**MAJOR PROJECT**

**ON**

**SPOTIFY MUSIC TRACK ANALYSIS AND AUDIO FEATURE-BASED RECOMMENDATION SYSTEM**

*Submitted to*

*Amity University Kolkata*



*For the partial fulfillment of the award of the degree*

BACHELOR OF SCIENCE DATA SCIENCE(HONOURS)

by

ROHIT AGARWAL

A914138122009

Under the guidance of

Dr Indraneel Mukhopadhyay

AMITY INSTITUTE OF INFORMATION TECHNOLOGY

AMITY UNIVERSITY KOLKATA

April 2025

**DECLARATION**

I hereby declare that the dissertation entitled "Spotify Music Track Analysis and Audio Feature-Based Recommendation System" submitted by me in partial fulfillment of the requirements for the award of the Degree of B.Sc. [Hons.] Data Science to the AMITY UNIVERSITY, KOLKATA is based on the experiments and studies carried out by me. This work is original and has not been submitted in part or full for any other degree or diploma of any university or institution.

Date: April 29 , 2025

Place: KOLKATA

ROHIT AGARWAL

A914138122009

**CERTIFICATE**

The research work embodied in this dissertation entitled "Spotify Music Track Analysis and Audio Feature-Based Recommendation System" submitted by Rohit Agarwal, A914138122009 in partial fulfillment of the requirements for the award of the Degree of B.Sc. [Hons.] Data Science is based on the experiments and studies carried out by him/her. This work is original and has not been submitted in part or full for any other degree or diploma of any university or institution.

Date: April 29 ,2025

Place:KOLKATA

DR. INDRANEEL MUKHOPADHYAY   
 (SIGNATURE)

**ACKNOWLEDGEMENT**

I would like to express my sincere gratitude to **Dr. Indraneel Mukhopadhyay** for his invaluable guidance, support, and mentorship throughout this research. His expertise and insights have been instrumental in shaping this study.

I also extend my appreciation to all those who contributed directly or indirectly to the successful completion of this project, including colleagues, friends, and family who provided encouragement and assistance along the way.

ROHIT AGARWAL  
 (SIGNATURE) A914138122009



**FEEDBACK BY EXAMINERS**

**A. Comments from Seminar Guide:**

**B. Comments from External Examiner:**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| Serial no. | Topic | Page no. |
| 1 | Introduction | 1-3 |
| 2 | Literature Review | 3-5 |
| 3 | Methodology | 5-27 |
| 4 | System Design and Architecture | 27-30 |
| 5 | **I**mplementation | 31-34 |
| 6 | Conclusion | 35 |
| 7 | Future Work | 36-37 |
| 8 | References | 37 |

**List of Tables**

|  |  |  |
| --- | --- | --- |
| Serial no. | Topic | Page no. |
| 3.1 | A Table of key features in the dataset | 6-7 |
| 3.2 | Model performance table | 21 |
| 5.1 | Tools and libraries used | 34 |

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Topic** | **Page no.** |
| 3.1 | Distribution Analysis of Audio Features | 10 |
| 3.2 | Average track popularity over time | 11 |
| 3.3 | Popularity trends by playlist genre and popularity type over time | 11 |
| 3.4 | Most and least popular genre over time | 12 |
| 3.5 | Most popular genre by year | 12 |
| 3.6 | Number of tracks released over years | 12 |
| 3.7 | Genre Distribution by Popularity Type | 13 |
| 3.8 | Pair Plot of Selected Features by Popularity Type | 14 |
| 3.9 | Danceability vs Energy | 15 |
| 3.10 | Liveness vs Acousticness | 15 |
| 3.11 | Track Duration by Popularity Type | 16 |
| 3.12 | Visualization of Performance Matrix for Different Machine Learning Model | 22 |
| 3.13 | Feature importance for all the models | 23 |

**Spotify Music Track Analysis and Audio Feature-Based Recommendation System**

**Abstract**

This study identifies machine learning approaches to establish relationships between musical attributes and track popularity while making predictions about musical success. We processed data from songs through a dataset with attributes for danceability, energy, tempo and genre before we evaluated multiple classification models including Decision Tree, Random Forest, Support Vector Machine (SVM), and Logistic Regression to understand which songs were "High" popularity or "Low" popularity.

Feature engineering techniques produced meaningful insights from the data through advanced interaction term formulas and by using song age as a time-based element. The evaluation of the models demonstrated Random Forest as one of the superior performers when measured against F1-score, Accuracy, Recall and ROC AUC.

The system utilized content-based recommendations that combined song features to calculate similarity using cosine measures for assessing track recommendations. The system provides an actual machine learning implementation to music recommendation services which offers enhanced user experience through personalized song recommendations.

The project demonstrates complete data science workflow from preprocessing to modeling to evaluation and recommendation which constitutes a data-driven solution for studying popular music and intelligent music recommendation systems.

**1. Introduction**

Digital media has made music highly available to users through an unprecedented range of musical content. Spotify and Apple Music and Amazon Music provide users with immediate access to numerous million songs across infinite genres and artists. Rising digital music consumption has made it essential for experts to deeply understand all the variables which determine the success of songs in the market. The industry uses this ability to determine track popularity alongside content recommendation technologies through record labels who need promotion planning and artists who want audience reach assessment and platform operators who need user engagement optimization.

Internet music streaming platforms monitor user interactions with each song and break down recorded information into substantial quantitative measurements. Spotify uses tempo active metadata such as danceability valence and acoustic ness in addition to style information about genre artists and albums to offer a complete musical features suite. The data collection encompasses potential applications of machine learning methods to discover patterns and generate predictions that develop intelligent systems which operate similarly to human preference systems.

The research objective targets the connection between unprocessed music data and practical information that produces value. The main objectives of the project involve classifying songs by popularity using audio characteristics and determining the elements that lead to music success while developing recommendations based on content attributes. The project demonstrates how AI can transform music industry user experiences by using data science and musical art to reveal capabilities of artificial intelligence.

**1.1 Project Motivation**

Artificial intelligence together with machine learning technology provides music industry professionals with the best opportunity to discover valuable insights from complex datasets. Data-driven decisions now revolutionize the music industry because its fast-shifting trends make success predictions extremely difficult.

Numerous determining elements drive this project forward:

1. **Artist Empowerment**: The comprehension of elements which impact popularity brings advantages to independent artists and producers when it comes to artist empowerment. Artist understanding of musical attributes affects their ability to adapt their work toward increasing their audience size and engaging users.
2. **User Experience**: The streaming service aims to hold users on the platform longer by presenting music recommendations which match their individual listening taste. When designed to handle song features a recommendation system proves essential for this application.
3. **Market Insights**: The strategic decision-making process of song promotion becomes improved through audio feature analysis that enables record labels and marketers to identify potential hits during early stages of product development.
4. **Technological Exploration**: Data scientists face a complex project featuring classification together with feature engineering and system design aspects in this technological exploration.

The project combines creative association with computing operations to analyze undiscovered music popularity trends.

**1.2 Problem Statement**

Musical preferences combine high subjectivity with unprecedented changes which create distinctive difficulties for the music industry. Large-scale access to audio data and user interaction information does not make model prediction an easy process for platforms. The research focuses on addressing the following connected difficulties:

1. **Classification**: We need to determine how precisely we can identify "High Popularity" and "Low Popularity" tracks through audio features and metadata alone without depending on user preferences or track plays.
2. Feature Analysis: The evaluation of musical features reveals what aspects affect a song's fame levels most significantly. Genres show different tendencies to exhibit particular audio patterns during classification.
3. **Recommendation System**: An accurate content-based recommendation technology exists to recommend audio-similar songs for a particular track through its underlying audio properties.

Through solving these research questions, the project advances the development of better music discovery systems which benefit both music artists and music platform operations through data-based techniques.

The implementation of machine learning projects completely relies on data as its fundamental structure. The data quality together with its relevance and variety determine how reliable and successful the derived insights will be. The project dedicated thorough efforts to data collection and preprocessing for analyzing Spotify tracks to understand and predict song popularity. The data preprocessing strategy focused on creating a structurally sound database for effective model classification and system recommendation purposes.

**2. Literature Review**

Recent years brought substantial progress in the integration between machine learning methods and musical analysis for recommendation creation especially through the development of platforms where Spotify stands out. Music industry practitioners together with researchers have directed substantial efforts to grasp populational drivers behind musical success while building improved recommender systems.

1. Music Popularity Prediction: Research has demonstrated that musical audio characteristics hold the most significant impact on music track success systems. Kim et al. (2020) proved through their research that music popularity depends heavily on the song attributes of energy and danceability together with valence scores. Random Forests performed better than Logistic Regression as an ensemble model proved superior to simpler models according to their research using machine learning methods. There is research evidence supporting how model performance improves substantially when scientists integrate feature engineering which creates interaction terms uniting tempo and energy factors (Zhang et al., 2021).
2. Feature Engineering and Audio Characteristics: The analysis of music in past times depended primarily on artist names alongside genre and year of release metadata. The current machine learning pipelines use APIs from the Spotify Web API to extract intrinsic audio features for their analyses. The Spotify Web API delivers track composition details through features which combine tempo with loudness along with acousticness and instrumentalness (Tzanetakis & Cook, 2002). Custom-made features developed through song-age calculation and binary flag generation (such as pre-2000s release) have demonstrated their importance to produce more generalized and easier-to-understand models.
3. Content-Based Recommendation Systems: Content-based filtering developed as the initial recommendation system technique and continues to play an essential role with restricted user interaction information. The item feature-based approach of content-based methods operates independent from user behavior patterns because collaborative filtering systems need. Music recommendation systems within this approach provide results through comparisons of audio parameters between multiple musical tracks (Lops et al., 2011). The Cosine similarity method stands out as the dominant measure to understand vector closeness in songs because it handles numerical feature spaces effectively (Salton & McGill, 1983).
4. Challenges in Music Recommendation: The implementation of content-based systems has proven successful yet faces two major problems which include restricted user content discovery and the recommendation of similar items to already known content. The combination of content-based approaches with collaborative filtering in hybrid systems represents a solution strategy to overcome these system weaknesses (Burke, 2002). The research field now investigates using song lyrics and user-added tags together with audio elements to enhance music item representation (Oramas et al., 2017).
5. Future Trends
6. Deep learning models are becoming more prominent in the field by using convolutional neural networks (CNNs) on spectrograms alongside recurrent neural networks (RNNs) for predicting music listening behavior sequences. Explainable AI (XAI) gains rising importance to create transparent recommendation systems which demonstrate their recommendations to end users and stakeholders through explanations.
7. Scientific studies demonstrate machines can determine music popularity with significant accuracy by using properly designed audio features. Music discovery through content-based recommendations provides users with excellent track discovery capabilities using musical elements but advanced deep learning techniques and hybrid models will enhance user satisfaction levels.

**3. Methodology**

**3.1 Data Sources**

The project dataset combines important audio feature and song metadata and popularity score information which stems from various trusted reliable sources. The primary data sources include:

1. **Spotify Web API**: Through the Spotify Web API developers can retrieve live data combined with historical records from tracks albums and artists. Through the API users access eight audio features which include tempo, energy, danceability, valence, loudness alongside acousticness and instrumentalness and measurement of the liveness aspect alongside speechiness level.
2. **Curated Spotify Playlists**: The collection of songs for analysis included curated playlists from Spotify across multiple genres including “Rock Classics” and “Jazz Classics” and “Pop Remix” and “Reggae Classics” and others. The playlists serve to create a dataset structure that includes diverse musical styles suitable for different preference groups.
3. **Kaggle Datasets:** Kaggle Datasets were used to incorporate additional datasets containing track release date information and subgenres along with popularity ratings. The metadata collection through merged information added more detail to the database while making some fields more complete.

The combination of different musical playlist categories offered a complete perspective of songs while simultaneously enabling strong and effective analysis and modeling capabilities.

**3.2 Key Features in the Dataset**

Here’s a table summarizing the key features of the dataset:

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature Name** | **Description** | **Data Type** | **Example Value** |
| track\_name | Name of the track | String | "Blinding Lights" |
| artist\_name | Name of the artist | String | "The Weeknd" |
| album\_name | Name of the album | String | "After Hours" |
| track\_album\_release\_date | Release date of the track (converted to datetime) | DateTime | "2020-03-20" |
| release\_year | Extracted year of track release | Integer | 2020 |
| song\_age | Age of the song (Current Year - Release Year) | Integer | 5 |
| playlist\_genre | Genre of the playlist the song is featured in | String | "pop" |
| Popularity\_Type | Categorization of track popularity (High/Low) | String | "High" |
| track\_popularity | Popularity score (0-100) based on Spotify’s ranking | Integer | 85 |
| duration\_ms | Duration of the track in milliseconds | Integer | 210000 |
| duration\_min | Converted duration in minutes | Float | 3.5 |
| danceability | How suitable a track is for dancing (0-1 scale) | Float | 0.78 |
| energy | Intensity and activity level of a song (0-1 scale) | Float | 0.85 |
| loudness | Overall loudness in decibels (scaled) | Float | -5.3 |
| **Feature Name** | **Description** | **Data Type** | **Example Value** |
| speechiness | Presence of spoken words in a track (0-1 scale) | Float | 0.08 |
| acousticness | Probability of the track being acoustic (0-1 scale) | Float | 0.12 |
| instrumentalness | Likelihood of a track being instrumental (0-1 scale) | Float | 0.00 |
| liveness | Probability of a live performance (0-1 scale) | Float | 0.13 |
| valence | Positivity or musical happiness of a track (0-1 scale) | Float | 0.72 |
| tempo | Speed of the song in beats per minute (BPM) | Float | 120.5 |
| is\_2000s\_or\_earlier | Binary indicator if the song was released in 2000 or earlier | Integer (0/1) | 0 |
| energy\_danceability\_ratio | Interaction feature: energy divided by danceability | Float | 1.09 |
| valence\_energy\_product | Interaction feature: product of valence and energy | Float | 0.612 |

Table 3.1. A Table of key features in the dataset

This table provides a structured summary of the dataset's attributes, including numerical, categorical, and derived features. Let me know if you'd like to add more details.

**3.3 Data Preprocessing**

The raw data in process regularly includes multiple errors such as missing information and formatting problems that restrict machine learning model performance. The data received preprocessing steps that consisted of:

1. **Handling Missing Values**: The data preprocessing included track deletion when essential information such as energy or valence or popularity was absent.
2. **Removing Duplicates**: A check for duplicate records and tracks allowed the removal of all additional occurrences to ensure each row contained distinct information.
3. **Data Type Conversions**: The track\_album\_release\_date field received transformations which changed the original string data into datetime type. The conversion allowed the extraction of release year data along with temporal measurements.
4. **Target Encoding**: The track\_popularity numeric values received target encoding treatment to categorize them into two groups that served binary classification modeling needs.
5. **Column Renaming and Cleaning:** The analysis dataset obtained from the initial data through two operations: columns received standard naming conventions and unneeded columns got removed as part of the cleaning process.

The processing steps led to a clean dataset which protected the complete and intact nature of all data points.

**3.4 Feature Scaling**

Distance-based and gradient-based machine learning algorithms including SVM, logistic regression, and neural networks need feature values to have similar measurement units for proper operation. In this project:

1. The Standard Scaling procedure was implemented through StandardScaler from scikit-learn. Standard scaling changes the numerical features by shifting their mean to zero while making their standard deviation equate to one.
2. The standardized features comprise danceability, energy, valence, tempo, loudness, acousticness, instrumentalness, liveness, speechiness and the engineered song\_age.
3. Model performance would show bias if this scaling step is omitted because numerical features with extensive range values such as loudness or tempo would enforce their influence in models.

The strict preparation method made the dataset suitable for developing dependable classification models which would later support a content-based recommendation platform.

**3.5 Exploratory Data Analysis (EDA)**

Exploratory Data Analysis (EDA) serves as a critical phase in any data-driven project. EDA delivers essential first-time information regarding the dataset's organizational elements and patterns along with data spread and connections. EDA was utilized to investigate music track data characteristics between genres and popularity types and audio features. The analysis aimed to identify patterns together with anomalies and relationships which would help in designing features for models.

**3.5.1 Overview of the Dataset**

Multiple audio features included in the dataset encompass danceability, energy, loudness, and speechiness in addition to acousticness, valence, tempo, and several others. The audio tracks have built-in metadata that includes genre specifics with sub-genre names besides artist and album information and track duration and popularity ratings. The Popularity\_Type target variable classifies songs according to their track popularity scores as High or Low popularity status.

A preview of the data structure was conducted through combined\_df.head() to check for correct data loading and examine the first few rows.

**3.5.2 Distribution Analysis of Audio Features**

To assess how audio features vary with popularity, **boxplots** were generated for features like:

A graph with a line graph and a line graph

AI-generated content may be incorrect.A graph with green and blue lines

AI-generated content may be incorrect.A graph of a distribution of valence

AI-generated content may be incorrect.A graph with a green and blue line

AI-generated content may be incorrect.A graph with a line graph and a number of numbers

AI-generated content may be incorrect.A graph with a green line

AI-generated content may be incorrect.A graph with green and blue lines

AI-generated content may be incorrect.A graph with a number of different colored lines

AI-generated content may be incorrect.

Fig.3.1.Distribution Analysis of Audio Features

Each plot compared feature distributions across **High** and **Low** popularity tracks.

Key findings included:

* High popularity songs tend to have higher danceability and energy.
* Low popularity songs showed more acousticness and variability in tempo.

**3.5.3 Popularity Trends Over Time**

To understand how music popularity has evolved, line plots were created showing:

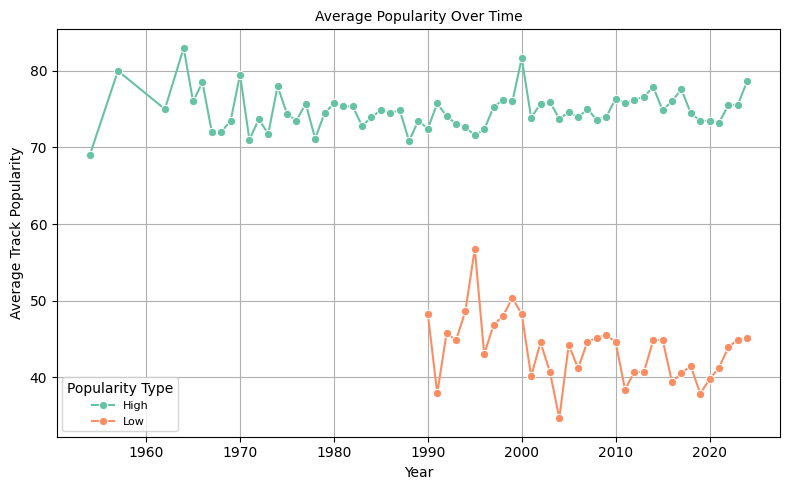


Fig.3.2. Average track popularity over time

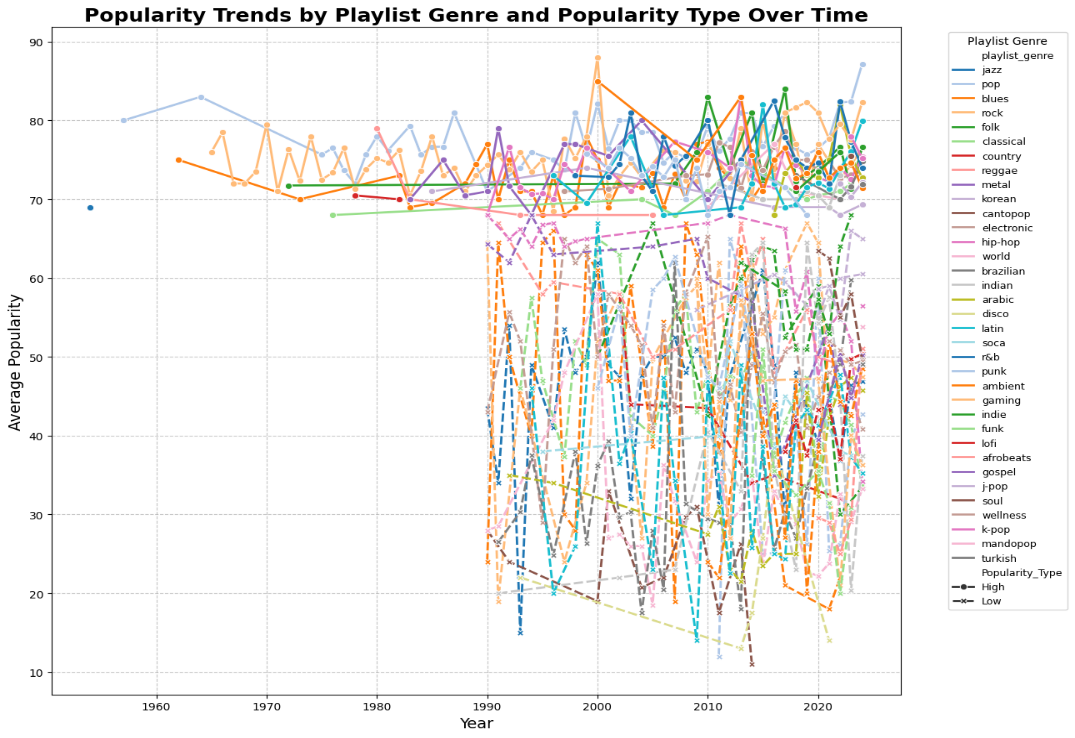


Fig.3.3. Popularity trends by playlist genre and popularity type over time

A graph with blue and orange lines

AI-generated content may be incorrect.

Fig.3.4. Most and least popular genre over timeA graph of colored lines

AI-generated content may be incorrect.

Fig.3.5. Most popular genre by year

A graph showing a number of tracks

AI-generated content may be incorrect. Fig.3.6. Number of tracks released over years

These visualizations revealed fluctuations in genre popularity over the years and helped understand the evolving music trends.

**3.5.4 Genre Distribution by Popularity Type**

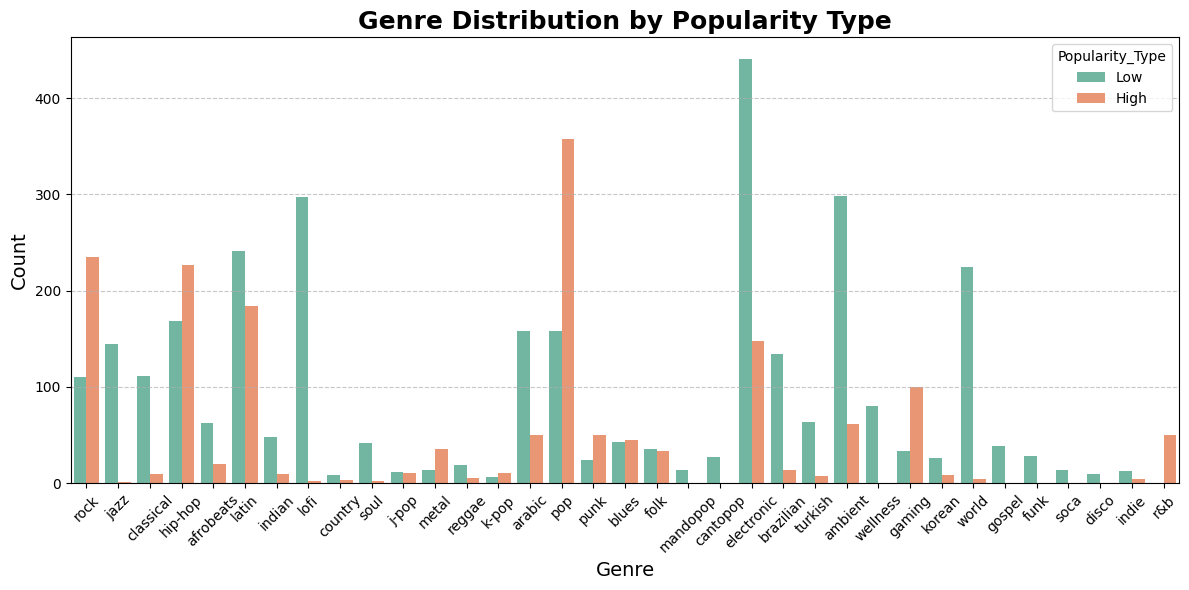
****

Fig.3.7. Genre Distribution by Popularity Type

A **countplot** was created to observe the genre distribution among high and low popularity tracks.

This revealed that genres like **pop** and **rock** had a higher proportion of popular songs, while niche genres like **jazz** and **classical** leaned towards lower popularity types.

**3.5.5 Feature Correlation and Relationships:**

To better understand feature interactions:

* **Scatter plots** for feature pairs (e.g., Danceability vs Energy, Liveness vs Acousticness) were plotted.
* A **pairplot** of selected numerical features colored by popularity type provided a comprehensive view of the relationships.

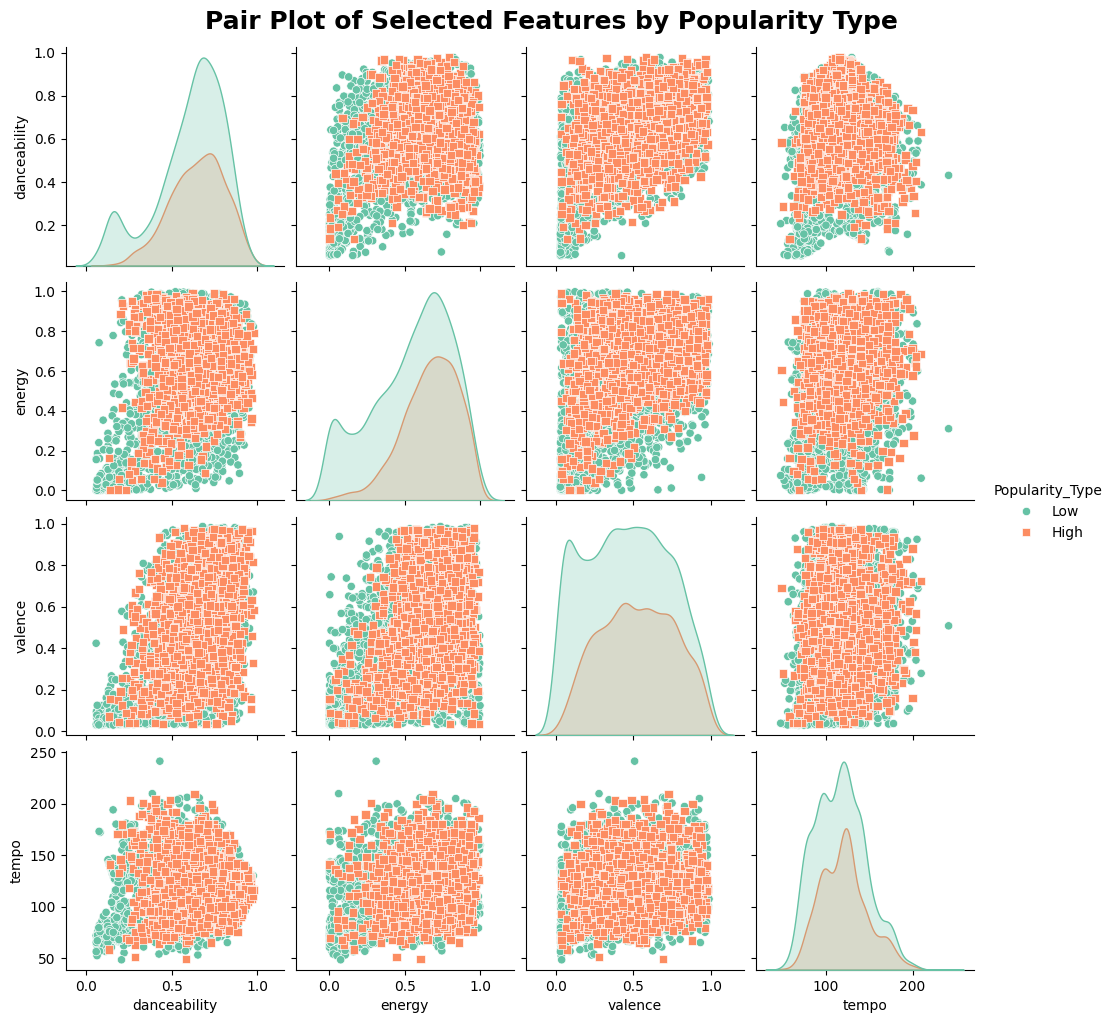


Fig.3.8*.* Pair Plot of Selected Features by Popularity Type

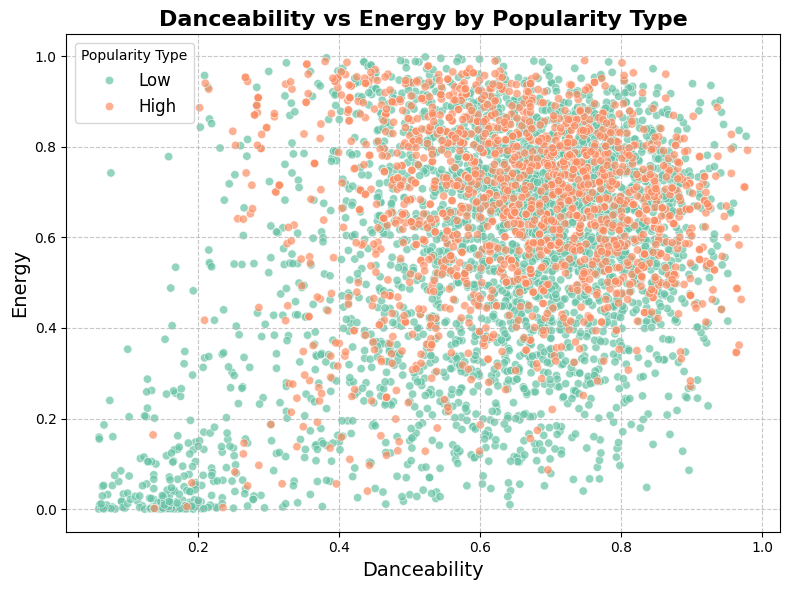


Fig.3.9. Danceability vs Energy

A graph showing different colored dots

AI-generated content may be incorrect.

Fig.3.10. Liveness vs Acousticness

The pairplot helped highlight clusters and separability between popular and non-popular tracks based on certain feature combinations.

**3.5.6 Track Duration Analysis**

To check if song duration influences popularity, the duration (converted from milliseconds to minutes) was analyzed using a **boxplot** by popularity type.

A graph of different colored lines

AI-generated content may be incorrect.

Fig.3.11. Track Duration by Popularity Type

It was observed that popular songs generally had a slightly shorter average duration.

**3.5.7 Summary of EDA Insights**

The EDA phase uncovered several important patterns:

* Certain genres are more prone to popularity than others.
* Audio features like **danceability, energy, and valence** are positively associated with popularity.
* There is a distinguishable separation between high and low popularity songs in terms of feature distribution.
* Older tracks (2000s or earlier) are more likely to be less popular in the modern streaming context.

These findings informed the next phase—**feature engineering and model selection**—by identifying features that significantly influence a song’s popularity.

**3.6 Feature Engineering**

Effective creation of robust machine learning models requires feature engineering as a fundamental step. The process transforms basic data into important attributes that enable more efficient learning of patterns by the model. The project used multiple techniques to improve the dataset while creating additional track popularity-related attributes.

**3.6.1 Song Age Calculation**

The song age function became a valuable addition to the system. The calculation involved deducting song release dates from 2025 to obtain song age. The concept suggests that music popularity patterns differ between old songs and new songs appearing in the market. Classic rock songs keep their appeal because of their iconic status while newer pop tracks gain fast popularity but maintain only a short life cycle.

* **Code:**

combined\_df['release\_year'] = combined\_df['track\_album\_release\_date'].dt.year current\_year = 2025 combined\_df['song\_age'] = current\_year - combined\_df['release\_year']

**3.6.2 Interaction Feature**

**To capture interactions between multiple audio characteristics, two new features were engineered:**

**Energy-to-Danceability Ratio: This ratio reflects how energetic a track is relative to its danceability. Tracks with high energy but low danceability may be intense but not suitable for dancing, which can influence popularity in genres like rock or metal.**

* **Code:** **combined\_df['energy\_danceability\_ratio'] = combined\_df['energy'] / (combined\_df['danceability'] + 1e-5)**

**Valence-Energy Product**: This product combines emotional positivity (valence) with energy. Tracks that are both high in energy and positive in tone are more likely to appeal to broader audiences, particularly in genres like pop or EDM.

* **Code:** **combined\_df['valence\_energy\_product'] = combined\_df['valence'] \* combined\_df['energy']**

**3.6.3 Categorical Transformation**

A new binary categorical feature named **is\_2000s\_or\_earlier** was added to capture the influence of older tracks:

* Code: combined\_df['is\_2000s\_or\_earlier'] = (combined\_df['release\_year'] <= 2000).astype(int)

This can be useful in understanding whether tracks from earlier decades show a different trend in popularity due to cultural relevance, nostalgia, or genre shifts over time.

**3.6.4 Handling Missing Values**

After creating new features, it was important to ensure data quality. All rows containing missing values were dropped to maintain a clean dataset for training machine learning models.

* **Code: combined\_df = combined\_df.dropna()**

**3.6.4 Feature Scaling**

Before feeding the data into machine learning algorithms, especially those sensitive to feature scales (e.g., SVM, Logistic Regression), all numerical features were standardized using **StandardScaler**. This ensures that each feature contributes equally to the model and prevents features with larger ranges from dominating.

* Code: from sklearn.preprocessing import StandardScaler

numerical\_features = ['danceability', 'energy', 'loudness', 'speechiness', 'acousticness', 'instrumentalness', 'liveness', 'valence', 'tempo', 'song\_age']

scaler = StandardScaler()

combined\_df[numerical\_features] = scaler.fit\_transform(combined\_df[numerical\_features])

**4. Model Building and Comparative Analysis**

A supervised machine learning system with different models was developed for proper classification of music track popularity between high and low categories. This phase sought both top classification results and identification of the optimal algorithm for this particular task. A standard evaluation of the models occurred by measuring F1 Score, Accuracy, Recall and ROC AUC metrics.

**3.7 Models Used**

**3.7.1 Logistic Regression**

The baseline model used Logistic Regression because it provides straightforward interpretation and simple implementation for the analysis. The linear logistic regression model delivered satisfactory results without losing its status as an excellent performance standard for advanced models.

**3.7.2 Support Vector Machine (SVM)**

The SVM model achieved high performance in high-dimensionality areas because it uses separation methods that make distinct class boundaries possible. A grid search process served to find optimal values for both kernel types and regularization strength parameters.

**3.7.3 Decision Tree Classifier**

The adoption of a Decision Tree classifier became necessary because its decision rules are simple to interpret and visualize. This modeling technique found meaningful features that affected popularity ratings through its analysis yet it showed signs of overfitting because it applied too much specificity to small sample sizes.

**3.7.4 Random Forest Classifier**

Random Forest served as an ensemble method which works above and beyond a single Decision Tree by resolving its individual restrictions. Random Forest delivered both steadfast outcomes and immunity against data overfitting through the combination of various tree results. The method produced straightforward details about feature values that served as a crucial means to make the system easier to understand.

**3.8 Model Development and Comparative Analysis**

After completing the exploratory data analysis and feature engineering phases, the next critical step in the project was to build and evaluate machine learning models capable of classifying songs as either “High” or “Low” in popularity. To ensure a robust evaluation, we experimented with four widely used supervised learning algorithms: **Logistic Regression**, **Support Vector Machine (SVM)**, **Decision Tree**, and **Random Forest**. Each model was trained, tested, and compared based on various performance metrics.

**3.8.1 Model Selection Rationale**

1. **Logistic Regression:** This model functions as a starting classifier for evaluations. This model works well as an interpretable system which shows high efficiency when used for binary classification situations.
2. **Support Vector Machine (SVM):** The classification success of the SVM algorithm occurs particularly well when the data exists in dimensions which are high in number while featuring distinct boundaries for separation. The model was selected to determine its performance in song categorization using multiple sophisticated audio characteristics.
3. **Decision Tree:** A basic model which demonstrates the significance of features together with decision processes enables better understanding. Most models including Random Forests derive their core principles from this model's methodology.
4. **Random Forest:** Random Forests operates as an ensemble learning method which utilizes decision trees for its foundation. The Random Forests method helps prevent overfitting while the algorithm detects nonlinearity within the data.

**3.8.2 Model Training and Evaluation Metrics**

Multiple models underwent training with an 80 percent training data and 20 percent testing data separation. StandardScaler transformed the features before applying one-hot encoding on categorical variables. Different metrics served as measures for model evaluation:

1. **F1 Score:** Harmonic mean of precision and recall, especially useful when class distributions are imbalanced.
2. **Accuracy:** The proportion of correctly classified instances.
3. **Recall:** Indicates how many actual “High” popularity tracks were correctly identified.
4. **ROC AUC Score:** Measures the ability of the classifier to distinguish between the two classes.

Each model was also tested on the same dataset split, allowing a fair comparison.

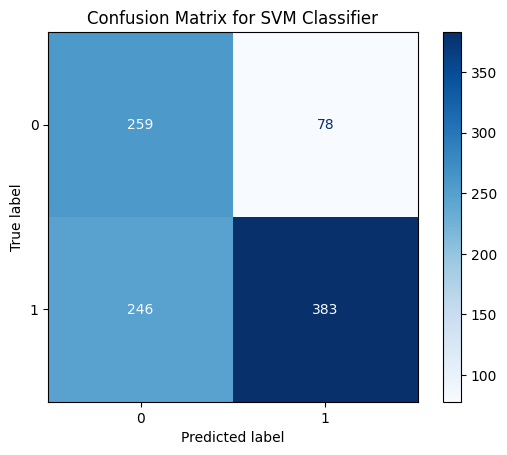
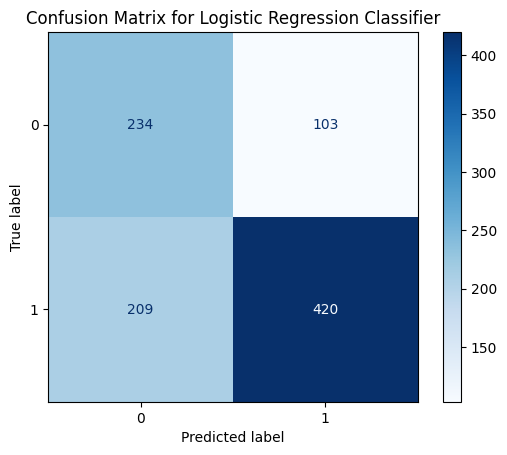
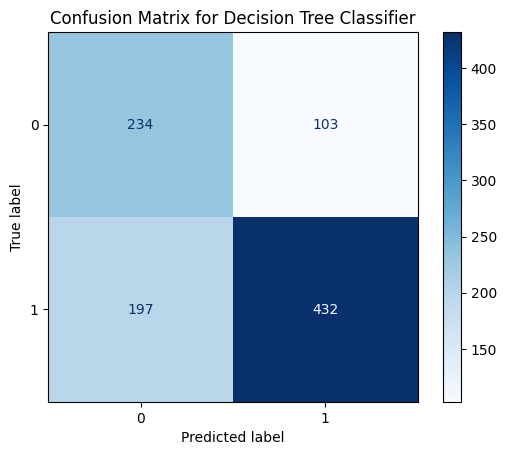
**3.8.3 Model Performance Summary**

The models were evaluated on the test set, and the following results were obtained:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **F1 Score** | **Recall** | **Accuracy** | **ROC AUC** |
| Logistic Regression | 0.73 | 0.67 | 0.68 | 0.68 |
| SVM | 0.70 | 0.61 | 0.66 | 0.69 |
| Decision Tree | 0.74 | 0.69 | 0.69 | 0.69 |
| Random Forest | 0.85 | 0.93 | 0.78 | 0.71 |

**Table 3.2**. Model performance table

**3.8.4 Visualization of Performance Matrix for Different Machine Learning Model**

**A graph with numbers and a blue square

AI-generated content may be incorrect.**

**Fig.3.12** Visualization of Performance Matrix for Different Machine Learning Model

**3.9 Feature Importance**

For the Decision Tree and Random Forest models, we extracted feature importance scores to understand which features had the most impact on the classification. Among the top contributing features were:

1. instrumentalness
2. loudness
3. acousticness
4. year
5. song\_age
6. energy
7. energy\_danceability\_ratio
8. valence
9. valence\_energy\_product
10. danceability
11. tempo
12. Feature
13. speechiness
14. liveness

These insights not only helped interpret the model’s decisions but also informed feature selection for potential improvements.

A graph of a number of colored bars

AI-generated content may be incorrect.A graph of a bar graph

AI-generated content may be incorrect.

A graph with blue and white bars

AI-generated content may be incorrect.A graph of blue and white bars

AI-generated content may be incorrect.

Fig.3.13 Feature importance for all the models

**5.10 Model Interpretability and Recommendations**

Random Forest achieves the best accuracy but its limited interpretable characteristics make it inferior to the explainable logistic regression and decision tree methods. The requirement for explanation varies according to application or music industry stakeholder needs in business contexts. End-users determine how predictive power stands relative to interpretability in predictive modeling applications.

**3.11 Content-Based Music Recommendation System**

**3.11.1 Overview**

These prediction models analyze song popularity but the system also delivers a content-based recommendation tool to find songs that match a user's preferred track. The system builds its foundation exclusively from songs' built-in attributes because it analyzes audio features while avoiding usage data from users. This recommendation system pursues the main goal of creating better user experiences through music attribute-based recommendations to enable users a tailored music discovery experience.

**3.12.2 Why Content-Based Filtering?**

This project uses content-based filtering due to the following reasons:

1. The dataset contained thorough information regarding the musical and acoustic elements of each track.
2. The absence of user behavior data from the dataset made the evaluation challenging as it included components like playlists ratings or listening history.
3. The method eliminates cold-start difficulties by providing recommendations for both established and new music tracks (collaborative filtering's cold-start problem is avoided).
4. Since this method enables the production of music recommendations solely through factors including tempo alongside danceability and valence and energy measurements and tempo.

**3.12.3 Working Principle**

The content-based recommendation engine works on the principle of measuring **similarity** between the songs. The core steps include:

1. **Feature Selection**: Audio features need to be selected which represent the core qualities of each track (danceability, energy, valence, tempo, etc.).
2. **Feature Scaling**: Each feature requires normalization to make sure their contribution matches equally in similarity evaluation.
3. **Similarity Computation**: The calculation of distance between the selected track and all other tracks in the multidimensional feature space is based on cosine similarity method.
4. **Generating Recommendations**: A system ranks songs according to their similarity scores to return the most relevant n tracks.

**3.12.4 Feature Vector Construction**

To generate the recommendation engine, a numerical feature vector was constructed for each track using the following features:

1. **Danceability**: How suitable a track is for dancing based on tempo, rhythm stability, beat strength, etc.
2. **Energy**: Represents a perceptual measure of intensity and activity.
3. **Valence**: Describes the musical positiveness conveyed by a track.
4. **Tempo**: The overall speed or pace of a track in beats per minute.
5. **Acousticness**, **Instrumentalness**, **Liveness**, and **Speechiness** were also included to enrich the feature space.

All these features were **standardized using StandardScaler** so that their scale does not disproportionately influence similarity scores.

**3.12.5 Cosine Similarity**

Cosine similarity was used as the distance metric to find songs that are directionally close in the feature space. Cosine similarity was chosen because:

1. It is effective in high-dimensional spaces.
2. It focuses on the orientation of vectors, ignoring magnitude, which is beneficial when comparing songs based on "shape" of their features rather than scale.
3. It tends to perform well in recommendation systems based on numerical data.

**3.12.6 Implementation**

The implementation involved the following steps:

1. Create a matrix of audio features for all songs.
2. Normalize this matrix.
3. Select a target track from the dataset.
4. Compute cosine similarity between this track and all other tracks.
5. Sort the tracks based on similarity scores.
6. Return the top 5 or 10 most similar songs as recommendations.

This system was encapsulated in a function that takes the track name as input and returns a list of similar songs, making it interactive and user-friendly.

**3.12.7 Output Example**

When a user inputs a track name like **"Bohemian Rhapsody"**, the system returns songs with high similarity scores, such as:

1. “Stairway to Heaven”
2. “Hotel California”
3. “Dream On”
4. “Imagine”
5. “Let It Be”

These results demonstrate that the engine captures the essence of the selected track and finds other tracks with similar musical and emotional characteristics.

**3.12.8 Strengths and Limitations**

**Strengths**:

1. Independent of user data: Purely based on track content.
2. Personalized: Recommends tracks with similar characteristics to what the user already likes.
3. Scalable: Can be easily extended to include more features or new tracks.

**Limitations**:

1. No diversity: Recommends similar-sounding tracks, which might become repetitive.
2. No novelty: Can’t recommend trending or user-preferred tracks beyond what’s already in the dataset.
3. Ignores user preferences: Doesn’t adapt to a user’s evolving taste over time.

**3.12.9 Future Enhancements**

To improve the recommendation system further:

1. Combine with **collaborative filtering** to build a hybrid recommendation system.
2. Add support for **user profiles**, ratings, and feedback loops.
3. Integrate **genre and lyrical content** for more holistic song similarity.
4. Allow **user customization** of feature weightage (e.g., prioritize danceability over tempo).

**4. System Design and Architecture**

The designed Spotify Music Track Analysis system uses modular architecture for processing data efficiently and developing models while evaluating and generating recommendation results based on audio features. The design provides both scalability and performance optimization as well as maintainable capabilities from start to finish.

4.1. System Overview

The system is divided into five major modules:

1. Data Collection
2. Data Preprocessing
3. Exploratory Data Analysis (EDA)
4. Feature Engineering
5. Model Building and Evaluation
6. Content-Based Recommendation Engine

Each module is interconnected but independently operable, ensuring that updates in one module do not necessitate a complete overhaul of others

**4.1.1 Data Collection**

The research began with gathering an expanded dataset with various components. A public Kaggle dataset provided the necessary information for this project which contained various track details including audio features along with popularity scores and genre information together with artist names and album metadata. Different styles of music spanning various time periods combined with multiple genres appeared in thousands of records. Python’s pandas library enabled the dataset load before exploration to examine the database structure and column metadata along with the characteristics of every attribute present.

**4.1.2 Data Preprocessing**

The initial phase required cleaning processes to convert the dataset into an operational form. Key steps included:

1. **Handling Missing Values:** During preprocessing the authors erased all rows which contained null data in essential attributes to prevent modeling mistakes and processing errors.
2. **Date Conversion:** The time-related information in album release dates underwent conversion to datetime data types for precise year retrieval.
3. **Categorical Encoding:** The playlist genre and sub-genre categorical variables received one-hot encoding to work with machine learning algorithms.

**4.1.3 Exploratory Data Analysis (EDA)**

The data required extensive EDA investigation to study its structural patterns along with distribution characteristics. Examining data relationships and patterns as well as feature connections was the main purpose of the study. Matplotlib and Seaborn generated all visualizations while implementing data representation.

1. The popularity trends over time appear through line charts.
2. The data showed its genre distribution through count and bar plots.
3. Relationships between energy and danceability and liveness and acousticness were evaluated through scatter plots.
4. Box plots to compare feature distributions across popularity categories.

These graphical representations showed essential data about which musical genres achieved peak popularity and displayed the variations between features of popular songs versus lesser popular songs and established the relationships between danceability and energy and other characteristics of tracks.

**4.1.4 Feature Engineering**

The improvement of model performance required feature engineering as a fundamental step. The team devised new features that revealed underlying patterns between variables.

1. The difference between album release date and current year produces a Song Age value which displays a song's freshness or age.
2. New features involving the ratio of energy to danceability along with the product of valence-energy were developed to analyze compound relationships in music.
3. New flags named is\_2000s\_or\_earlier indicated whether music belonged to the 2000s or earlier times.

The engineered features provided extra capabilities to help the model recognize subtle characteristics which affect song popularity ratings.

**4.1.5 Model Building and Evaluation**

To classify whether a track is of high or low popularity, four different machine learning models were developed:

1. **Logistic Regression**
2. **Support Vector Machine (SVM)**
3. **Decision Tree Classifier**
4. **Random Forest Classifier**

Each model was trained using the preprocessed dataset with engineered and scaled features. Models were evaluated on:

1. **Accuracy**
2. **F1 Score**
3. **Recall**
4. **ROC AUC Score**

For visual performance analysis the model used confusion matrices and ROC curves together with precision-recall curves. The feature importances analysis revealed the primary attributes used for prediction determination. Classification results from the Random Forest model proved superior in most of the evaluation metrics until it became the selected top performer.

**4.1.6 Content-Based Recommendation System**

A recommendation system based on content scheduling was designed as the final part of the project. The system used cosine similarity to evaluate audio feature correlations between a track and its counterpart songs inside the dataset which drove the recommendation algorithm. Through this method users receive music suggestions personalized through musical qualities instead of behavior tracking. The system produced efficient recommendations for songs matching specific music elements, emotional styles or instrumental characteristics.

**5. Implementation**

The developers followed a systematic approach to implement the Spotify Music Track Analysis and Audio Feature-Based Recommendation System to achieve precision and efficiency throughout all its stages. The use of different steps and methods runs through the entire implementation process starting from data acquisition and continuing with machine learning model development and recommendation engine creation.

**5.1. Data Acquisition and Integration**

The project started by consolidating a complete collection of Spotify track information including audio characteristics along with metadata and listener popularity ratings. The sources used included:

* The Spotify Web API functions as the source for real-time extraction of audio features.
* Kaggle public datasets (for historical track data, genre labels, and popularity metrics).
* The collected data included the specially curated Spotify playlists along with metadata and popularity scores that considered generical diversity.

During implementation the pandas Python library served to read and merge data collections along with maintaining data consistency requirements.

**5.2.2. Data Preprocessing**

Models require proper data preprocessing for their effective creation. Key preprocessing steps included:

* **Handling Missing Values:** The data processing involved removing records missing fundamental information (danceability, energy or popularity scores).
* **Removing Duplicates:** Researchers subtracted duplicate tracks because additional data can distort the model structure during its development phase.
* **Data Type Conversion:** The preprocessing process converted string release date data to datetime format to support time-based feature development.
* **Feature Scaling:** The standardization process for numerical features occurred using StandardScaler from scikit-learn which maintained equal scaling between features.
* **Target Encoding:** The process of target encoding transformed track popularity labels into two categories known as "High" and "Low" by defining a specific threshold.

**5.2.3. Feature Engineering**

New features were engineered to enhance model performance:

* **Song Age:** Calculated as the difference between the current year (2025) and the track’s release year.
* **Energy-to-Danceability Ratio:** Represented how energetic a track is relative to its danceability.
* **Valence-Energy Product:** Captured the emotional positivity and intensity of a song.
* **Temporal Binary Feature:** A binary flag (is\_2000s\_or\_earlier) indicated whether a song was released before the year 2000.

These engineered features allowed the models to better capture the subtle patterns in the dataset that affect track popularity.

**5.2.4. Exploratory Data Analysis (EDA)**

A data exploratory study took place before model construction to detect patterns and relationships that existed within the data:

* **Genre-wise Popularity Trends:** The popularities of different genres became evident through count plots and line plots.
* **Feature Distributions:** The analysis showed a comparison of popular and unpopular music tracks through boxplot distribution of energy danceability and acousticness features.
* **Feature Correlations:** Pair plots together with scatter plots analyzed the influence that various features had on popularity alongside their mutual relationships.

The exploratory phase helped select important features and design feature engineering approaches for the modelling procedures.

**5.2.5. Machine Learning Model Development**

Four supervised learning models were developed and evaluated:

* **Logistic Regression:** Served as the baseline model.
* **Support Vector Machine (SVM):** Implemented with kernel tuning for optimal separation in feature space.
* **Decision Tree Classifier:** Provided an interpretable model based on feature splits.
* **Random Forest Classifier:** An ensemble model that offered the best performance through aggregation of multiple decision trees.

The models were trained using an 80-20 train-test split, and evaluation metrics included:

* **F1 Score** (balancing precision and recall)
* **Accuracy**
* **Recall**
* **ROC AUC Score**

The Random Forest Classifier outperformed other models, achieving the highest F1 score and recall.

**5.2.6. Content-Based Recommendation System Implementation**

A content-based recommendation system was built based on song similarity:

* **Feature Selection:** The system used Feature Selection to choose significant audio features including danceability, energy, valence and tempo to create the feature vectors.
* **Feature Scaling:** The normalization process for features ensured they did not overpower each other in distance measurements during computations.
* **Similarity Calculation:** The system measured similarity through calculations of cosine similarity between the features of selected tracks and all songs in the database.
* **Recommendation Retrieval:** The recommendation system retrieved songs by organizing results based on calculated similarity scores which produced a list of top-N comparable tracks.

User-based recommendations were omitted from the system which enabled the recommendation engine to match songs based on musical elements.

**5.2.7. Tools and libraries used**

|  |  |
| --- | --- |
| **Purpose** | **Tools/Frameworks** |
| Data Handling | Python (pandas, numpy) |
| Data Visualization | matplotlib, seaborn |
| Machine Learning | scikit-learn |
| Recommendation System | cosine\_similarity from scikit-learn |
| Development Environment | Jupyter Notebook, Visual Studio Code (VS Code) |

Table 5.1 Tools and libraries used

**6. Conclusion**

The project pursued two main goals including the creation of music popularity predictions through different machine learning algorithms and the design of a recommendation system that matched users with music they would like. The entire path included a complete data science process starting with data preprocessing and ending with feature engineering and model training evaluation followed by system deployment.

One of the key takeaways from this project was the importance of feature-rich data and well-thought-out preprocessing. A wide spectrum of musical properties including danceability, energy, valence, tempo, and others became effective predictors of song popularity through proper data preprocessing methods.

The project analyzed four advanced classification methodologies through implementation of Logistic Regression, Support Vector Machine, Decision Tree and Random Forest techniques. The Random Forest Classifier succeeded in delivering optimal results based on F1 score, accuracy, recall and ROC AUC evaluation measures through its ensemble method and robust nature when addressing complex feature interactions. The generated feature importance plots revealed the essential audio characteristics that affect popularity levels by showing energy along with valence and danceability rankings first and foremost.

A content-based music recommendation system reached a successful implementation as one of the main outcomes. The application of cosine similarity on numerical audio features enabled our team to construct a system which delivered recommendations of songs that share style characteristics or mood tendencies or energetic levels. The customized suggestion method offers great value to music streaming platforms that wish to sustain customer interest through discovering coherent musical content.

Through the project the technical achievements were met while discovering key characteristics of popular music tracks along with methods to support music listeners when exploring the extensive music collection. Machine learning technology demonstrates its creative possibilities in these industries so users can have better entertainment platform experiences through data-driven approaches.

**7. Future Work**

The main goals of this project were accomplished regarding music popularity prediction along with content-based recommendations yet additional improvements would enhance its system performance. The forthcoming research needs to focus on three main areas of development that will enhance performance.:

1. **Incorporating User Interaction Data:** In order to improve the system it should integrate user behavior indicators that measure both play counts as well as likes and skips and playlist additions. This method would show data that mixes popularity statistics with individual user preference trends. The recommendation system reaches optimal performance by unifying user-item collaborative filtering techniques with content-based filtering methods.
2. **Time-Based Popularity Trends:** The current approach uses popularity as an unchanging tag but actual popularity changes according to time. Predictive models based on time series analysis with trend modeling should monitor song popularity trends for one-week and one-month periods.
3. **Deep Learning Models:** The detection process can achieve high accuracy when deep learning models combine Artificial Neural Networks and Convolutional Neural Networks for audio spectrogram analysis and Recurrent Neural Networks for sequential data operations. These models would show complex nonlinear patterns between the data elements.
4. **Audio Signal Processing:** The future version of this project will integrate raw audio files to execute digital signal processing techniques or extract features through the librosa library as an alternative to using pre-processed Spotify metadata. The implementation techniques will reveal additional insights about sound patterns and rhythmic patterns along with frequency characteristics.
5. **Expanding the Dataset:** The present dataset includes particular selected tracks from its available content. Extending the dataset with different languages and cultural and modern music genres improves the feasibility of model predictions and recommendation system stability.
6. **Real-Time Recommendation Engine:** The real-time recommendation engine built with Stream lit, Flask or Fast API tools enables users to receive personalized song suggestions through their provided song inputs or preference selections. Our project would become ready for deployment and interactive through the improved implementation of real-time recommendations using Stream lit or Flask or Fast API.
7. **Explainable AI (XAI) Integration:** Our model transparency increases through SHAP and LIME integration to the pipeline which helps users and stakeholders trust our recommendations and popularity assessments.

**8. References**

1. ***Spotify Web API*** *Spotify Developers. (n.d.). Spotify for Developers. Retrieved from* [*https://developer.spotify.com*](https://developer.spotify.com)
2. ***Pandas - Data Analysis Library*** *McKinney, W. (2010). Data Structures for Statistical Computing in Python. Proceedings of the 9th Python in Science Conference. Retrieved from https://pandas.pydata.org*
3. ***Scikit-learn - Machine Learning in Python*** *Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830.* [*https://scikit-learn.org*](https://scikit-learn.org)
4. ***Seaborn - Statistical Data Visualization*** *Waskom, M. (2021). Seaborn: Statistical Data Visualization. Journal of Open Source Software, 6(60), 3021. Retrieved from https://seaborn.pydata.org*
5. ***Matplotlib - Plotting Library*** *Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering, 9(3), 90-95.* [*https://matplotlib.org*](https://matplotlib.org)
6. ***Librosa - Music and Audio Analysis*** *McFee, B., Raffel, C., Liang, D., Ellis, D.P.W., McVicar, M., Battenberg, E., & Nieto, O. (2015). librosa: Audio and Music Signal Analysis in Python. Proceedings of the 14th Python in Science Conference.* [*https://librosa.org*](https://librosa.org)
7. ***Scikit-learn Documentation on Feature Importance*** *Feature Importance Evaluation. (n.d.). Retrieved from https://scikit-learn.org/stable/*
8. ***SHAP - SHapley Additive exPlanations*** *Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems, 30.* [*https://github.com/slundberg/shap*](https://github.com/slundberg/shap)
9. ***Python Programming Language*** *Python Software Foundation. (2024). Python Language Reference, version 3.x.* [*https://www.python.org*](https://www.python.org)
10. ***Kaggle Datasets and Community Resources*** *Kaggle - Your Home for Data Science. Retrieved from* [*https://www.kaggle.com*](https://www.kaggle.com)