A yellow and black logo

AI-generated content may be incorrect.

**VIRGINIA COMMONWEALTH UNIVERSITY**

**INFO 648 – BUSINESS DATA ANALYTICS**

**PROFESSOR: NING LUO**

**Team Project**

**PREDICTING SPOTIFY SONG POPULARITY**

**Submitted by:**

**Group-02**

**Date of Submission: 05-05-2025**

**Group Member Names and Contributions**

**Aakash Kathirvel:** K Means, Association

**Chandhini Kerachan Muraleedharan:** Data preprocessing, KNN

**Matt Ferguson:** Logistic Regression

**Milan Chaudhari:** Decision Tree, Cost Benefit Analysis

**EXECUTIVE SUMMARY**

The project, commissioned by Spotify, was aimed at helping Spotify with recommending new songs to their users via their music streaming services. Using the given dataset of US Spotify chart-topping songs from 1988-2020, Spotify directive was to predict future popularity of newly released songs.

By using various supervised machine learning techniques, our team successfully managed to develop predictive models to determine whether a song will attain a popularity score of 64 and above. The ultimate task was to help Spotify proactively recommend popular songs to their users that resonated with them, and thereby improving user engagement and satisfaction.

The project was executed in three phases.

In the first phase, we were tasked with sample selection and data preprocessing where we applied an appropriate strategy on handling missing values and selecting a set of attributes which were most relevant in predicting a song's popularity. We built three classification models, comparing their performance in identifying the most accurate predictor of song popularity.

In the second phase, we reassessed the models using a profit-maximization framework based on revenues defined by the stakeholders and their respective costs associated with correct and incorrect song predictions. This aided us to align model performance with real-world business value and context.

The final phase of the project involved unsupervised clustering based on musical attributes like danceability and energy, with a focus on understanding the effects of valence of a song in predicting the song’s success and whether the effect differs across different types of music. We also applied association analysis in order to understand feature importance linked to popular songs.

Our findings offer actionable managerial insights for Spotify, including recommendations for future songs that Spotify should consider including, with a primary focus on higher user engagement and popularity scores that can help drive decision-making.

**SUMMARY FOR DATA PRE-PROCESSING**

To prepare the spotify data for predictive modeling, there were several data cleaning processes which were undertaken.

The dataset titled **“songs\_utf.csv”** was imported into Google Colab for preprocessing. It consisted of 1,500 records containing song metadata and audio features. The first step involved encoding categorical variables, specifically converting the 'explicit' attribute into a binary format. Irrelevant textual columns such as ‘artist’, ‘song’, and ‘genre’ were excluded to avoid noise during model training. A new binary target variable was constructed, identifying songs with a popularity score of 65 or higher as popular (1), and others as non-popular (0).

Any rows containing missing values were removed to maintain data integrity. Descriptive statistics were generated for numerical features, and a correlation matrix was used to identify potential multicollinearity, which was not significant. The final dataset, comprising 25 cleaned features, was standardized using appropriate scaling methods to ensure consistency before model training.

1. **Data loading and inspection:** The dataset was imported using pandas and initially inspected with **head(), info(), and describe()** to understand the structure and features of the dataset.

A heatmap of the correlation matrix was generated to explore relationships among numerical variables.

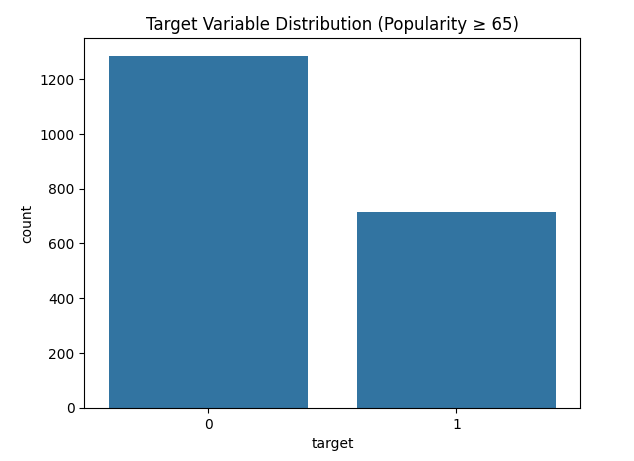
1. **Handling missing values:** Missing values were identified using **df.isnull().sum()**. All rows containing missing values were dropped to ensure that a clean dataset is ready for modelling.

A final check was done to confirm zero remaining missing entries.

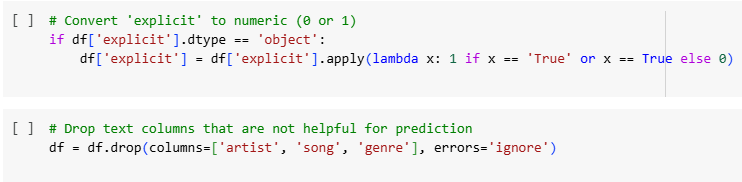
1. **Feature Engineering:** The explicit column, originally containing boolean or string values, was encoded into binary form (1 for explicit, 0 otherwise).

A new target variable target was created where songs with a popularity score >=65 were labeled as 1(popular), and others as 0(not popular).

A distribution plot of the target variable was generated to visualize class imbalance.



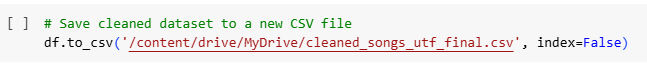
1. **Feature Selection:** Non predictive text columns such as artist, song, and genre were removed from the dataset to streamline modeling and reduce noise.



1. **Output:** The cleaned and transformed dataset was saved in two stages:

**cleaned\_songs\_utf.csv**- initial cleaned version.

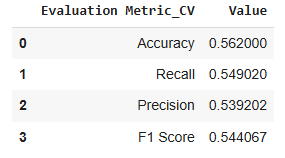
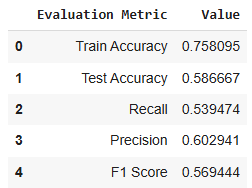
**cleaned\_songs\_utf\_final.csv**- final version after verifying no missing values.



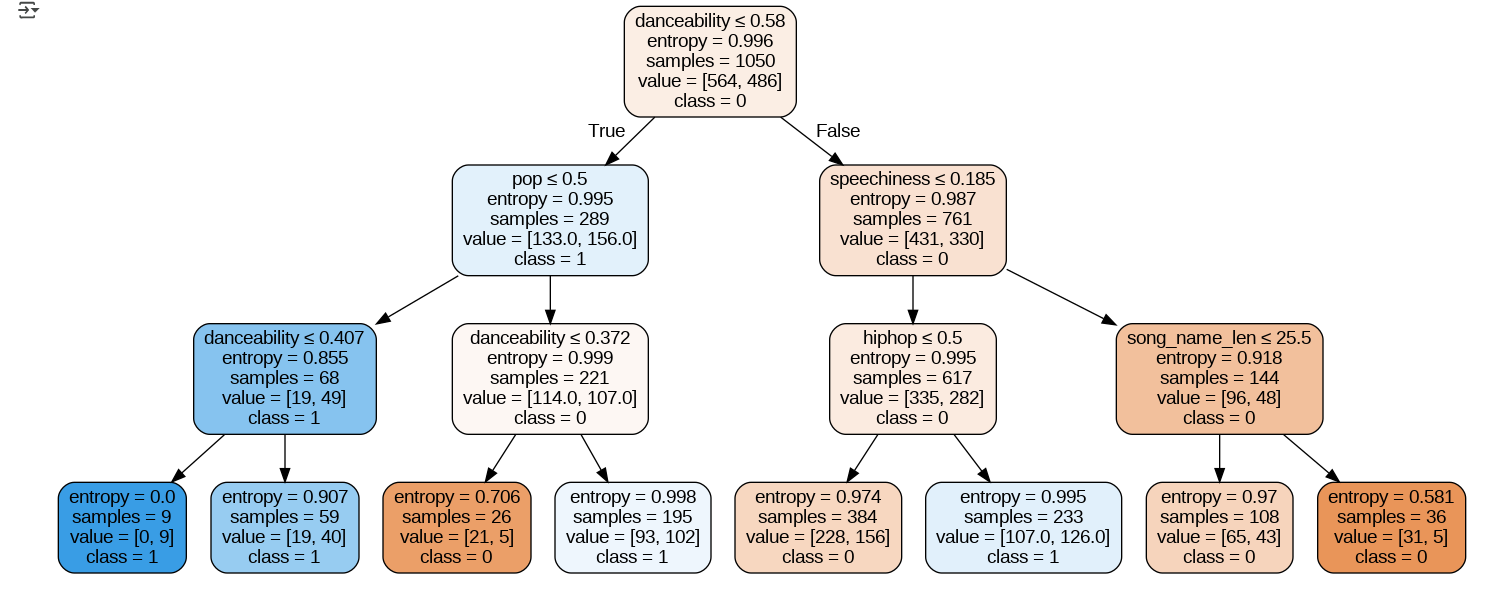
**SUMMARY OF RESULTS AND INSIGHTS FOR Q1**

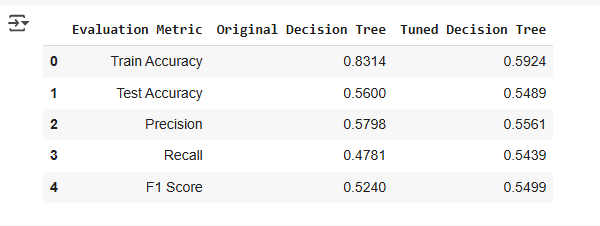
The primary objective was to predict whether a song would be considered popular based on its characteristics. A binary classification approach was adopted, and three different machine learning models were developed: **K-Nearest Neighbors (KNN), Decision Tree, and Logistic Regression.**

1. The **KNN** model, while simple to implement, displayed limited generalization capability with a test accuracy of approximately 58.6%. Cross-validation accuracy remained around 56.2%, indicating the model’s sensitivity to noisy and non-linear data patterns.

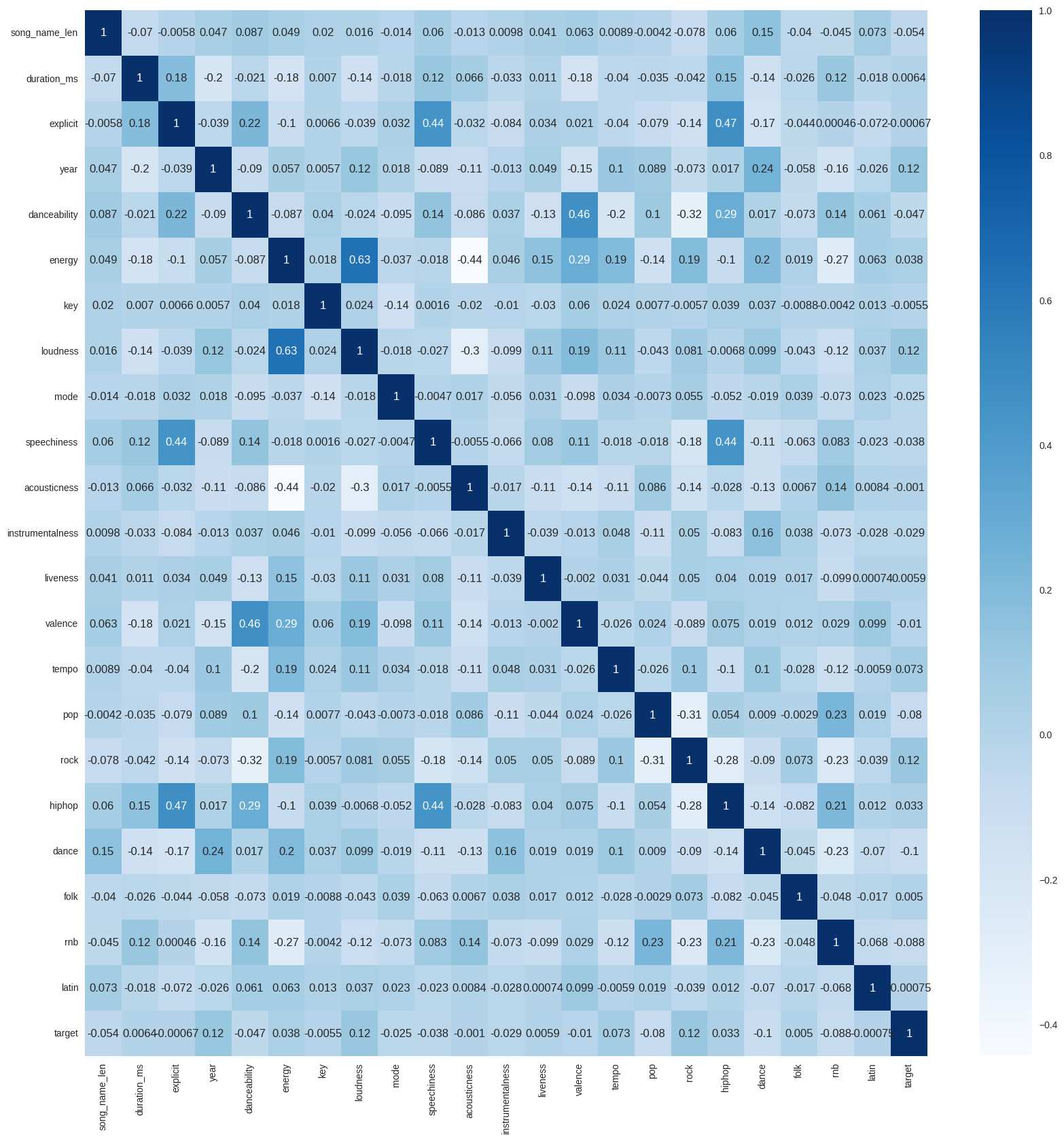


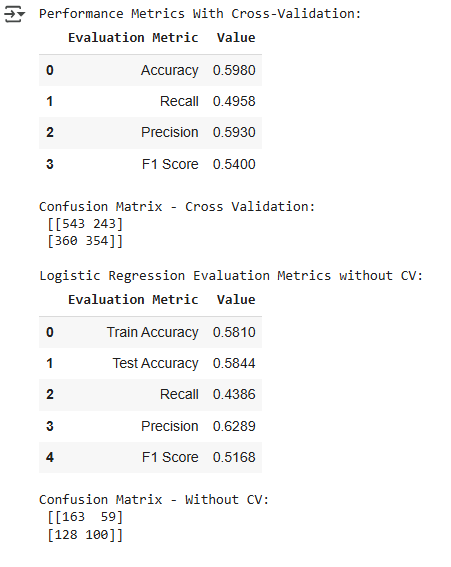
1. The **Decision Tree model** suffered from overfitting and yielded a test accuracy of about 56%. Logistic Regression, on the other hand, demonstrated the most consistent performance, achieving a test accuracy of approximately 59% and cross-validation accuracy close to 59.8%.





1. Based on overall accuracy and model interpretability, **Logistic Regression** was selected as the final model. It revealed that features such as danceability, tempo, energy, and loudness positively influenced a song’s likelihood of being popular. These findings highlight the importance of musical energy and rhythm in determining user engagement.





**Conclusion:**

Based on our model evaluation, the tuned **Logistic Regression** model achieved the highest test accuracy of 58.44%. It also produced the highest precision score of 0.6289, indicating strong performance in correctly identifying popular songs while minimizing false positives.

Although the **Decision Tree** has a slightly better F1 score, the overall accuracy metric positions **Logistic Regression** as the best model out of the three we tested on for Spotify’s recommendation system under this criterion.

**Managerial Insights:**

From a managerial standpoint, the results highlight specific features that are positively associated with higher song popularity.

In particular, danceability, tempo, energy, and loudness stood out as the most influential predictors based on the logistic regression model’s odds ratios.

Therefore, Spotify should focus on promoting songs that score higher on these attributes, as they are more likely to engage users and perform well on the platform.

These insights can guide Spotify’s playlist curation and recommendation strategies, ensuring a greater emphasis on energetic and rhythmically engaging tracks.

**SUMMARY OF RESULTS AND INSIGHTS FOR Q2**

To align predictive modeling with business goals, a profit-driven evaluation metric was applied using a confusion matrix-based cost structure. Each model was assessed based on its financial impact using the following values: a true positive yields $1,000 in revenue, a false positive incurs a $700 loss, a false negative results in a $900 loss, and a true negative has no financial implication.

The KNN model resulted in 123 true positives, 81 false positives, and 105 false negatives, yielding a net loss of $28,200. The Decision Tree model performed slightly worse, with a total loss of $38,900. Logistic Regression had the highest predictive accuracy but resulted in a greater financial loss of $52,000 due to higher false negatives.

Despite being less accurate, the KNN model was the most cost-effective option under the specified financial criteria. This analysis highlights the need to consider both predictive performance and economic implications when selecting models for deployment.

**Cost Benefit Analysis**

Revenue for correctly predicting a popular song: **1000**

Cost for incorrectly predicting a non-popular song as popular: **-700**

Cost for incorrectly predicting a popular song as non-populars: **-900**

1. **KNN**

From the confusion matrix:

A blue squares with numbers

AI-generated content may be incorrect.

TP = 123, FP = 81, FN = 105, TN = 141

Total Value = (123 × 1000) + (81 × -700) + (105 ×-900) = 123,000 – 56,700 – 94,500 = **–$28,200**

1. **Decision Tree**

From the confusion matrix:

A blue squares with numbers

AI-generated content may be incorrect.

TP = 124, FP = 99, FN = 104, TN = 123

Total Value = (124 × 1000) + (99 × -700) + (104 × -900) = 124,000 – 69,300 – 93,600 = **–$38,900**

1. **Logistic Regression**

**A blue squares with white text

AI-generated content may be incorrect.**

From the confusion matrix:

TP = 100, FP = 59, FN = 128, TN = 163

Total Value = (100 × 1000) + (59 × -700) + (123 × -900) = 100,000 – 41,300 – 1,10,700 = **–$52,000**

**Conclusion:**

The **K-Nearest Neighbors (KNN)** model yielded the **least monetary loss** at **–$28,200**, outperforming both the Decision Tree and Logistic Regression models in terms of cost-effectiveness. Although all models resulted in a net loss due to high false positive and false negative penalties, **KNN demonstrated the best trade-off** between correct and incorrect classifications when assessed financially.

**Managerial Insights:**

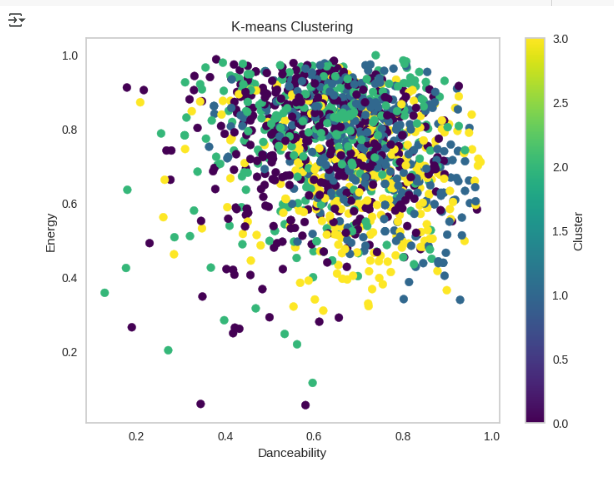
This analysis underscores the importance of aligning model selection not just with accuracy, but with the business objectives and financial outcomes.

While KNN may not have been the most accurate model in Q1, it is the **most economically viable** choice for Spotify's recommender system under the given cost structure.

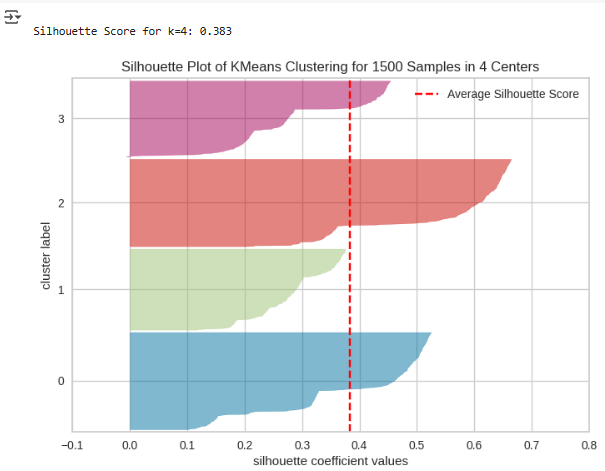
**SUMMARY OF RESULTS AND INSIGHTS FOR Q3**

To understand how musical/song characteristics influence popularity, particularly valence, and improve Spotify's playlist recommendations, we conducted an unsupervised clustering analysis using K-Means.

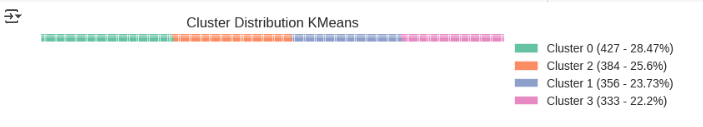
The clustering was based on key musical features: Danceability, energy, and optionally included genre indicators such as pop, hiphop, folk, rnb, and latin. This grouping allowed us to analyze patterns within similar types of music.

****

We used both the Elbow Method (WCSS) and Silhouette Score to determine the optimal number of clusters. The elbow curve suggested diminishing returns after k=4, while the silhouette analysis also supported this value, providing a moderate average score of 0.38.



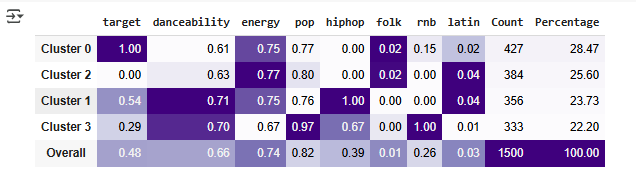
After confirming the optimal number of clusters, we assigned all songs to one of the four clusters and visualized them using a cluster scatter plot and a Waffle chart for distribution.



**Cluster Characteristics and Valence Impact**

Each cluster revealed unique musical and popularity patterns:

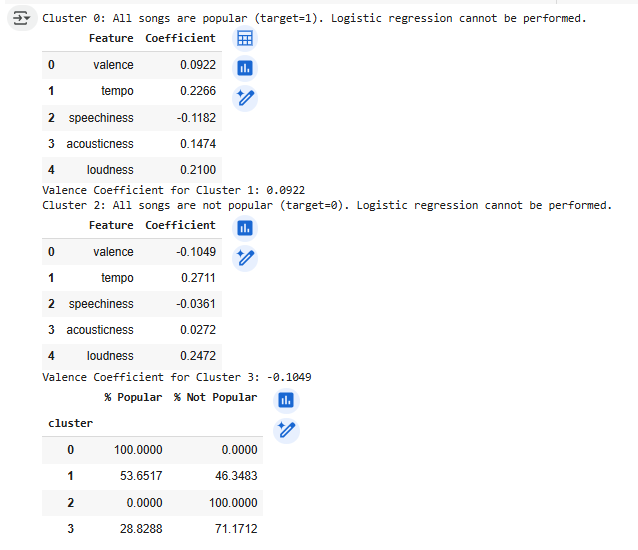
* **Cluster 0**: 100% of songs were popular, indicating strong overall appeal with high energy and danceability.
* **Cluster 2**: 0% popular — all songs were non-popular, suggesting a lack of market resonance.
* **Cluster 1**: 53.7% popular — this cluster had a mix of upbeat and emotional songs.
* **Cluster 3**: Only 28.8% popular — songs in this group were less favored by listeners.



To evaluate the role of valence (musical positivity), Logistic Regression models were trained within each cluster (excluding 0 and 2 due to label homogeneity). The results were:

* **Cluster 1**: Valence coefficient = **+0.0922**, indicating a **positive influence** on popularity.
* **Cluster 3**: Valence coefficient = **–0.1049**, indicating a **negative influence** on popularity.

This shows that valence does not have a uniform effect, it helps in some musical contexts and hurts in others.

****

Additionally, association rule mining was conducted to identify feature combinations frequently associated with popular or unpopular songs. Popular songs were often characterized by genre blends such as Pop and Hip Hop or Pop and Latin, clean lyrics, and the major musical key. Conversely, unpopular songs often included solo genre tracks and lacked strong branding or artist association.

**Features Associated with Popularity**

We also analyzed the average difference in features between popular and non-popular songs:

* Loudness, tempo, and duration were significantly higher in popular songs.
* Features like pop and rnb genres, danceability, and valence were slightly lower in popular tracks.
* These findings suggest that energy and intensity-related features are stronger indicators of success than mood or genre alone.



**Conclusion:**

These insights suggest that Spotify can improve its recommendation system by incorporating cluster-specific strategies. For instance, recommending high-valence, high-energy tracks in upbeat playlists or curating artist-based genre blends can enhance engagement. Songs with multi-genre elements and emotional positivity should be prioritized to meet varied user preferences and maximize playlist effectiveness.

**Managerial Insights**

* **Cluster-aware playlists:** Different user groups respond differently to song traits. Spotify should personalize playlists based on a song’s cluster.
* **Target energy-driven songs:** Loudness and tempo were consistently linked to popularity — promoting these songs can improve engagement.
* **Valence needs context:** It should not be used globally. In some clusters it enhances popularity (Cluster 1), while in others it reduces it (Cluster 3).
* **Avoid uniform models:** Feature effects vary by cluster, indicating that localized or hybrid models could improve recommendation accuracy.

|  |
| --- |
| **GOOGLE COLAB LINK:** [Spotify\_final.ipynb](https://colab.research.google.com/drive/1RgY-bOLN8cZTx2QIKrWpTdEqAAdAVXZ5?usp=sharing) |

**APPENDIX**

**1. Confusion Matrices for All Models**

**A blue squares with white text

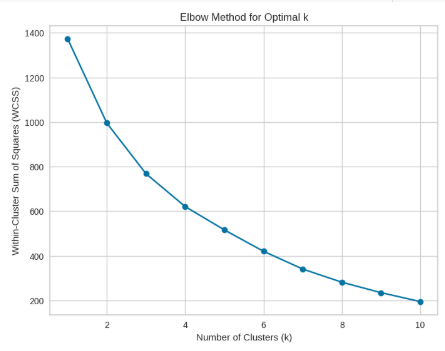
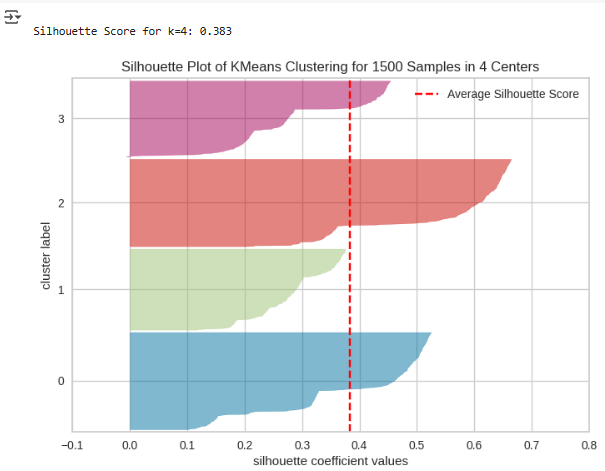
AI-generated content may be incorrect.**A blue squares with numbers

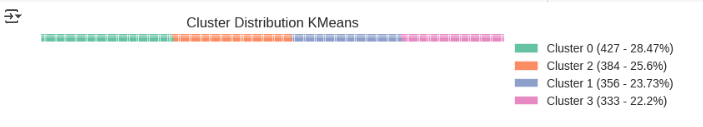
AI-generated content may be incorrect.

A blue squares with numbers

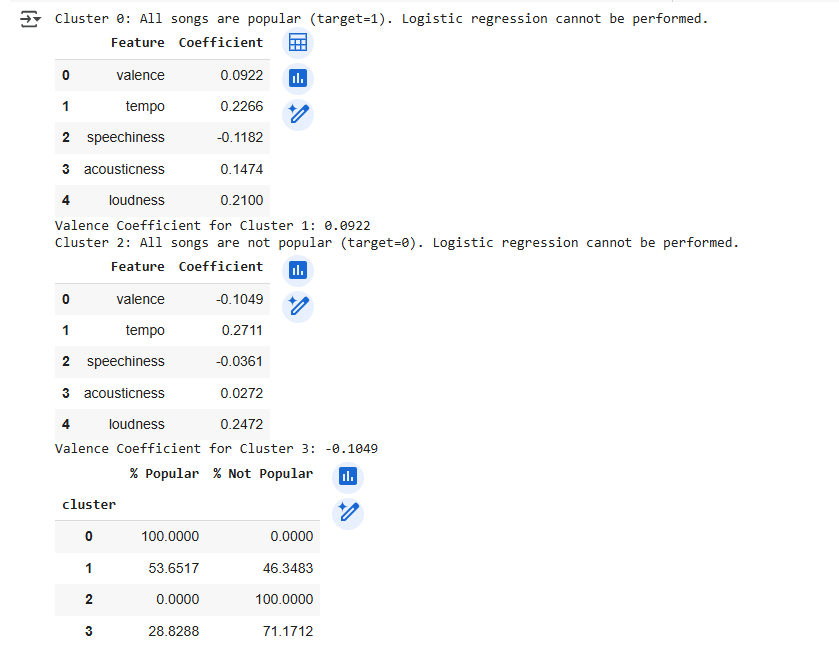
AI-generated content may be incorrect.

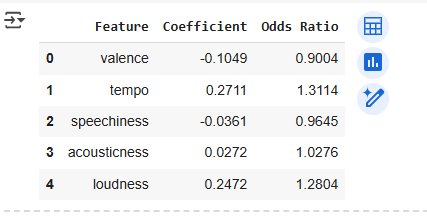
**2. K-Means Cluster Visualizations**

****

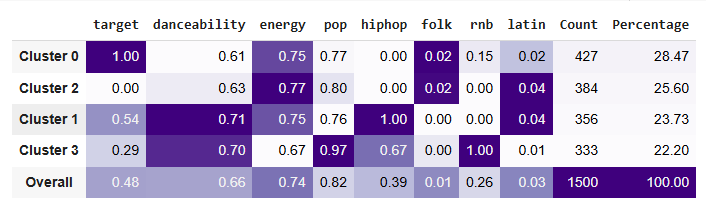


**3. Logistic Regression Coefficients and Odds Ratios**

****

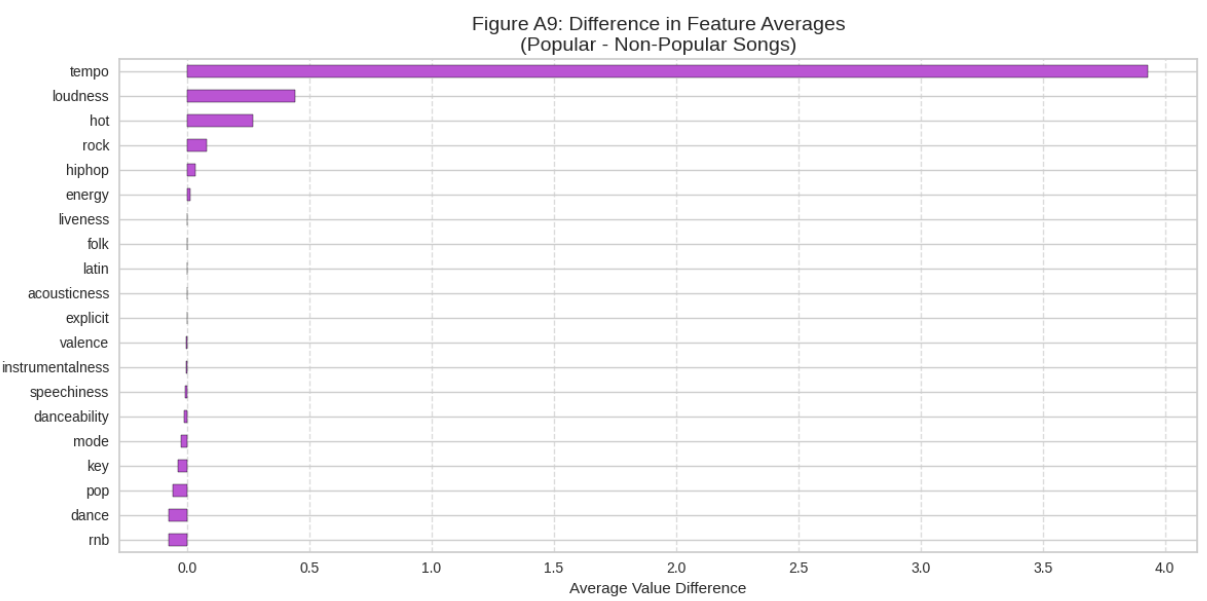
****

**4. Regression summary of velence impact per cluster**

****

**5. Feature importance table and bar chart**

****

****