

**BSEF20M507**-Farwa Ali.

**BSEF20M527**-Ayesha Arshad.

**BSEF20M531**-Bisma Nadeem.

**BSEF20M536**-Zain Fatima.

MACHINE LEARNING PROJECT

***SPOTIFY RECOMMENDATION ENGINE***

**PROBLEM STATEMENT**

With the rapid growth of digital music platforms like Spotify, users are faced with an overwhelming number of songs and playlists to choose from. Navigating through this vast music library to find personalized recommendations that align with individual tastes and preferences has become a significant challenge.

The problem at hand is to design and implement a personalized music recommendation engine for Spotify users. The primary challenge lies in developing a model that can effectively analyse and understand the musical preferences of individual users based on their previous liked and disliked songs. This necessitates the training of a machine learning model to empower Spotify users to effortlessly explore new music tailored to their preferences, thereby enhancing their overall music listening experience.

**INTRODUCTION**

To address the above problem, the objective of our project is to develop a machine learning-based recommendation engine for Spotify users. By leveraging a dataset of **195** songs, encompassing both liked and disliked tracks, the aim is to create an efficient and accurate system that predicts and suggests new songs to users based on their musical preferences. By providing personalized recommendations, this engine will enhance the music discovery experience and enable users to explore a diverse range of songs that align with their unique tastes and preferences.

**ABOUT DATA SET**

Our data set is a collection of **100** liked songs and **95** disliked songs of an individual user from his/her playlist of spotify.

**Data Features** include:

* **Acousticness**: A measure from 0.0 to 1.0 indicating the likelihood of a track being acoustic. Higher values suggest a higher confidence that the track is acoustic.
* **Danceability**: A value from 0.0 to 1.0 representing how suitable a track is for dancing based on factors like tempo, rhythm stability, beat strength, and regularity. Higher values indicate higher danceability.
* **Duration**: The length of the track in milliseconds.
* **Energy**: A measure from 0.0 to 1.0 representing the intensity and activity of a track. Energetic tracks feel fast, loud, and noisy, while low-energy tracks have a more subdued quality.
* **Instrumentalness**: Predicts the presence of vocals in a track. Values close to 1.0 suggest a higher likelihood of the track being instrumental, while values above 0.5 generally represent instrumental tracks.
* **Key**: Indicates the key of the track using standard Pitch Class notation. Each integer value corresponds to a specific pitch.
* **Liveness**: A value indicating the probability of a track being performed live. Higher values suggest a greater likelihood of the track being recorded in a live setting.
* **Loudness**: The overall volume of a track in decibels (dB). Negative values indicate quieter tracks, while positive values indicate louder tracks.
* **Mode**: Represents the modality (major or minor) of a track. 1 denotes major, while 0 denotes minor.
* **Speechiness**: Detects the presence of spoken words in a track. Values above 0.66 suggest tracks that are primarily spoken, while values below 0.33 indicate non-speech-like tracks, typically music.
* **Tempo**: The estimated speed or pace of a track in beats per minute (BPM).
* **Time Signature**: An estimated overall time signature of a track, indicating the number of beats in each bar or measure.
* **Valence**: A measure from 0.0 to 1.0 describing the positivity conveyed by a track. Higher values indicate a more positive or happy tone, while lower values suggest a more negative or sad tone.
* **Liked (**variable that has to be predicted**)**: **1** for liked songs, **0** for disliked songs

**PRE-PROCESSING OF DATA**

1. **Duplicate Removal**: Duplicates can introduce bias and adversely affect the performance of the recommendation engine. To ensure data integrity, we conducted duplicate removal using the **drop\_duplicates()** function from the pandas library. This operation removed any duplicate entries present in the dataset, ensuring that each song is represented only once.
2. **Handling Missing Values**: Missing values can impact the accuracy of the recommendation engine by introducing inconsistencies in the data. To address this, we employed the **dropna()** function from pandas to eliminate rows with missing values. This step effectively removed instances where essential song features were incomplete, ensuring a complete and reliable dataset for training our model.



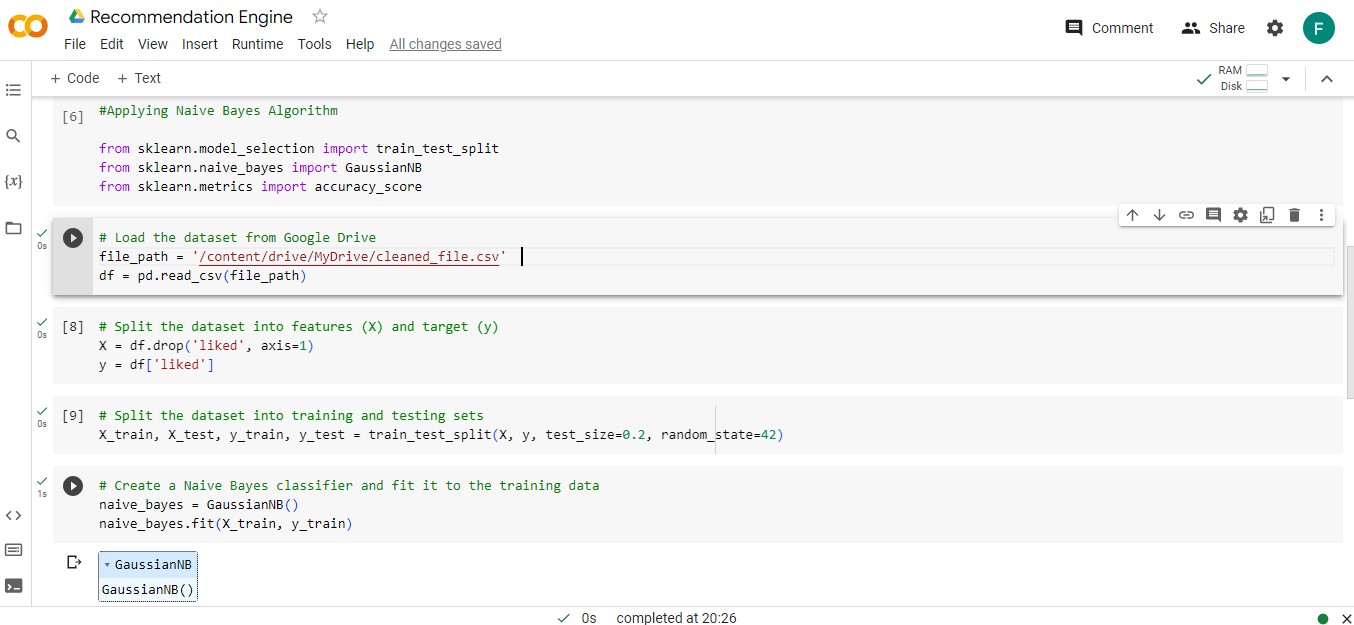
**SELECTION OF ALGORITHMS**

We selected following algorithms to check find their accuracy on our data set

* Naive Bayes
* K Nearest Neighbor
* Decision Tree

**NAIVE BAYES**

Naive Bayes is a probabilistic classification algorithm that assumes independence between features. Naive Bayes can be used in a music recommendation system to predict user preferences based on features such as danceability, acousticness and other metadata. Naive Bayes is computationally efficient and performs well with high-dimensional data, making it suitable for large music libraries.

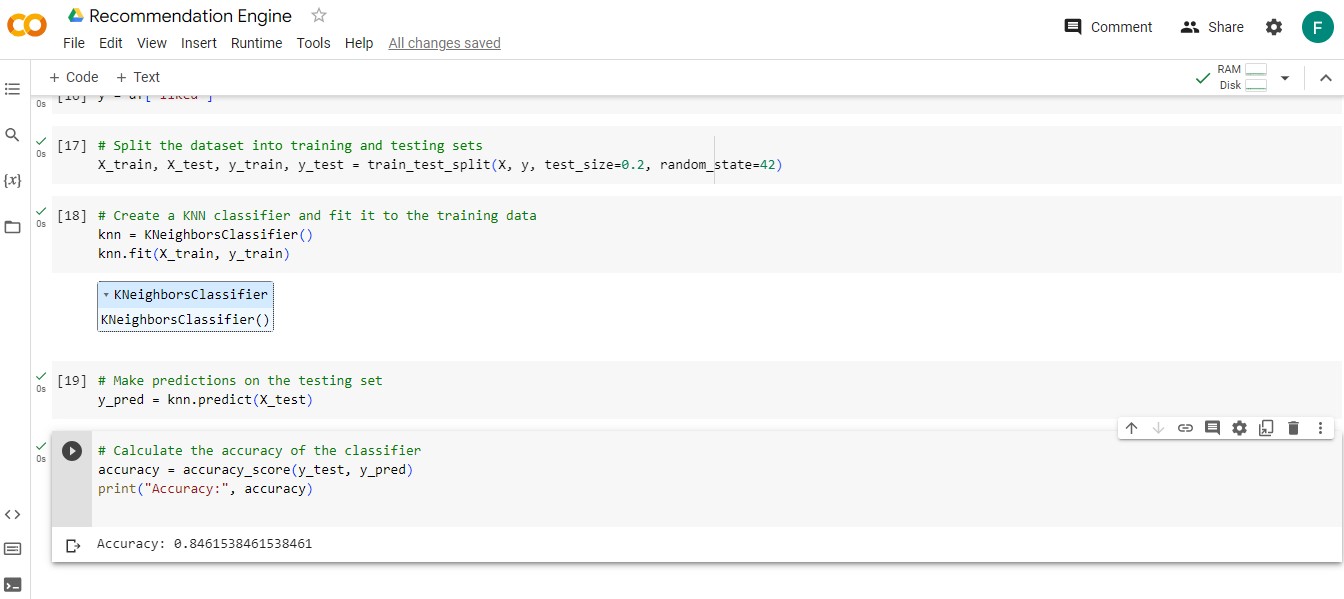




**K NEAREST NEIGHBOR**

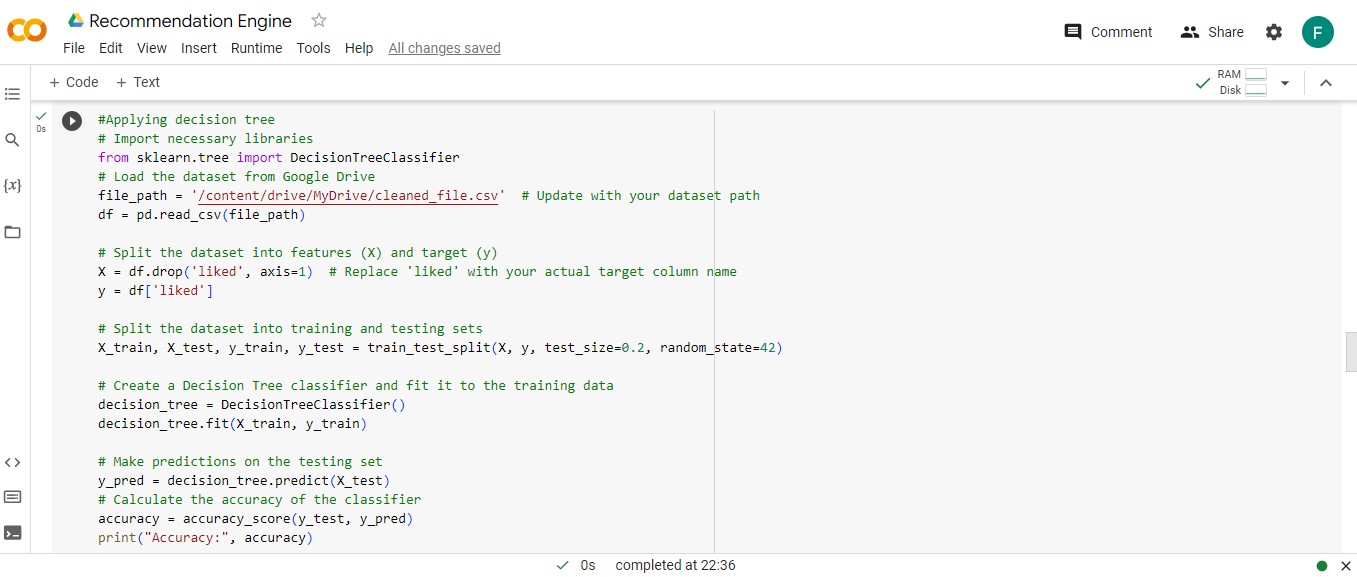
KNN is a non-parametric classification algorithm that classifies data points based on their similarity to neighboring points. In a music recommendation system, KNN can be used to find similar songs based on different features. KNN is intuitive and easy to implement, making it suitable for small to medium-sized music datasets.





**DECISION TREE**

Decision Trees create a tree-like model to make decisions based on feature values. They can be used in a music recommendation system to classify songs or identify patterns based on audio features or metadata. Decision Trees are interpretable and can capture complex interactions between features, making them suitable for both small and large music datasets.





**SELECTING DECISION TREE**

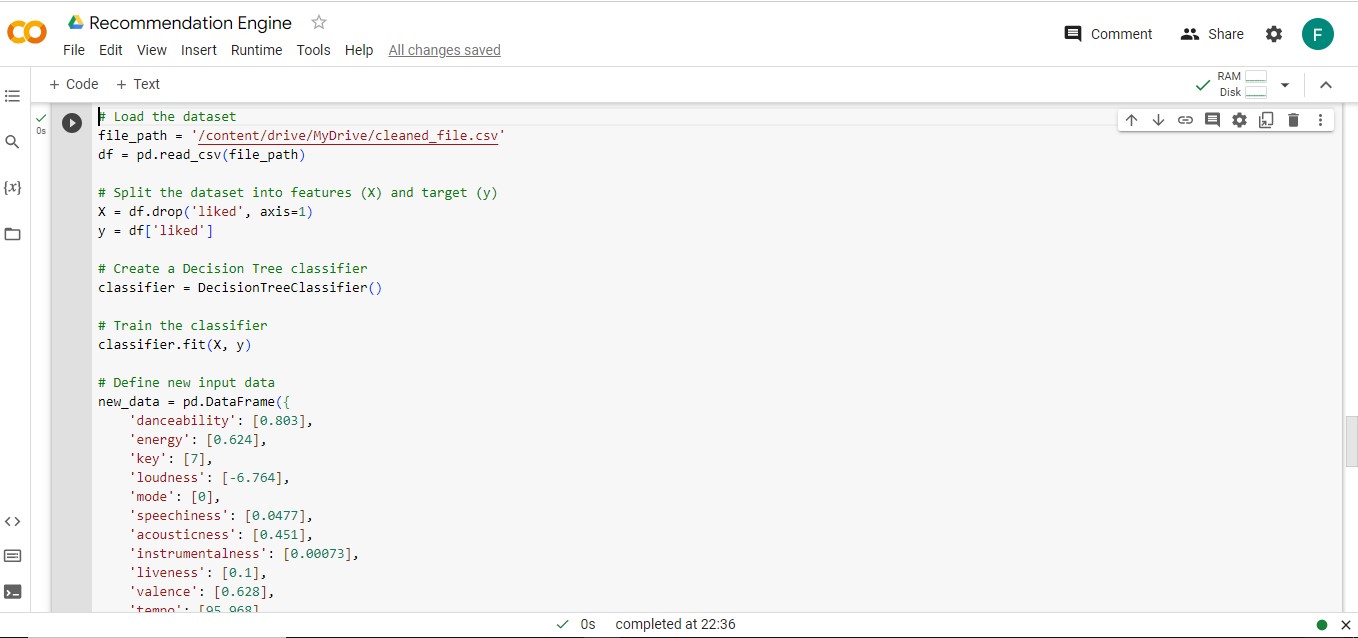
The Decision Tree algorithm was chosen for our Spotify recommendation engine due to its higher accuracy of **0.8717948** compared to other algorithms considered. Through evaluation and comparison of multiple algorithms, the Decision Tree algorithm demonstrated superior performance in accurately predicting song likability based on the provided features.

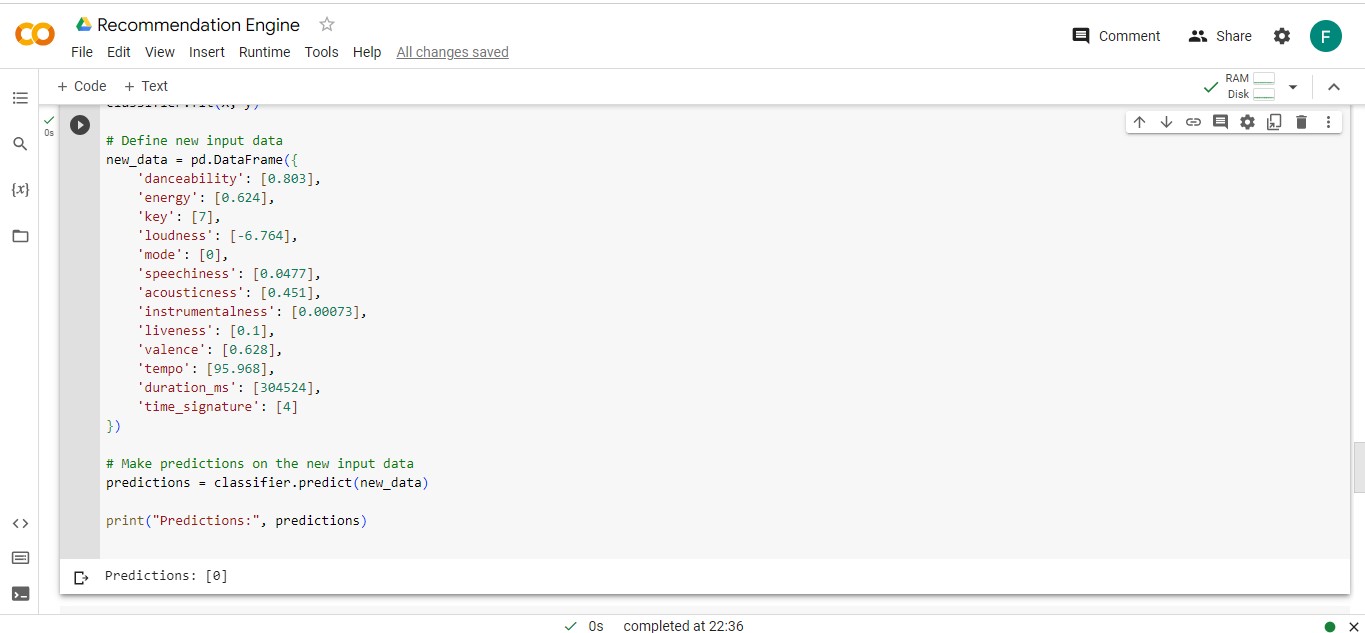
**PREDICTING DATA FROM DATASET AND INCOMING NEW DATA**

To make predictions in our Spotify recommendation engine, we utilized a Decision Tree classifier. Using the trained model, the classifier made predictions on the likability of the song based on its features. The predictions were obtained and represented the recommendation engine's estimation of whether the song would be liked or disliked. This prediction process allows our recommendation engine to categorize and suggest songs to users based on their unique preferences, enhancing their music discovery experience.

**PREDICTION FROM DATA SET**

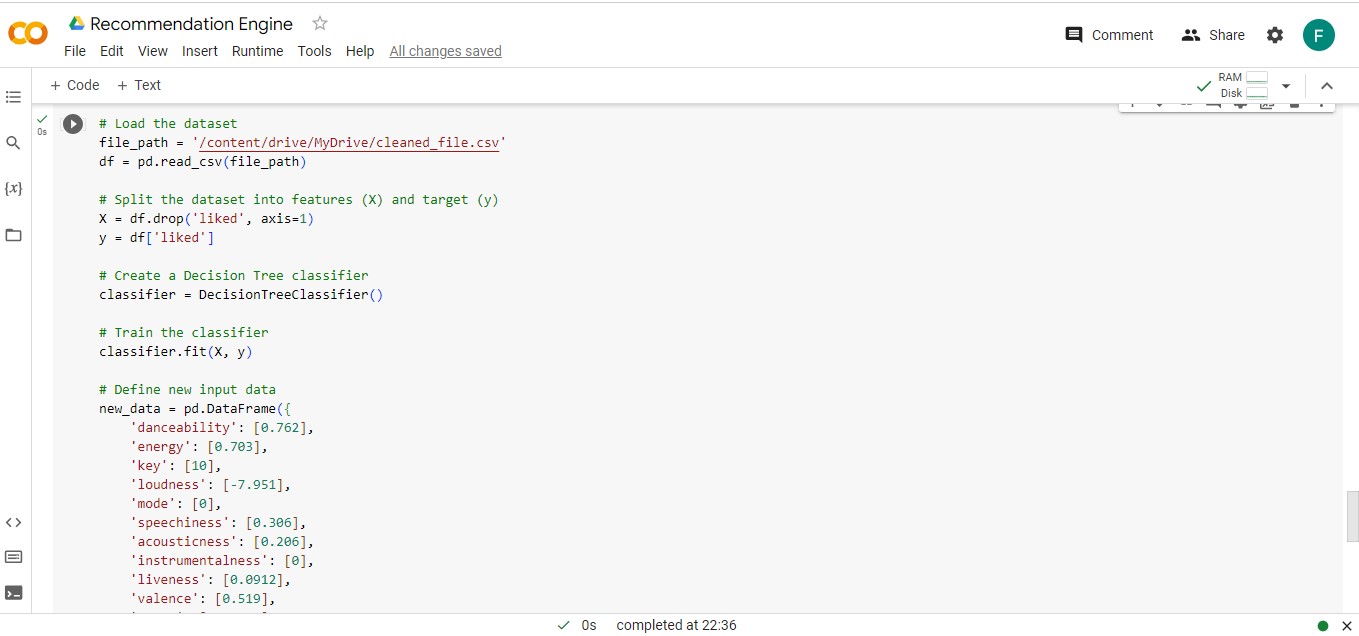
We predicted data taken from our data file to check it is predicting right.

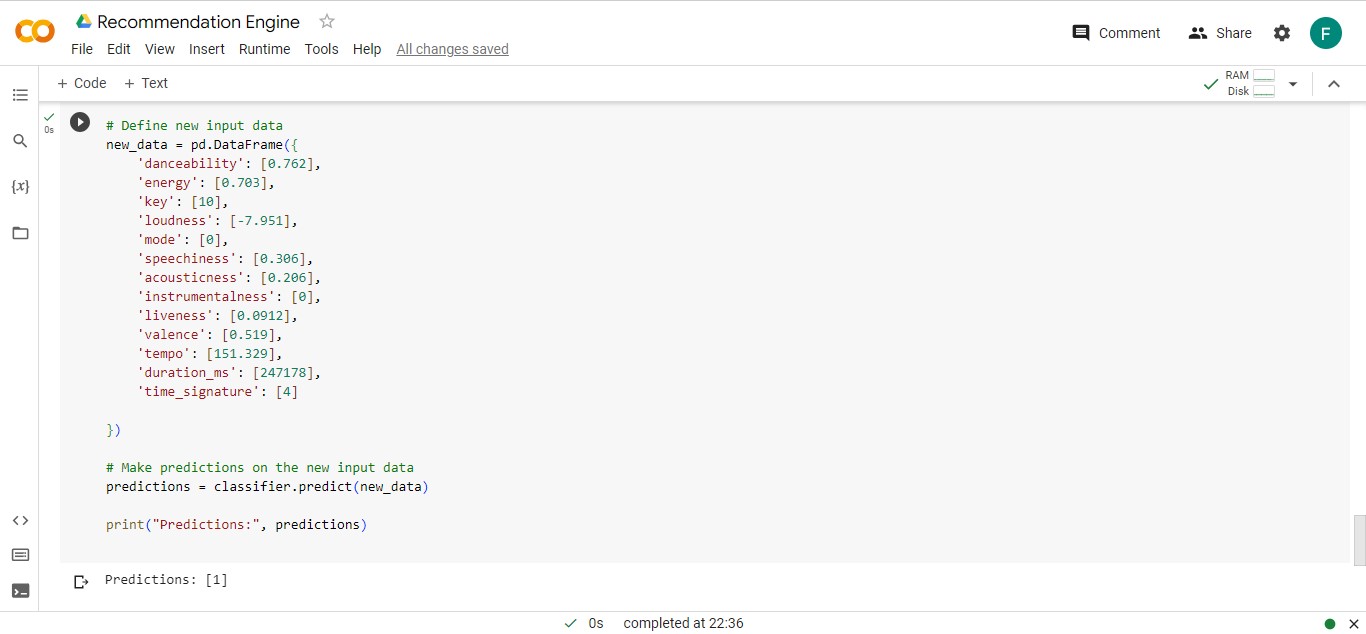
****

****

**PREDICTION FROM DATA SET**

We performed second prediction also from our dataset to check it is behaving right and giving right values or predictions

****

****

**PREDICTION OF INCOMING DATA**

Now, we predicted new incoming data by giving random values of different features in data set and we got a dislike prediction.





**CONCLUSION**

In conclusion, we have successfully developed a Spotify recommendation engine using machine learning techniques. By leveraging a dataset of songs and applying data preprocessing techniques to clean and prepare the data, we built a robust model. Through the use of the the Decision Tree classifier, we were able to predict song likability based on various song features. The recommendation engine provides personalized song recommendations, allowing users to discover new music tailored to their preferences. The accuracy achieved by our chosen algorithms ensures reliable and effective song recommendations, enhancing the overall music listening experience for Spotify users.