Mood–Based Classifier and Recommender System

Final Project Presentation

Group-41

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI

Motivation Video

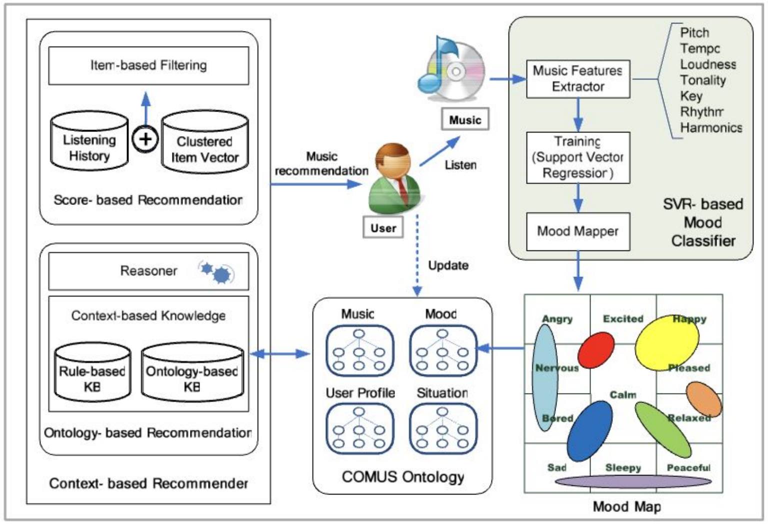
Motivation

Music deeply resonates with our emotions, yet choosing a song to match one's mood in today's vast digital library is daunting. Most platforms only suggest based on user history, neglecting the emotional aspect. Our project offers a **Mood-Based Song Classifier and Recommender System**, using machine learning to discern moods from recent song choices. We aim for a personalized listening experience, bridging the gap between emotions and song recommendations. Our goal is to align every song suggestion with the user's emotional state.

Literature Review

- **SVR-based music mood classification and context-based music recommendation by Seungmin Rho , Byeong-jun Han and Eenjun Hwang.**

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**.



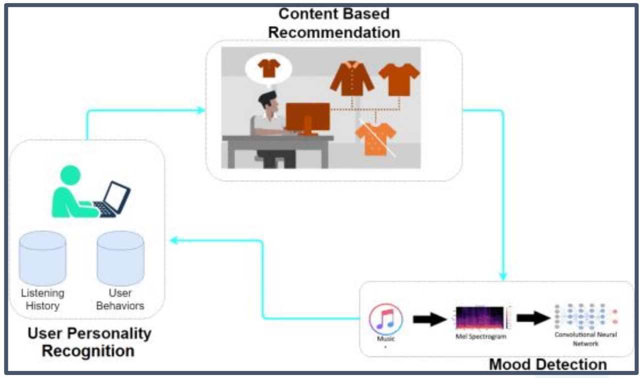
Literature Review

- **Moodplay: Interactive Mood-based Music Discovery and Recommendation by Ivana Andjelkovic, Denis Parra, John O'Donovan.**

  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**.

- **An Emotional Recommender System for Music by Vincenzo Moscato, Antonio Picariello and Giancarlo Sperli.**

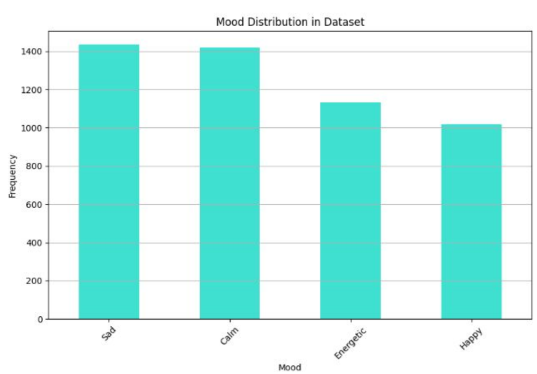
  In this paper, the authors describe a novel music recommendation technique based on the **identification of personality traits, moods, and emotions of a single user**. It embeds users' personality and mood with a **content -based filtering** approach to obtain accurate and dynamic results.



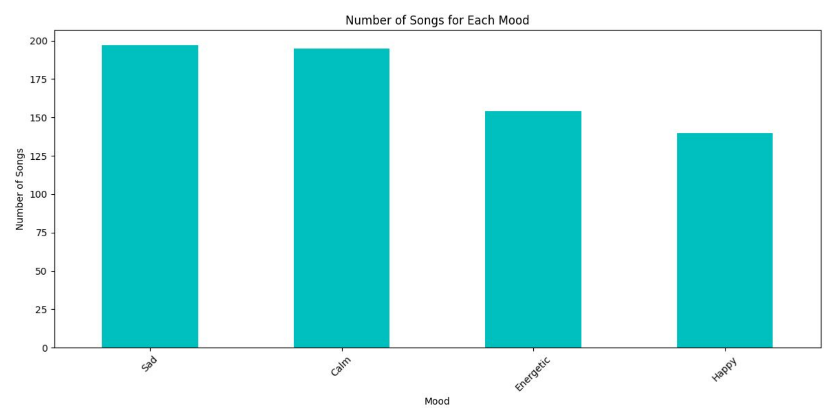
Dataset Description

We used the Spotify Music data [LINK] which contains songs from different artists and multiple genre. All the songs features were extracted using Spotify API.

We have populated the data for the recommender system to 2501 songs.



New data



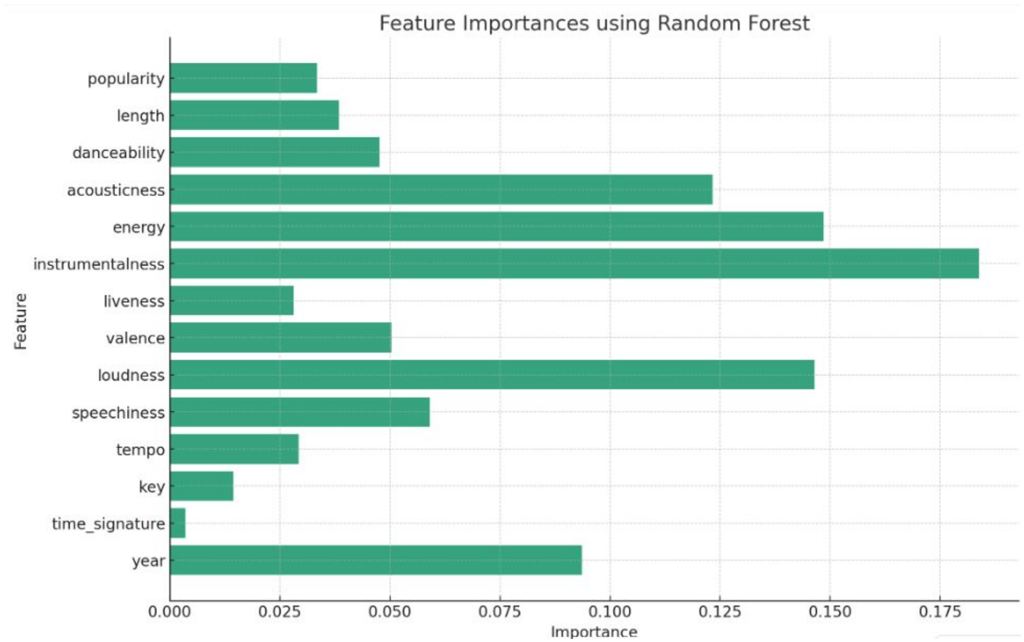
Old Data

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

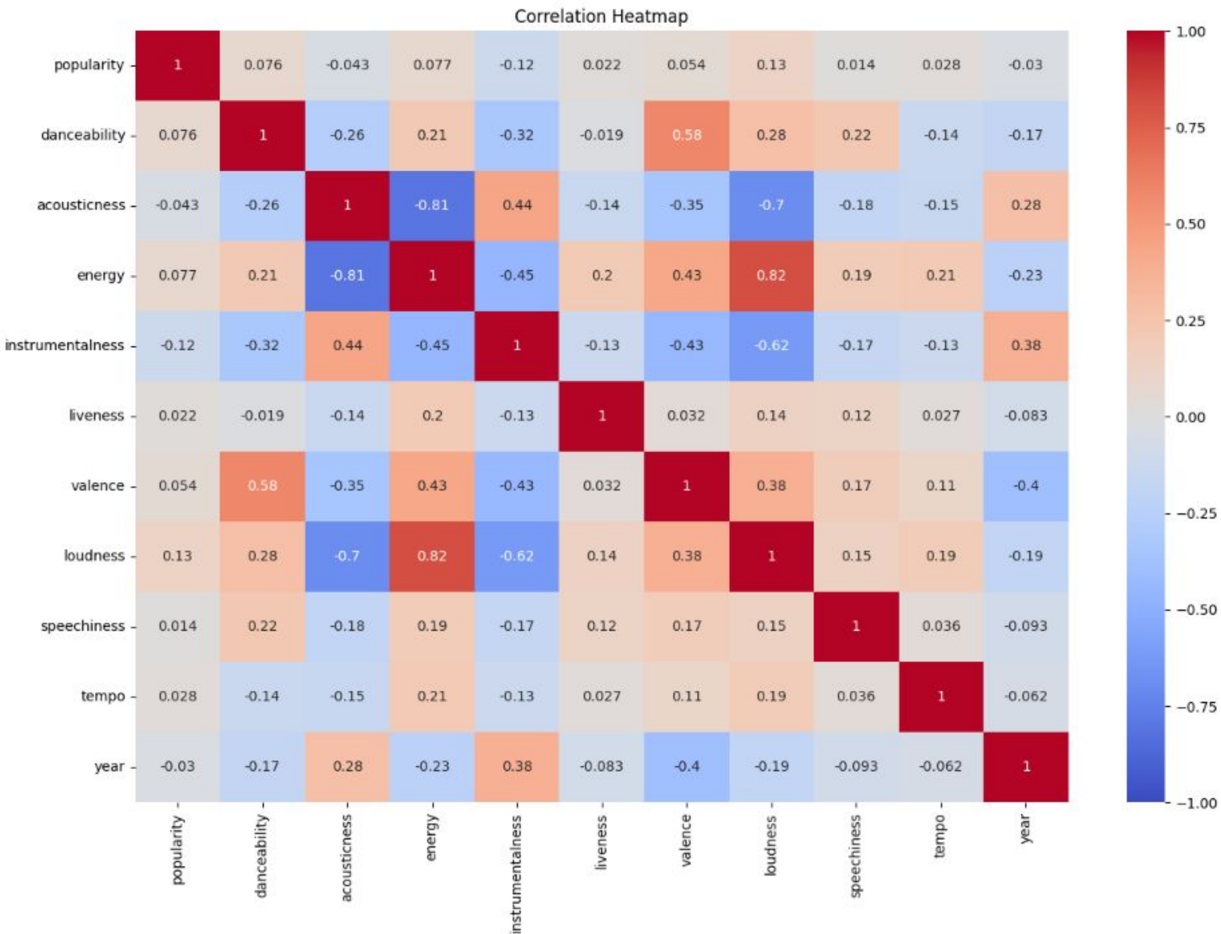
Raw Features

DataSet Preprocessing

- Categorical Features such as name, artist, and id are dropped.
- Numerical Features are standardised to ensure they have a mean of 0 and a standard deviation of 1.
- Label encoding of categorical target label mood.
- Feature Selection has been been done using Random Forest and based on the avg. impurity loss time signature and key are dropped.



Dataset Visualisation

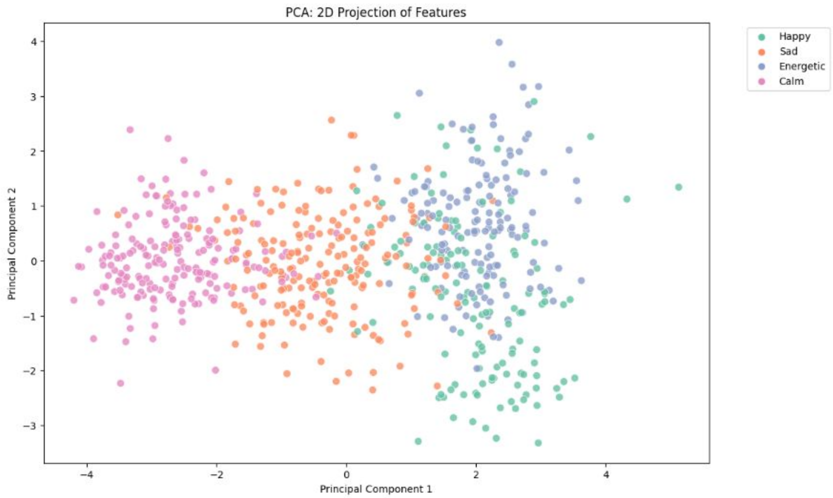


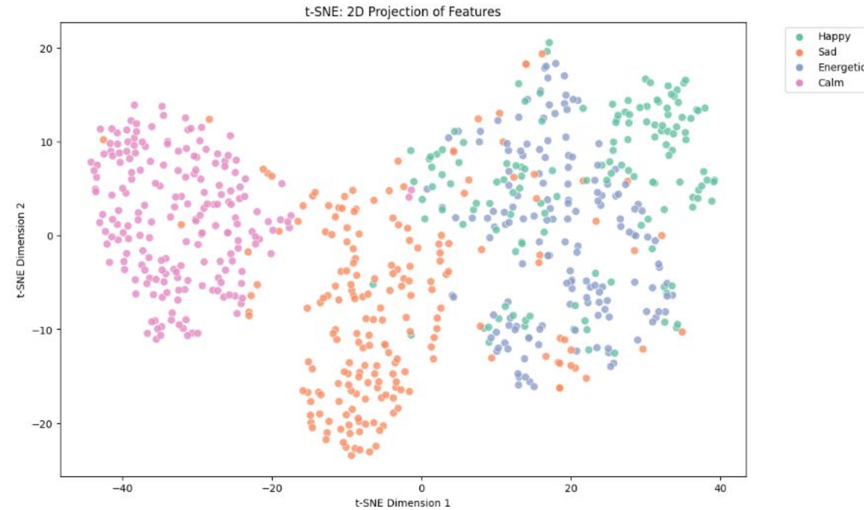
Positive Correlation:
Energy and Loudness

Negative Correlation:
Acousticness and Energy

Data Visualization

Dimensionality Reduction





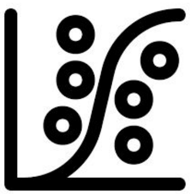
Principal Component Analysis

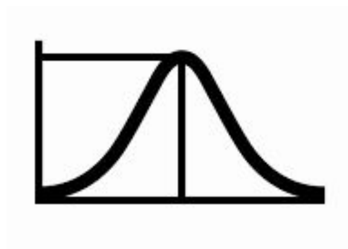
T-distributed Stochastic Neighbor Embedding

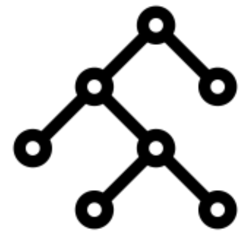
Moods are clearly separable (in separate clusters) except for energetic and happy songs, where there is a some amount of overlap.

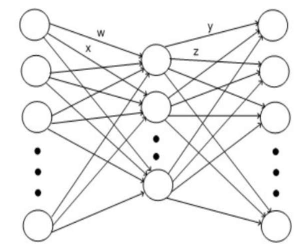
Methodology

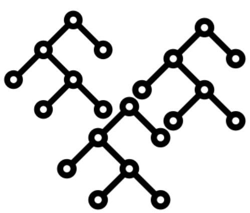
Classifier Models Tested

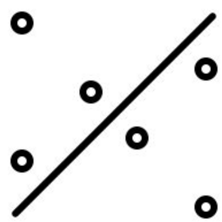

Logistic Regression


Gaussian Naive Bayes

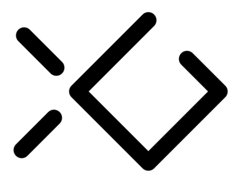

Decision Tree

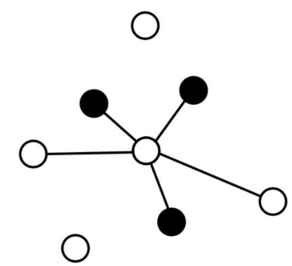

ANN


Random Forest


Support Vector Machine

- Linear
- Polynomial
- RBF


XGBoost Classifier


KNN

Recommender System

- First input user’s listening history during a session in list format and specific number of song recommendations needed.
- Then our system computes the average values across 11 musical features like the valance, danceability, from the user’s session playlist. This average is then used as input for the XGBoost classifier to infer the user’s current mood.
- The system computes the average values across 11 musical features like the valance,danceability, etc. from the user’s session playlist. This average is then used as input for the XGBoost classifier to infer the user’s current mood.
- With the mood determined , we now start searching the songs for the user.
- To achieve this, we employed a **Euclidean search technique**, identifying songs that closely match the computed average feature values and ensuring that finding is done in such a way around that the mood of these songs aligns with the inferred mood of the user in Classifier.

Recommender System – Working

```
results = Music_Recommender([
    {'name': "All-Jacked-Up", 'year': 2005},
    {'name': 'Lighters-Up', 'year': 2005},
    {'name': 'Take-the-Time', 'year': 1992}
], data_moods)
```

3 Songs are input to the Recommender System

Recommender System – Working

```
results = Music_Recommender([
    {'name': "All-Jacked-Up", 'year': 2005},
    {'name': 'Lighters-Up', 'year': 2005},
    {'name': 'Take-the-Time', 'year': 1992}
], data_moods)
```

3 Songs are input to the Recommender System

Average is calculated for all the 11 features and then determined the mood of the user by it.

[[0.25178567 -0.29098346 -0.93822972 1.08701203 -0.84986594 -0.33774234
  1.08108242 0.66089143 3.89387694 0.57020192 -0.9529511]]

Model Classifies the mood as Energetic

Recommender System – Working

```
results = Music_Recommender([
    {'name': "All-Jacked-Up", 'year': 2005},
    {'name': 'Lighters-Up', 'year': 2005},
    {'name': 'Take-the-Time', 'year': 1992}
], data_moods)
```

3 Songs are input to the Recommender System

Average is calculated for all the 11 features and then determined the mood of the user by it.

Model Classifies the mood as Energetic

	year	name	artist	mood
1294	1998	Johnny Jump Up/Morrison's Jig	Gaelic Storm	Energetic
696	1992	Session	The Offspring	Energetic
1835	2003	Lying from You	Linkin Park	Energetic
424	2000	Points of Authority	Linkin Park	Energetic
1942	2004	Let It Bleed	The Used	Energetic
1803	2003	The Crowing	Coheed and Cambria	Energetic
1596	2001	Dead Cell	Papa Roach	Energetic
1903	2004	Still Running	Chevelle	Energetic
581	2008	Toxic	A Static Lullaby	Energetic
104	2005	Come Out And Play (Keep 'Em Separated)	The Offspring	Energetic

Recommendations based on Euclidean Distance

Results and Analysis

- We selected **Gaussian Naive Bayes** and **Decision Tree** as the baseline models because of their minimal complexity.
- We tested the models listed below and obtained the metrics before and after hyperparameter tuning with **cross-validation (k=10)**.
- Decision Trees, in particular, showed noticeable improvement after hyperparameter tuning, emphasizing the value of tuning especially for models with numerous parameters like trees.
- SVM (both Linear and RBF kernel) also improved, suggesting that the default hyperparameters were not optimal for this dataset.

Before Hyper-parameter tuning:

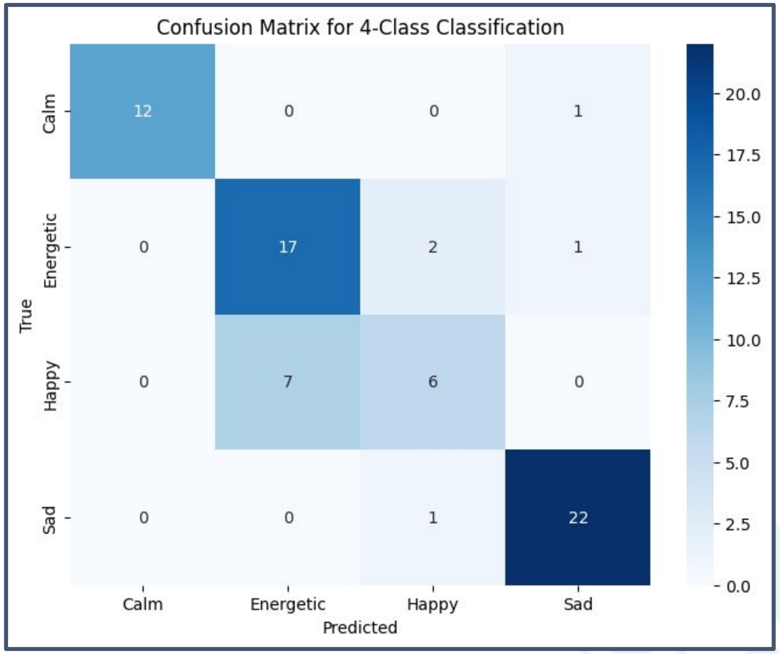
	Accy	Precision	Recall	F1
LR	0.8489	0.8622	0.8489	0.8493
SGD	0.7410	0.7549	0.7410	0.7354
GNB	0.8129	0.8285	0.8129	0.8122
DTs	0.7625	0.7636	0.7625	0.7608
RFC	0.8489	0.8558	0.8489	0.8493
XGB	0.8561	0.8562	0.8561	0.8550
SVM Linear	0.8417	0.8546	0.8417	0.8419
SVM Poly	0.8345	0.8460	0.8345	0.8335
SVM RBF	0.8417	0.8474	0.8417	0.8396
KNN	0.8213	0.8252	0.8273	0.8247
ANN	0.8273	0.8176	0.8243	0.8245

After Hyper-parameter Tuning:

	Accy	Precision	Recall
SGD	0.8269	0.8010	0.7986
DTs	0.7769	0.7820	0.7769
RFC	0.8533	0.8553	0.8533
XGB	0.8633	0.8652	0.8633
SVM Linear	0.8545	0.8693	0.8345
SVM RBF	0.8273	0.8418	0.8373

Results and Analysis

- After hyper parameter tuning with cross-validation, we notice an improvement in the evaluation metrics of Random Forest , SVM, Decision Tree and SGD Classifier. There was no improvement in the evaluation metrics of the XGB model as it is already optimised.
- We get the best results from **XGBoost Classifier and Random Forests** with accuracies of **86.31%** 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.



Conclusion

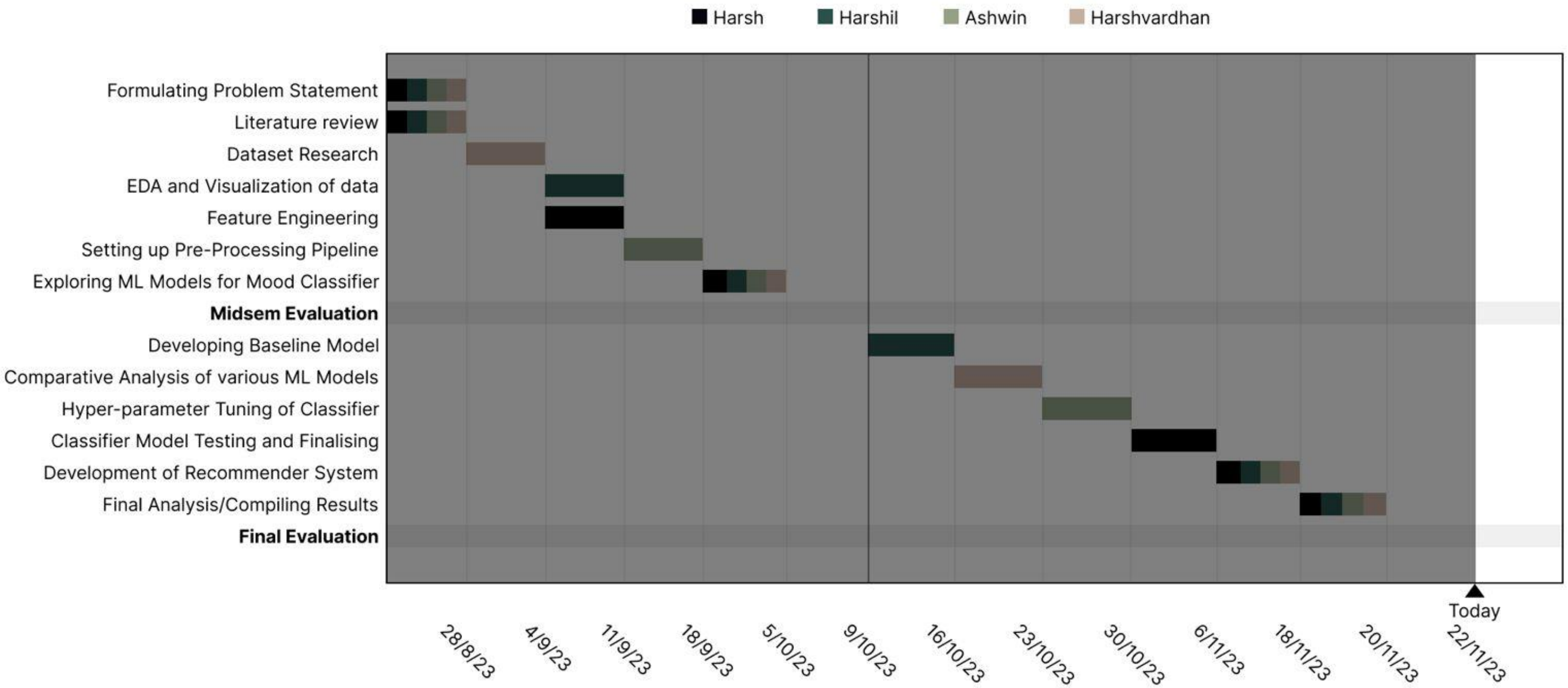
We have discovered that the features we've used are highly effective in discerning the mood of a song, and as a result, they can effectively represent the mood of the listener. The XGB Classifier model has yielded a promising metric score.
We have implemented a recommender which based on the given input, classifies the mood and analyze users' music preferences and offers personalized song recommendations that are both sonically compatible and emotionally resonant.

Recommender System Demo

Link:
https://www.kaggle.com/harshparimalpopat/final-ml-project

Timeline

Completed ALL the work according to the timeline



Individual Contribution

Ashwin R Nair

Setting Up and Designing a robust pre-processing pipeline, ensuring data cleaning,feature scaling, and encoding for optimal model performance. Implemented techniques to handle missing values and outliers, enhancing data quality.

Harsh Popat

Enhancing model predictive power by feature selection and hyperparameter used to improve model performance and help to get valuable insights from the data.

Harshil Mital

Performing Exploratory Data Analysis (EDA). Also, experimented with various ML models and conducting performance evaluations, leading to the selection of the most suitable algorithms.

Harshvardhan Singh

Did 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 to get familiar with the available techniques.

THANK YOU!

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

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
DELHI