Spotify Recommendation Engine

By: Group 3

The Original Model

In the initial stage of our Spotify Recommendation Engine project, we established a solid foundation with a straightforward yet effective model architecture and user interaction. The model primarily relied on the K-means clustering algorithm, importing only two packages for data manipulation and clustering implementation. By grouping songs based on audio features, distinct clusters were formed, facilitating personalized recommendations. User interaction was seamless, with users inputting song names and the algorithm fetching corresponding clusters for tailored song suggestions. However, the model had its

LIMITATIONS:

- Does not handle outliers.
- Excessive running time.
- Lack of Personalization
- Limited Musical Diversity
- Fixed Number of Clusters

Enhancements

We made significant enhancements to our Spotify Recommendation Engine by integrating three powerful clustering algorithms - K-Means, DBSCAN, and Agglomerative Clustering - into our code model, elevating the song recommendation process. The benefits of this integration for the users are as follows:

• *Diverse and Personalized Recommendations*: The utilization of multiple clustering algorithms enables us to capture a broader range of musical nuances, ensuring that our recommendations cater to diverse user preferences. Users now receive personalized suggestions that align better with their unique tastes.

• *Robust Handling of Outliers*: The inclusion of DBSCAN and Agglomerative Clustering allows our model to effectively handle outliers or songs with unusual audio characteristics. This ensures that our recommendations remain robust and encompass songs from various musical styles and genres.

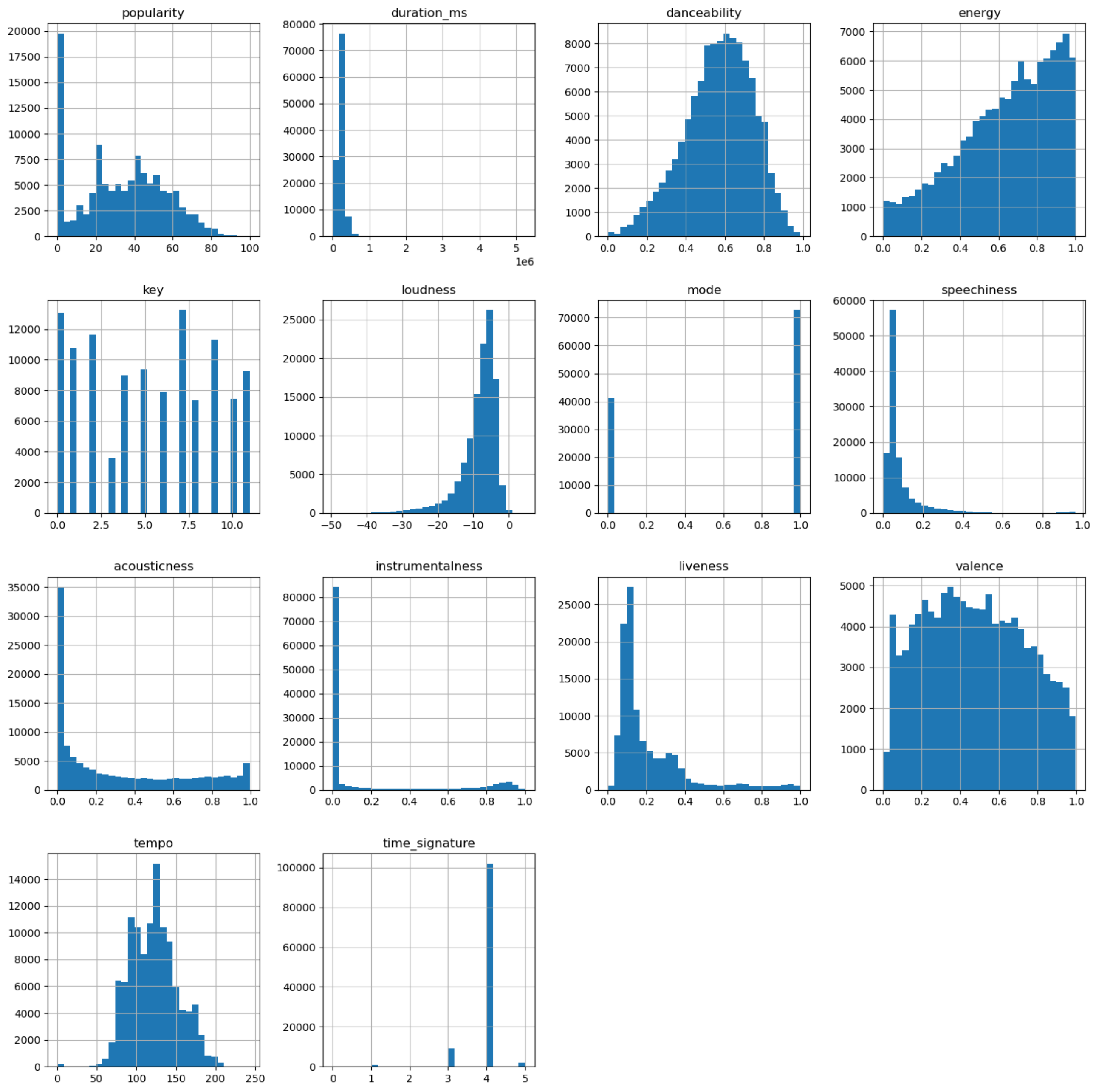
We successfully converted our model into a user-friendly Streamlit website, providing an intuitive interface for users to explore and interact with the recommendation engine. Users can now handpick specific values for various audio features, granting them greater control over the recommendations they receive. This interactive feature empowers users to discover songs that resonate with their preferred attributes, fostering a more engaging and personalized musical journey.

Findings from Data Exploration - Histograms and Heatmap

In the data exploration phase, we gained valuable insights into the relationships between variables and the optimal number of clusters for our Spotify Recommendation Engine. Through histograms, we visualized the distribution of various audio features, such as danceability, energy, and valence, revealing patterns in the data. These visualizations allowed us to better understand the distribution of song characteristics, enabling us to make informed decisions during model development.

Insights:

1. In the histogram for mood, there are two big peaks at either end, because songs are usually sad or happy.
2. Most songs have a similar time signature.
3. Most of the songs are speech heavy, which makes sense because there's only a small percentage of songs that are only instrumental.

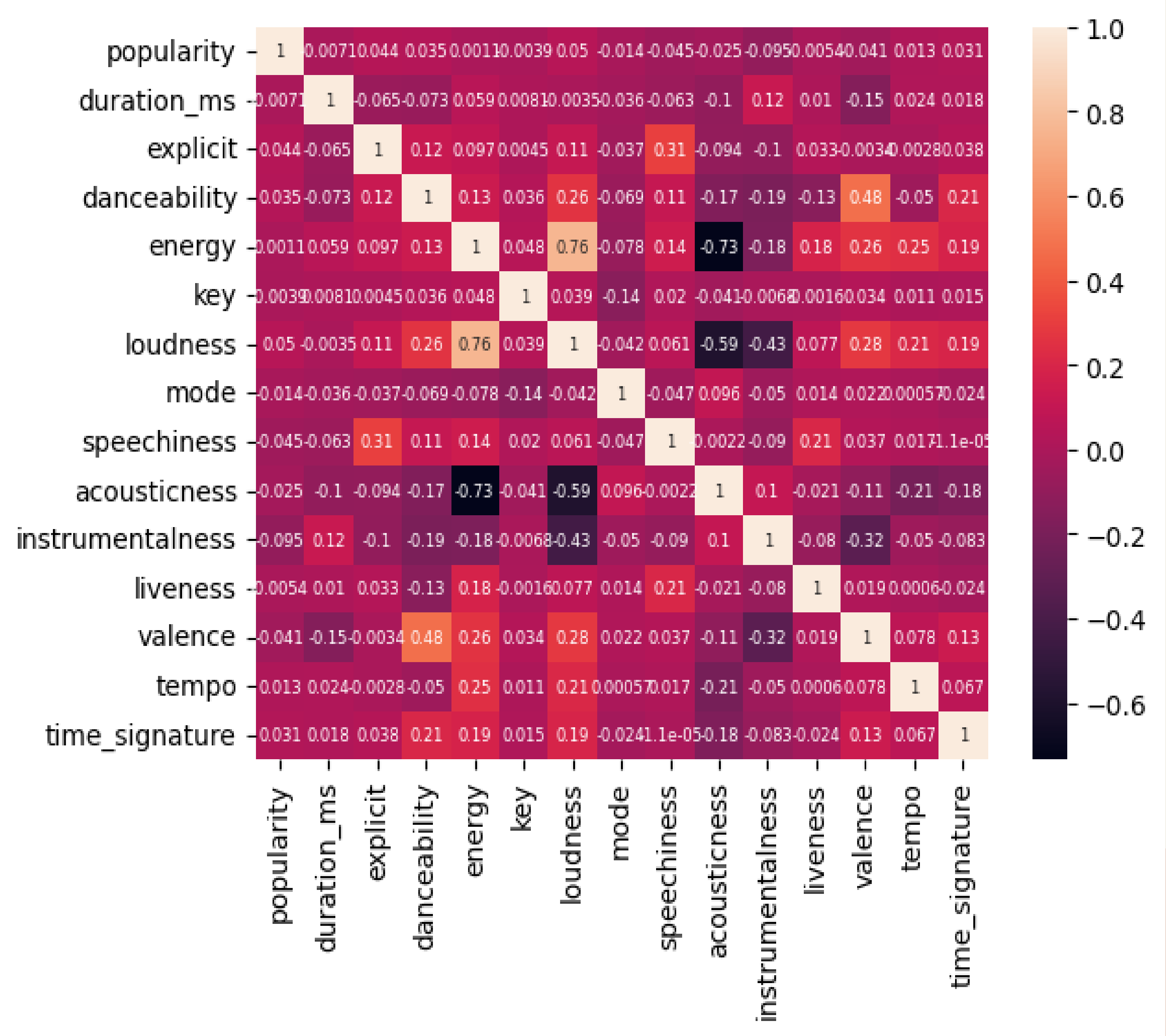


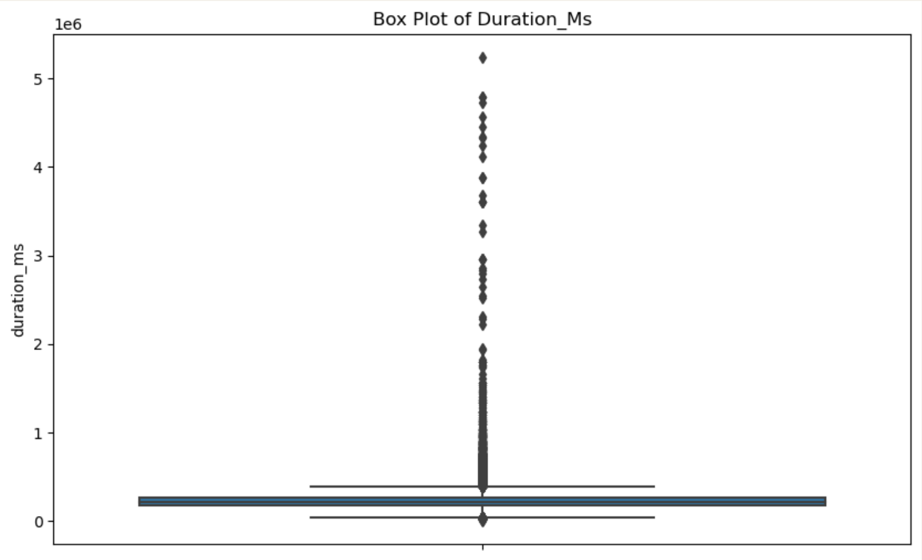
Heatmap

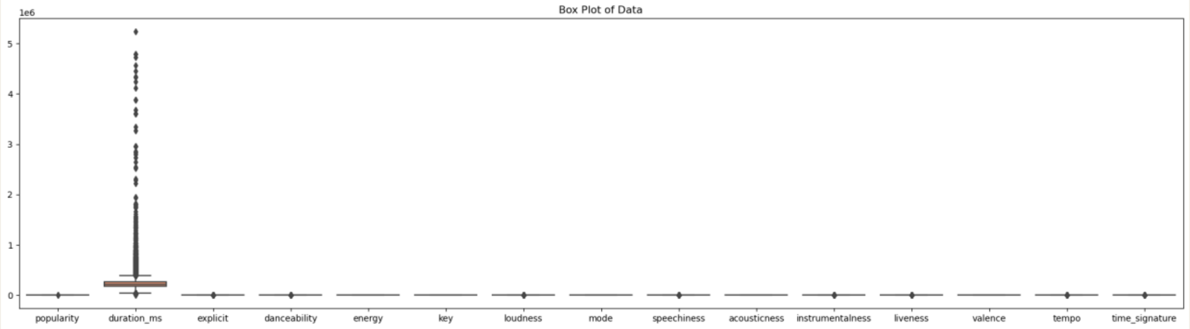
Additionally, we constructed a heatmap to explore correlations between the audio features and the clustering labels. The heatmap provided a comprehensive view of how each variable related to the clustering process, helping us identify key features that significantly impacted song grouping.

Insights:

1. Energy has a positive correlation with loudness and a negative correlation with acousticness. We can say this is true as acoustic songs are generally slow and do not induce as much energy.
2. Valence has a positive correlation with danceability, which makes sense because songs with high valence are positive and cheery.







Handling Outliers with Inter Quartile Range

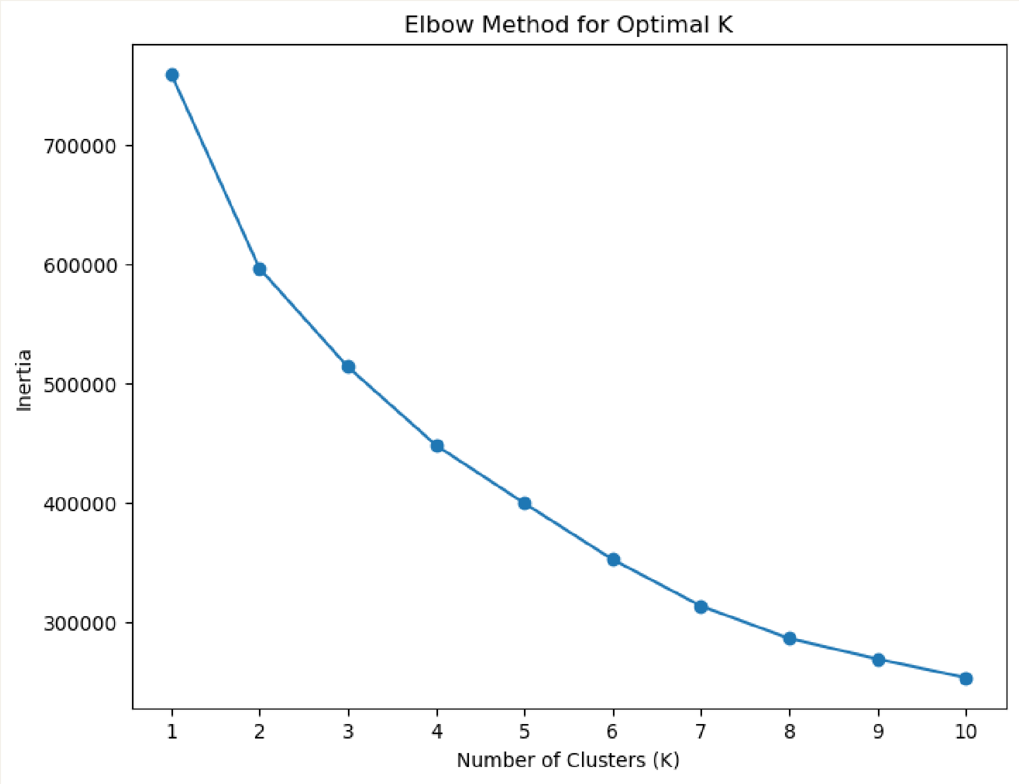
During data exploration, we detected outliers in the 'duration_ms' variable, which could potentially influence the clustering process and recommendation quality. To address this issue, we employed the Inter Quartile Range (IQR) method. By computing the IQR for 'duration_ms', we determined a threshold beyond which data points were considered outliers. We then removed these outliers to ensure a more robust clustering process.

Comparison of Clustering Algorithms in Three Models

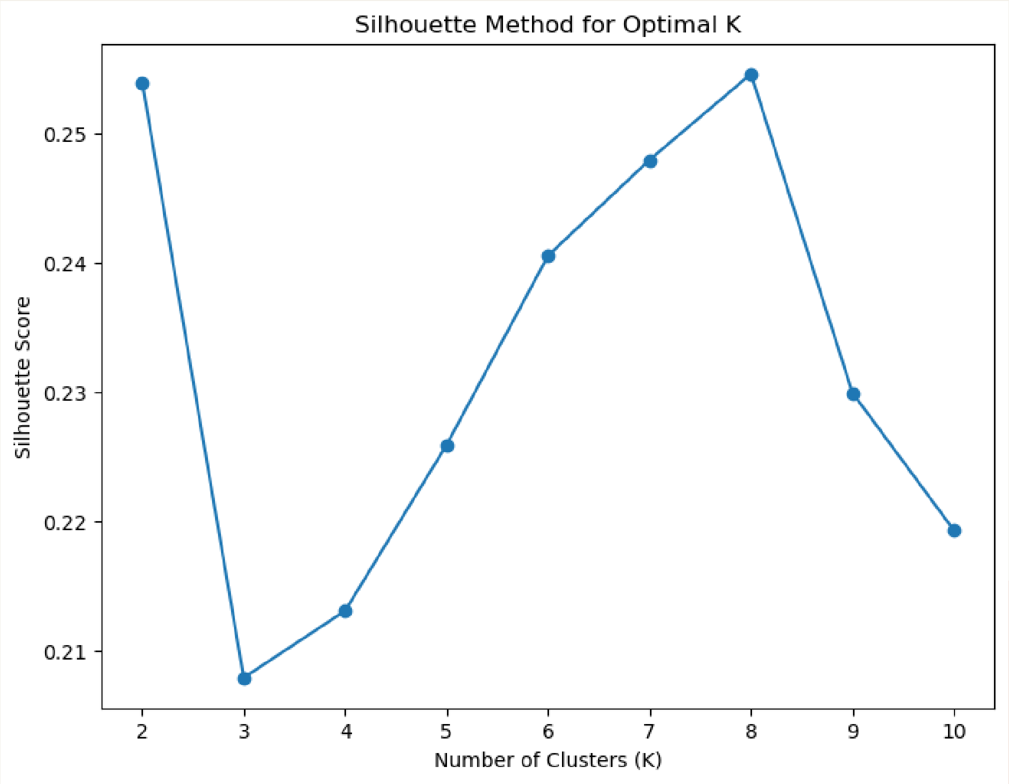
Features	K-Means	DBSCAN	Agglomerative Clustering
Model Complexity	Simple and easy to implement	More complex due to parameter tuning	Moderately complex, impacted by linkage methods
Handling Outliers	Sensitive to outliers, may affect cluster quality	Robust to outliers, automatically identifies noise points	Affected by outliers, depending on the linkage method
Cluster Shapes	Assumes clusters are spherical and of equal size	Adapts to any cluster shape, flexible	Flexibility in handling various cluster shapes
Number of Clusters	Requires pre-specification of clusters	Automatically determines the number of clusters	Requires specifying the number of clusters beforehand
Scalability	Efficient for large datasets	May be less efficient for large datasets	Efficient for small to medium-sized datasets
Personalization	Less personalized, may overlook finer preferences	Provides more personalized recommendations	Provides more personalized recommendations

Finding the Optimal Number of Clusters

Elbow Method



Silhouette Method

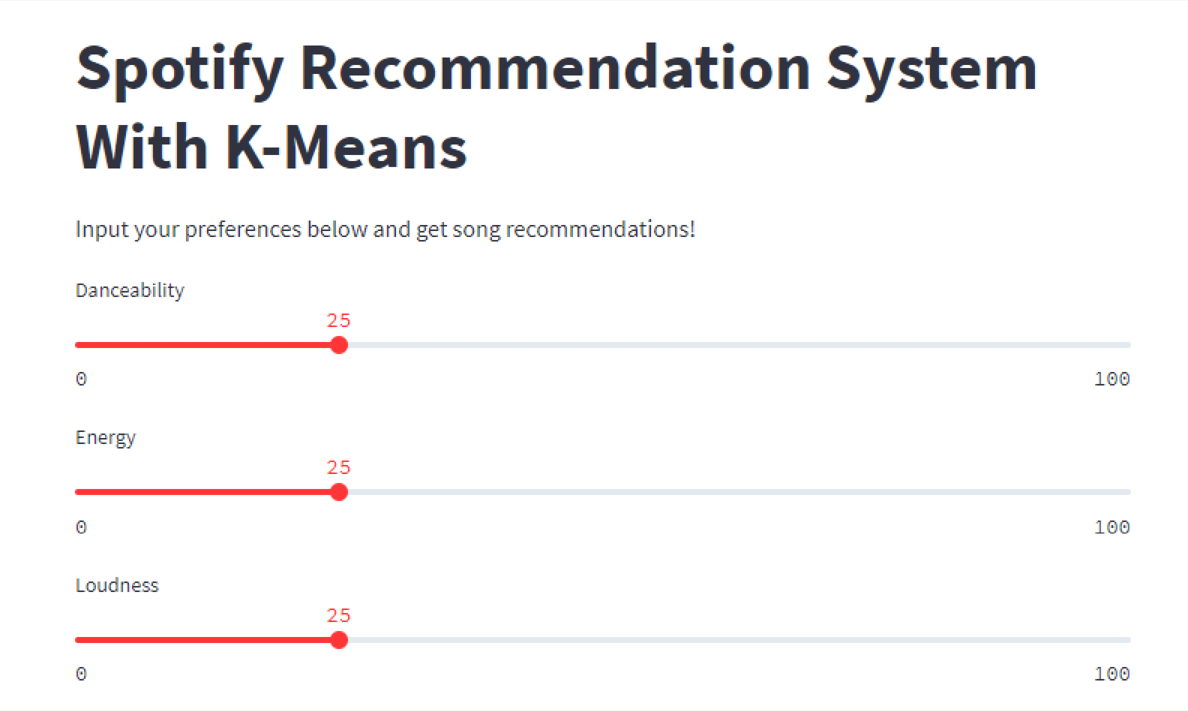


Incorporating User Feedback and its Impact

Our Spotify Recommendation Engine goes beyond model development to create an interactive and user-centric experience through our website. Utilizing sliders for audio feature selection, users can actively engage in fine-tuning their preferences, making the recommendation process more immersive and enjoyable.

By incorporating user accounts, we can store individual preferences, listening history, and feedback, leading to personalized and tailored song recommendations. Introducing genre selection sliders will enable users to refine recommendations based on their specific musical interests, enhancing user satisfaction. To maintain a balance between familiarity and novelty, we plan to add a novelty slider, which will allow users to discover new tracks while still enjoying their favorite songs.

The feedback loop and collaborative filtering will significantly enhance the engine's ability to understand and anticipate user preferences, resulting in more engaging and relevant recommendations



Future Enhancements

There are several exciting **future enhancements** and **integration possibilities** to elevate the user experience.

1. User Engagement Metrics :

2. Social Media Integration

3. Offline Mode

4. Content-Based Filtering

5. Integration with Spotify API or A Music App

6. Real-Time Updates