we retrieved the songs from the mood playlists with the highest number of listens and likes, as determined by Spotify and its users, using the Spotify API. We transformed these songs into a data frame format.

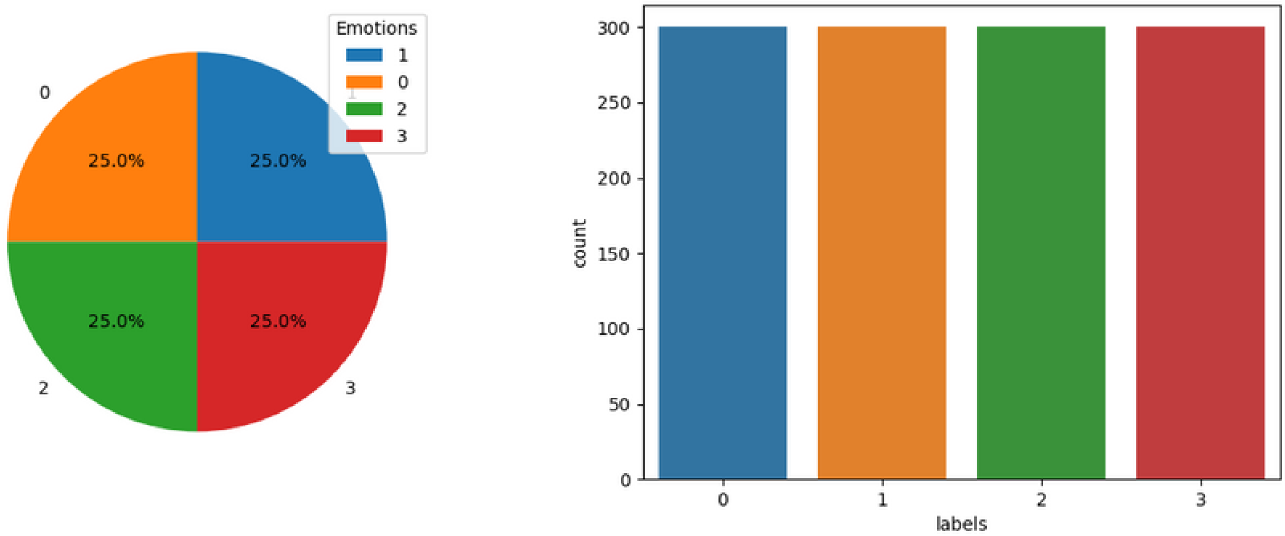
	track	artist	duration (ms)	popularity	uri	danceability	energy	key	loudness	speechiness	acousticness	instrumentalness	liveness	valence	tempo	time signature	labels
0	1999,00	Prince	379266	68	spotify:track:	0,87	0,73	5	-8,20	0,08	0,14	0,00	0,08	0,63	118523,00	4	1
1	23,00	Blonde Redhead	318800	43	spotify:track:	0,38	0,83	8	-5,07	0,05	0,02	0,20	0,15	0,17	120255,00	4	0
2	9 Crimes	Damien Rice	217946	60	spotify:track:	0,35	0,14	0	-15,33	0,03	0,91	0,82	0,09	0,12	136168,00	4	0
3	99 Luftballons	Nena	233000	2	spotify:track:	0,47	0,44	4	-12,86	0,06	0,09	0,56	0,11	0,59	193,10	4	1
4	A Boy Brushed	Underoath	268000	60	spotify:track:	0,42	0,93	1	-3,60	0,11	0,00	0,39	0,14	0,45	16988,00	4	2
5	A Burden to B	Emmanuelle Rimbaud	129410	27	spotify:track:	0,39	0,05	6	-2,64	0,07	1,00	0,96	0,11	0,17	71241,00	5	3
6	A La Plage	Ron Adelaar	141888	29	spotify:track:	0,50	0,06	5	-20,46	0,05	0,99	0,96	0,12	0,55	134209,00	4	3
7	A Little Less C	Elvis Presley	211173	10	spotify:track:	0,59	0,94	4	-9,39	0,05	0,00	0,28	0,16	0,58	114997,00	4	1
8	A Place for M	Linkin Park	184640	68	spotify:track:	0,60	0,91	11	-5,25	0,18	0,01	0,42	0,67	0,46	133063,00	4	2
9	ATTACK	Thirty Seconds To Mars	189200	7	spotify:track:	0,33	0,88	5	-1,89	0,06	0,00	0,00	0,73	0,30	175009,00	4	2
10	Adagio For St	Lucas & Steve	197189	55	spotify:track:	0,35	0,79	10	-6,70	0,07	0,00	0,43	0,22	0,04	127139,00	4	2

Based on the Robert Thayer's traditional model of mood, we continued to expand the dataset by maintaining a statistical balance in the specific values of songs with certain emotions, up until the point of diminishing returns.

Mood of Western songs	Mean Intensity	Mean Timbre	Mean Pitch	Mean Rhythm
Happy	0.2055	0.4418	967.47	209.01
Exuberent	0.317	0.4265	611.94	177.7
Energetic	0.4564	0.319	381.65	163.14
Frantic	0.2827	0.6376	239.78	189.03
Sad	0.2245	0.1572	95.654	137.23
Depression	0.1177	0.228	212.65	122.65
Calm	0.0658	0.1049	383.49	72.23
Contentment	0.1482	0.2114	756.65	101.73

Mean values for audio features found across moods. Source: Derived from Bhar (2014).

We completed the training set with a total of 1200 songs, which yielded the best results for both the train and test outputs. The final version of this training set contains 300 songs each from the four main emotions we defined. These songs can be found in the '1200_song_mapped.csv' file.



Pie chart and BAR plot of the dataset.

Exploratory Data Analysis

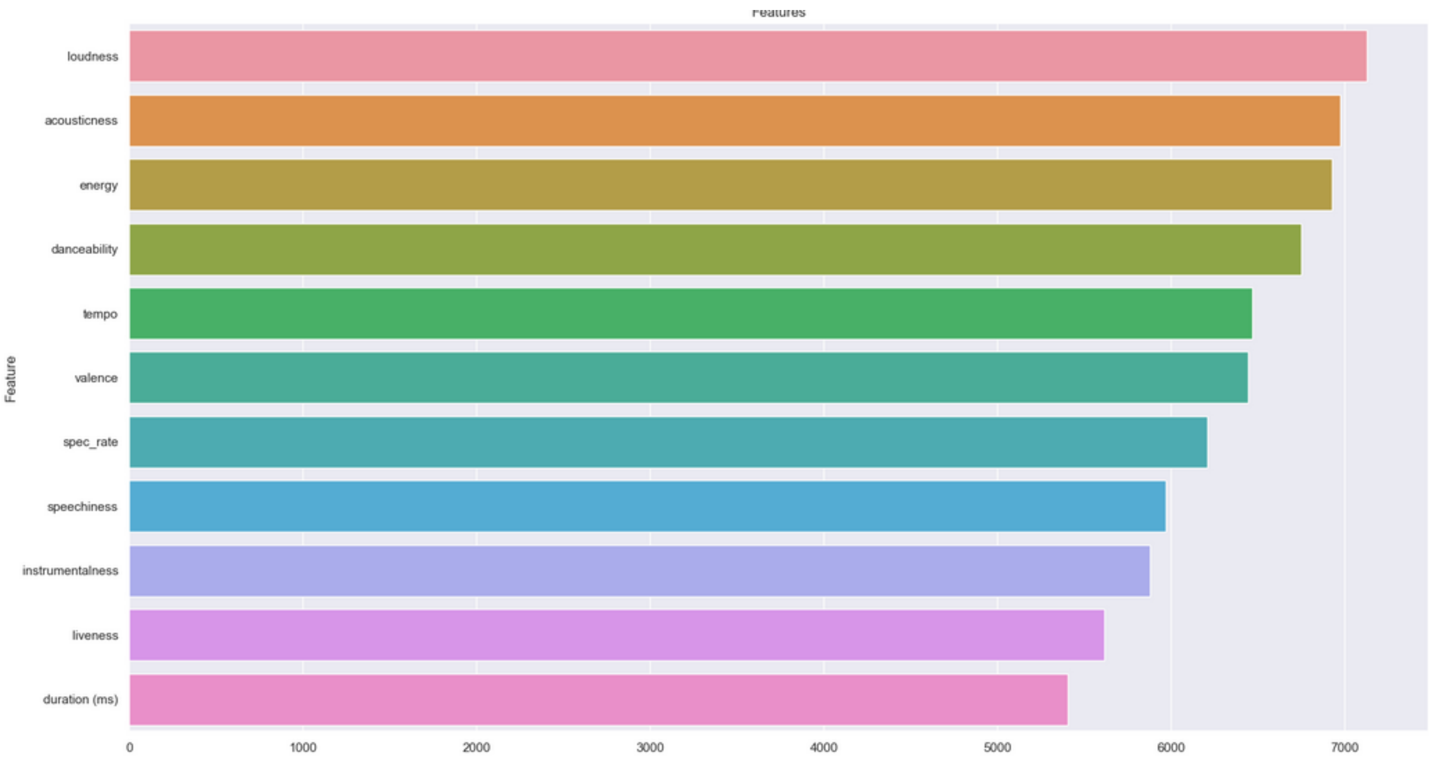
This stage of the project focused on identifying the key factors that determine the emotion of a song. Therefore, all variables in the dataset were examined with different songs and values for various emotions in order to establish sharper frameworks for emotions. As a result of the study, inconsistencies were detected in tempo, valence, and energy variables between energetic and happy songs.

Recommendations were generated using cosine similarity based on the 4-8-10 main variables that we believe determine the emotions of the songs, and the recommendation results were analyzed. You can find code implementations for EDA in 'main.py' file and test results of recommendation tests in 'cosine_similarity_test.py' file.

Feature Engineering

In the Feature Engineering phase, as we did not leave any missing values or outliers when constructing the dataset, we focused on feature extraction. Although we created new features based on multiple music theories, we observed that many of them performed poorly according to model importance outputs.

As a result, we completed this process by including the speech_rate feature, which represents the total vocal information in the song and dropping popularity, time signature and key features. We preferred to adhere to music theory as operations such as splitting, subtracting, adding, and multiplying that we tried with the features yielded unfavorable results in terms of correlation and model training. We used Robust Scaler for scaling the dataset and do not use any encoder because of we worked with only numerical variables.



Feature importances of final model.

Model and Hyperparameter Optimization

BASE MODELS	
LR	0,5725
KNN	0,3883
SVS	0,3967
CART	0,7242
RF	0,3983
Adaboost	0,7058
GBM	0,8042
XGBoost	0,8050
LightGBM	0,8142

Accuracy scores of models.

In the Model phase, we tested nine different models with the training set and obtained the best result using the LGBM model.

Additionally, we conducted detailed hyperparameter tuning with Random Forest and CART models, and performed an Ensemble Learning experiment using LGBM, RF, and CART.

During the process of examining accuracy, F1 score, recall, and precision values, we ultimately decided to choose LGBM as the final model.

After the hyperparameter optimization we get ~%83 accuracy score with LGBM model.

Test and Recommendation Dataset Creation

With the model training phase completed, we labeled the emotions of nearly 278,000 songs we obtained from Spotify, ensuring that it includes up-to-date songs. This allowed us to create a recommendation dataset.

We conducted further exploratory data analysis (EDA) on this test dataset to examine its consistency. Subsequently, using this dataset, we performed recommendation experiments using cosine similarity with weighted recommendations. You can find labelled dataset in '278k_song_labelled.csv" file.

Deployment

We used Streamlit to deploy the project. The application we developed takes the Spotify link of a song as input from the user and displays the most relevant predictions along with the song's duration, cover art, and release date. The user can easily open the song on Spotify by clicking on the cover art. We used Python, CSS, and HTML languages in the development of the application.

During the deployment phase, due to a version mismatch with the Spotify API v1, we prepared an exception message for cases where the recommendation count for certain songs is unusually low or non-existent.

To run the application, you need to add your Client ID and Secret ID values from Spotify to the 'create_access_token()' function in the 'Streamlit_app.py' file.

References

[1] Nuzzolo, M. (n.d.). Music mood classification. Electrical and Computer Engineering Design Handbook. https://sites.tufts.edu/eeseniordesignhandbook/2015/music-mood-classification/

[2] Cyril Laurier and Perfecto Herrera [2007], Audio Music Mood Classification Using Support Vector Machine. In Proceedings of the International Conference on Music Information Retrieval, Vienna, Austria.

[3] Lu, Liu and Zhang (2006), Automatic Mood Detection and Tracking of Music Audio Signals. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, # 1, January 2006