Mood-Based Song Classifier and Recommender System

Harshil Mital (2021050) Harsh Parimal Popat (2021048)
Harshvardhan Singh (2021052) Ashwin R Nair (2020037)

10-10-2023

1 Abstract: Motivation

By creating a Mood-Based Song Classifier and Recommender System, we aim to leverage machine learning to predict users' moods based on previous songs and suggest appropriate songs. This enhances user experience, personalized music consumption, and offers insights into the emotional impact of music. The idea emerged from recognizing the powerful link between emotions and music preferences, driving us to build a practical system that enriches people's interactions with music on a daily basis.

2 Introduction Describing problem statement

Music, a universal language, deeply resonates with our emotions. In today's digital age, while users have countless songs at their disposal, selecting one that matches their mood remains challenging. This vast choice can lead to decision paralysis. Also, almost all platforms suggest songs from the user's liked songs or their general listening trends, and none consider the emotional component of song selection. Addressing this gap, our project introduces a Mood-Based Song Classifier and Recommender System. Utilizing machine learning, we aim to discern users' moods from their recent song choices, tailoring recommendations to their emotional state. This not only provides a personalized listening experience but also offers insights into the intricate relationship between emotions and musical preferences. We aspire to redefine users' daily interactions with music platforms, ensuring each song recommendation mirrors their emotional journey.

3 Literature Survey

SVR-based music mood classification and context-based music recommendation by Seungmin Rho , Byeong-jun Han and Eenjun Hwang. Link
This paper focuses on context based music recommendation. The authors first classified the mood after converting it into a regression problem based on support vector regression (SVR). For music recommendation, they assess the user's mood and situation using both collaborative filtering and ontology technology.

An Emotional Recommender System for Music by Vincenzo Moscato, Antonio Picariello and Giancarlo Sperli. Link
In this paper, the authors describe a novel music recommendation technique based on the

identification of personality traits, moods, and emotions of a single user. It embeds users' personality and mood with a content-based filtering approach to obtain accurate and dynamic results.

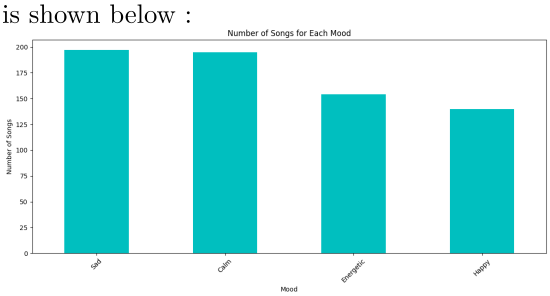
Moodplay: Interactive Mood-based Music Discovery and Recommendation by Ivana Andjelkovic, Denis Parra, John O'Donovan. Link
MoodPlay is a hybrid recommender system music which integrates content and mood-based filtering in an interactive interface. MoodPlay allows the user to explore a music collection by latent affective dimensions, by integrating user input at recommendation time with predictions based on a pre-existing user profile.

4 Dataset: Dataset details with data prepossessing techniques.

4.1 Dataset Features

We have picked Spotify Music data which contains 686 songs different artists and multiple genre. All the songs features are extracted using Spotify API. These provide insights into the characteristics and mood of songs by metrics such as tempo, energy, valence, and danceability. Feature listed down in the below table

The mood distribution of songs in dataset is shown below :



Feature	Type
Name	String
Album	String
Artist	String
Id	String
Release_date	Date
Popularity	Integer
Length	Integer
Danceability	Float
Acousticness	Float
Energy	Float
Instrumentalness	Float
Liveness	Float
Valence	Float
Loudness	Float
Speechiness	Float
Tempo	Float
Key	Integer
Time Signature	Integer

Raw Features

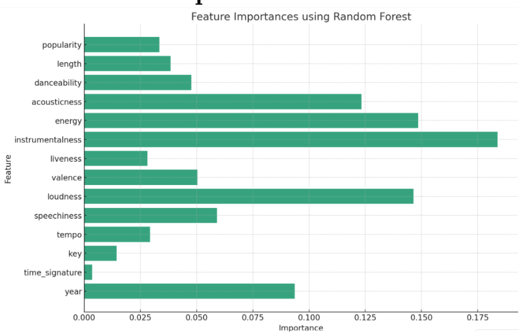
4.2 Dataset Preprocessing

We began by checking for any missing or null values in our dataset and found that there were none. Categorical Features such as 'name', 'album', 'artist', and 'id' are dropped as they wouldn't contribute significantly the models prediction.
Numerical Features are standardised to ensure they have a mean of 0 and a standard deviation of 1, which is crucial for machine learning algorithms that are sensitive to feature scales, such as gradient descent-based methods, SVM, and k-nearest neighbours.
Also, the categorical target label 'mood' is label encoded. This is essential because certain algorithms like XGB Classifier require categorical target labels to be integer-encoded.

4.3 Feature Selection

A Random Forest classifier is used to evaluate the importance of each feature in predicting the target variable 'mood'. Random Forest, being an ensemble of decision trees, inherently provides feature importance scores based on the average impurity reduction of the feature across all trees.

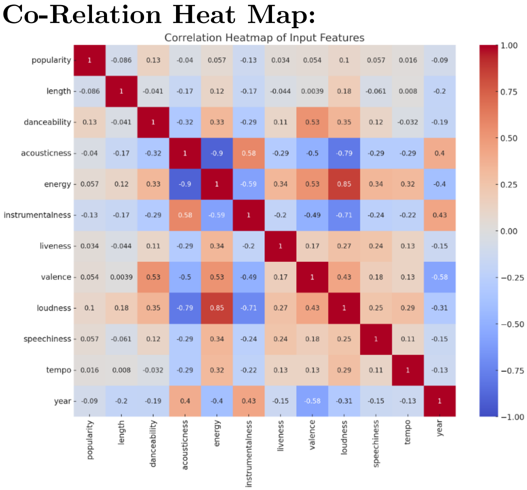
Feature Importance:



From the above graph, we observe that based on the importance scores and our analysis, we decided to drop 'time signature' and 'key' from our dataset to potentially improve our model's performance and reduce the risk of overfitting.

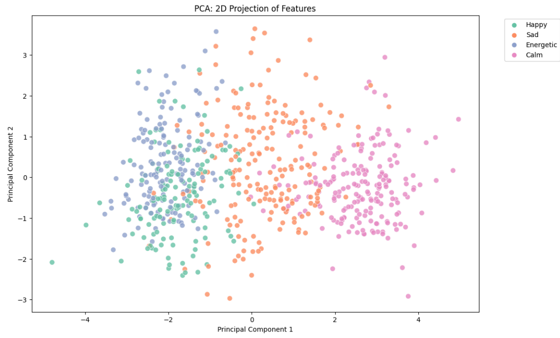
4.4 Data Visualisation

Co-Relation Heat Map:

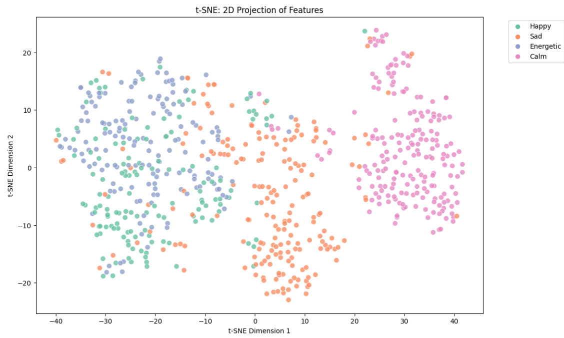


We plotted the correlation heatmap for numerical features and observed that There's a positive correlation between danceability and valence. This suggests that songs that are more suitable for dancing tend to have a more positive mood. Similarly, Acousticness and Energy have a negative correlation which suggests that songs with higher acousticness values tend to have lower energy.

PCA



T-SNE



Observations: Firstly, we can see from the PCA and TSNE plots, that moods are clearly separable (in separate clusters) except for energetic and happy songs, where there is a significant amount of overlap. This was expected as most energetic songs are happy also.

5 Methodology, model details

Our objective is to classify the mood of a person based on the song he/she is listening to and then generate a recommender system which recommends songs to user based on the mood detected. We have tried different ML models for classification and did feature selection to know which all features are important for classification.

Our project on the Song Recommender System is progressing, but we still have significant work ahead.

By integrating these methods, we aim to enhance the accuracy and relevance of our song recommendations, ensuring a better user experience.

5.1 Classification

We have run most of the classification model to find out which one is most important for dataset and have found the accuracy, precision and recall for those models.
We used the following models by dividing the data into 70:20:10 for training testing and evaluation: Logistic regression, SGD Classifier, Gaussian Naive Bayes, Decision Tree, Random Forest, SVM Linear and XGB Classifier.
Logistic Regression: A linear model for classification that estimates probabilities using a logistic function. We have used sigmoid function with cross entropy loss and L2 Penalty(Ridge)
SGD Classifier: SVM linear classifier fitted with Stochastic Gradient Descent with L2 regularization and $\alpha = 0.0001$.
Gaussian Naive Bayes: Probabilistic classifier based on Bayes' theorem, assumes that features have a Gaussian distribution.
Decision Tree: A tree structure where nodes represent features and branches represent decision rules. Splits are determined by minimizing entropy, thus maximizing information gain.
Random Forest: An ensemble of decision trees, trained on random subsets of data and aggregated for predictions.
XGB Classifier: Gradient boosting framework that uses decision trees, optimized for performance.
SVM: Classifier that constructs hyperplanes in a high-dimensional space to separate data. Linear: Uses a straight line for separation. Polynomial: Uses polynomial curves for separation. RBF: Uses non-linear boundaries based on distance from a central point.

5.2 Recommender System

The project is going on as per the timeline. We have mostly completed the first half of our project and the song recommender system is left to be done. Our planned approach involves the following technique: Content-Based Filtering: Using this approach, we'll identify songs that are most similar to a particular song based on content features. When a user likes a certain song, we can recommend other songs that are nearest to it in terms of content.

6 Results and analysis

The models tested are shown in the below table

Before Hyper-parameter tuning:

	Accy	Precision	Recall	F1
LR	0.8417	0.8591	0.8417	0.8401
SGD	0.8273	0.8325	0.8273	0.8251
GNB	0.7985	0.8144	0.7985	0.7978
DTs	0.7553	0.7613	0.7553	0.7540
RFC	0.8417	0.8439	0.8417	0.8410
XGB	0.8561	0.8580	0.8561	0.8559
SVM Linear	0.8417	0.8527	0.8417	0.8407
SVM Poly	0.8417	0.8652	0.8417	0.8404
SVM RBF	0.8201	0.8304	0.8201	0.8162

We did Hyperparameter which enhances model performance, reduces overfitting, and ensures efficient resource usage. Well-tuned hyperparameters lead to adaptable models that perform effectively across various datasets, making them more robust and versatile.

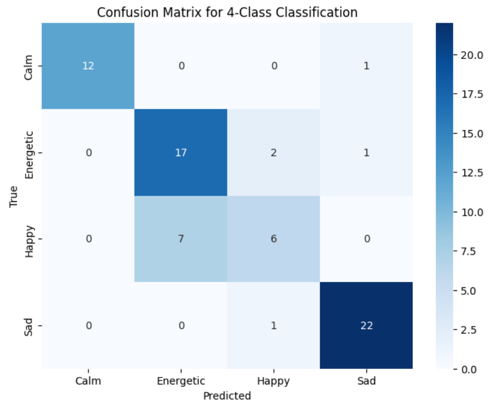
After Hyper-parameter Tuning:

	Accy	Precision	Recall	F1
SGD	0.8269	0.8010	0.7986	0.7969
DTs	0.7769	0.7820	0.7769	0.7755
RFC	0.8533	0.8553	0.8533	0.8528
SVM Linear	0.8545	0.8693	0.8345	0.8321
SVM RBF	0.8273	0.8418	0.8373	0.8248

After hyperparameter tuning with cross validation, we notice an improvement in the evaluation metrics of Random Forest (by optimizing the number of trees and their depth) , SVM (Linear and RBF by optimising the regularisation parameter and kernel-specific parameters), Decision Tree (by optimising the tree's depth and better split criterion) and SGD Classifier. There was no improvement in the evaluation metrics of XGB model as it is already optimised.

Setting a Baseline
Baseline models selected for evaluation are **Gausian Naive Bayes** and **Decision Tree** having accuracies 79.85% and 77.69% respectively.

Best Models
We get the best results from **XGBoost Classifier** and **Random Forests** with accuracies 85.61% and 85.33% respectively.



This is because ensemble models like Random Forest and XGB have higher accuracy and F1 scores compared to simpler models, given their ability to capture complex relationships and reduce overfitting.

7 Conclusion

7.1 Outcomes

We have discovered that the features we've used are highly effective in discerning the mood of a song, and as a result, they can effectively represent the mood of the listener. The XGB Classifier model has yielded a promising metric score, which gives us the confidence to move forward and develop a recommender system for songs. This system will be capable of suggesting songs to users based on the mood of the song they are currently listening to.

7.2 Future Work

We are yet to test more complex models and perform further hyper-parameter tuning on our current models to improve the quality of the overall system.
Further we plan to implement a recommender system which uses the moods classified on the basis of the classifier model to recommend songs based on identified mood.

7.3 Member Contribution

Harshil Mital:
Performing extensive Exploratory Data Analysis (EDA) with compelling visualizations, revealing key insights and patterns in the data. Also, experimented with various ML models and conducting performance evaluations, leading to the selection of the most suitable algorithms for the project's objectives.
Harsh Popat:
Crafting advance feature engineering techniques, transforming raw data into informative features. Employed domain knowledge and creative methods to generate relevant attributes, enhancing model predictive power by feature selection and hyperparameter used to improve model performance and helps uncover valuable insights from the data.
Harshvardhan Singh:
Did comprehensive literature review on related research to inform data selection and model design, ensuring the project's foundation is built on sound empirical and theoretical knowledge. Explored Various ML models and decided on which models to be used for the project.
Ashwin R Nair:
Setting Up and Designing a robust preprocessing pipeline, ensuring data cleaning, feature scaling, and encoding for optimal model performance. Implemented techniques to handle missing values and outliers, enhancing data quality.