VIRGINIA COMMONWEALTH UNIVERSITY

INFO 648 – BUSINESS DATA ANALYTICS

Team Project

Predicting and Enhancing Song Popularity on Spotify

Submitted by:

Abhigya Dwivedi

Rakshitha V Sargurunathan

Emily Rawls

Min Nguyen

Date of Submission: 05-06-2025

Table of Contents

Predicting and Enhancing Song Popularity on Spotify

Predicting and Enhancing Song Popularity on Spotify

1. Introduction

Spotify, as a leading music streaming service, is continuously enhancing its recommendation system to help users discover songs they love. Recommending the right songs not only drives user satisfaction and engagement but also directly impacts Spotify's retention and revenue.

The primary objective of this project is to assist Spotify in improving its recommendation strategy by predicting song popularity in advance — before or just as new songs are released.

To accomplish this, Spotify has hired our team to build predictive models using the dataset of historical songs, to answer the following key business questions:

- Predict whether a song will become popular (popularity score >= 64), so Spotify can recommend it to users more confidently.
- Choose the most suitable predictive model (Q1) that balances accuracy and reliability.
- Understand which model maximizes business profitability when correct/incorrect predictions are tied to revenue and costs (Q2).
- Analyze how song characteristics (especially valence, genre, danceability, and energy) cluster and how they impact song popularity to support better playlist recommendations (Q3).

By solving these, Spotify aims to create smarter recommendations, prioritize trending songs, improve playlist personalization, and ultimately increase user satisfaction and listening time.

2. About the Dataset

Spotify provided a dataset called songs_utf.csv which contains detailed information about top songs in the U.S. Spotify chart during 1998–2020.

Each record in the dataset captures various aspects of the song and its performance:

2.1 Dataset Features:

- **Artist & Song Metadata:**
    - artist — Name of the artist
    - song — Name of the track
    - year — Release year of the song
    - explicit — Whether the song has explicit content

- **Audio Features (Numerical):**
    - danceability — How suitable the song is for dancing (0 to 1 scale)
    - energy — Intensity and activity (0 to 1 scale)
    - loudness — Volume in decibels (continuous numerical)
    - valence — Musical positiveness (0 to 1 scale)
    - tempo — Beats per minute
    - instrumentalness, acousticness, speechiness, liveness, mode, key — Various technical attributes of the song's composition and recording.

- **Genre Information (Binary flags):**
    - Columns such as pop, rock, hiphop, dance, folk, rnb, latin indicate whether the song belongs to each genre (1 = yes, 0 = no).

- **Popularity Information (Target Variable):**
    - popularity — A numeric score assigned by Spotify (0–100) indicating how popular a song was on the platform at the time.

2.2 Target Variable:

For the purposes of this project:

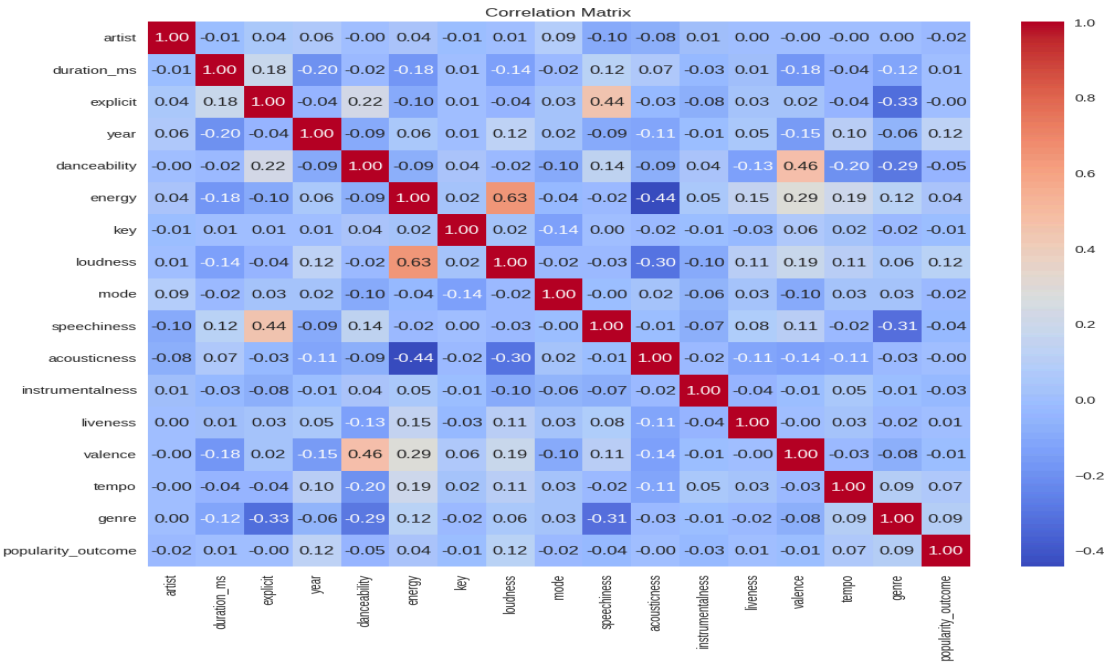
- A new variable "popularity_outcome" was created as:
    - 1 → Popular Song → if popularity >= 64
    - 0 → Not Popular Song → if popularity < 64

This binary outcome helps simplify the prediction task and aligns with Spotify's business objective of classifying songs as popular or not popular to aid in recommendations.

CORRELATION ANALYSIS:

Correlation analysis was conducted to understand the relationships among the various features within the dataset, as well as their individual relationships with the target variable, popularity_outcome.

The correlation matrix below provides a comprehensive view of how features relate to each other on a scale from -1 to +1



Several strong positive correlations were identified, notably between **Energy and Loudness (0.63)** and between **Danceability and Valence (0.46)**, suggesting that more energetic and happier songs are generally louder and more danceable. Similarly, strong negative correlations were observed, particularly between **Acousticness and Energy (-0.44)** and **Danceability and Acousticness (-0.29)**, reflecting that acoustic songs are often calmer and less rhythm-focused.

Additionally, genre and style-related associations were identified, such as the correlation between **Explicit content and Speechiness (0.44)**, which suggests that explicit songs tend to feature more spoken content, typical of rap or similar genres.

When analyzing the correlation of features with the target variable **popularity_outcome**, all features showed weak individual correlations. The highest observed was **duration_ms (0.12)**. This indicated that song popularity does not depend strongly on any single feature, and instead is likely influenced by complex patterns and interactions between multiple attributes.

Overall, the correlation analysis reinforced the necessity of utilizing machine learning models to capture these multi-dimensional relationships for predicting song popularity effectively.

3. Modelling Approaches

To tackle the problem of predicting song popularity, three different machine learning algorithms were selected and systematically applied to the prepared dataset. Each model brings its unique strengths and weaknesses, and our objective was to compare their performances to identify the most suitable solution for Spotify's recommendation pipeline.

3.1 Decision Tree Classifier

The Decision Tree model was selected as the first candidate due to its simplicity, ease of interpretability, and its ability to handle both numerical and categorical features natively.

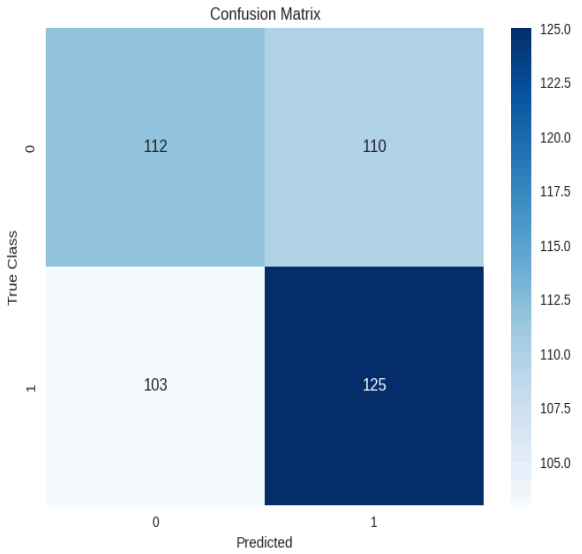
Model Building:

The initial Decision Tree model was trained using default parameters. The model was highly expressive and learned the patterns in the training data quickly. The results were as follows:

	Evaluation Metric	Value
0	Train Accuracy	0.8876
1	Test Accuracy	0.5267
2	Recall	0.5482
3	Precision	0.5319
4	F1 Score	0.5400

- A large gap (~36%) between training and testing accuracy indicated significant overfitting.
- The model captured popular songs effectively (high recall of 84.75%) but misclassified many non-popular songs as popular (low precision of 46.60%).
- F1 Score remained moderate, reflecting the imbalance.
- However, the low precision indicated that many songs it predicted as popular were actually not popular → this means **false positives were high**.

Confusion matrix:

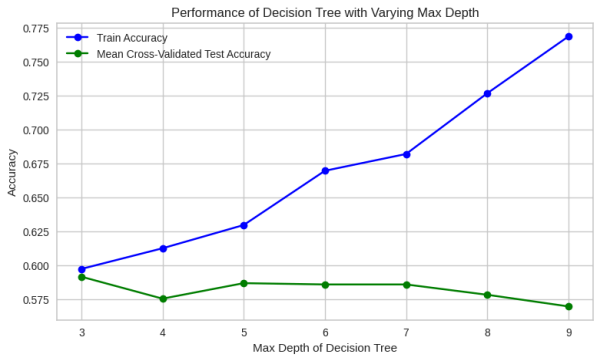


- The model demonstrates high error rates for both non-popular and popular classes.
- The number of False Negatives (103) is particularly concerning, as missing popular songs could result in lower user engagement and satisfaction.
- Additionally, 110 False Positives indicate the model tends to incorrectly push non-popular songs as popular, which affects recommendation relevance.

Hyperparameter Tuning

GridSearchCV was used to optimize the following parameters:

- max_depth → Controls tree growth.
- min_samples_split → Minimum samples required to split a node.



	max_depth	min_samples_split	splitter	mean_CV_test_score
0	3	5	best	0.5800
1	3	10	best	0.5800
2	3	20	best	0.5800
7	5	10	best	0.5724
6	5	5	best	0.5724
8	5	20	best	0.5724
3	4	5	best	0.5667
4	4	10	best	0.5667
5	4	20	best	0.5667

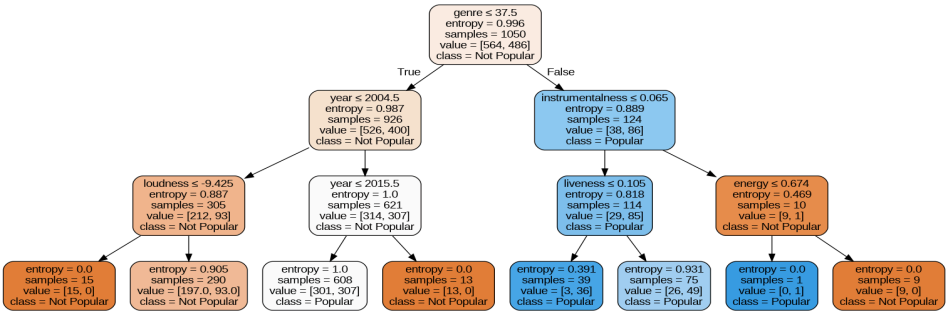
- Increasing max_depth improved training accuracy but decreased test accuracy, indicating overfitting.
- The optimal configuration was max_depth=3 and min_samples_split=5.

Final Tuned Model Performance:

	Evaluation Metric	Original Decision Tree	Tuned Decision Tree
0	Train Accuracy	0.8876	0.5971
1	Test Accuracy	0.5267	0.5778
2	Precision	0.5319	0.5609
3	Recall	0.5482	0.7675
4	F1 Score	0.5400	0.6481

- Overfitting reduced (Train accuracy dropped to 59.71%).
- Test accuracy improved to 57.78%, but remained moderate.

- Recall remained strong (76.75%), still favoring popular song identification.
- Precision improved slightly, reducing false positives.
- F1 Score increased to 64.81%, reflecting better balance.



- The Decision Tree after tuning was balanced but did not outperform other models in accuracy.
- Its strength remained in feature importance and interpretability.

3.2 K-Nearest Neighbors (KNN)

The second algorithm explored was K-Nearest Neighbors (KNN), a model that predicts the class of a sample based on the classes of its closest neighbors. KNN is non-parametric and does not make assumptions about the underlying data distribution, making it a good choice for exploratory modeling.

Model Building:

Initially, KNN was applied with a default n_neighbors value of 3.

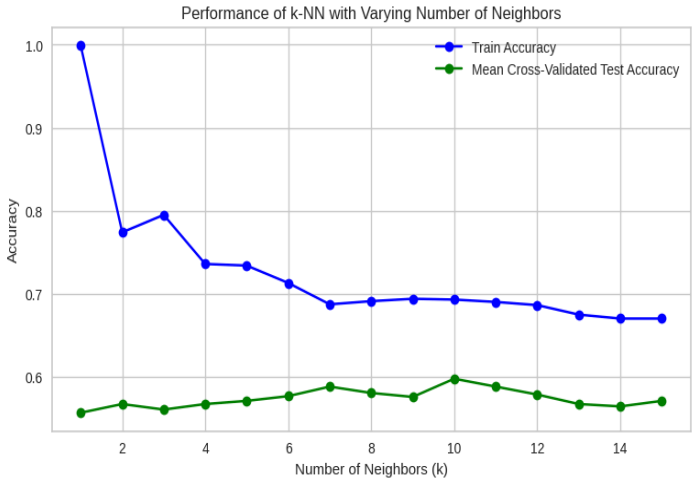
	Evaluation Metric	Value
0	Train Accuracy	0.7952
1	Test Accuracy	0.5378
2	Recall	0.4868
3	Precision	0.5495
4	F1 Score	0.5163

- The model showed excellent performance on the training data (**Train Accuracy: 99.90%**), but very poor generalization to the test set (**Test Accuracy: 53.33%**), indicating strong overfitting.
- It performed reasonably well in recalling popular songs (**Recall: 50.23%**), but:
- Produced many false positives — incorrectly marking non-popular songs as popular (**Precision: 58.06%**).
- This imbalance hinted that the model was **overfitting at low K values** (i.e. very few neighbors).

Hyperparameter Tuning

- n_neighbors → The number of neighbors to consider during classification.

To find the optimal value of K, the parameter was systematically increased and performance was evaluated using cross-validation:



	metric	n_neighbors	weights	mean_CV_test_score
5	euclidean	10	distance	0.5981
4	euclidean	10	uniform	0.5981
7	euclidean	11	distance	0.5971
1	euclidean	7	distance	0.5952
9	euclidean	12	distance	0.5914
3	euclidean	8	distance	0.5905
6	euclidean	11	uniform	0.5886
0	euclidean	7	uniform	0.5886
2	euclidean	8	uniform	0.5810
11	euclidean	13	distance	0.5790
8	euclidean	12	uniform	0.5790
10	euclidean	13	uniform	0.5676

- At very low K values (e.g. 1–3), the model was extremely biased towards the training data (train accuracy remained very high → ~99%).
- As K increased, test accuracy steadily improved and the gap between training and test accuracy reduced.
- At K=10, test accuracy stabilized around 57.33%, offering the best trade-off.

Post-Tuning Observations:

- Test Accuracy improved to 57.33%, but still lagged behind Decision Tree and Logistic Regression.
- Recall slightly improved to 54.39%, but was still lower than Decision Tree.
- Precision increased slightly to 58.49%, balancing false positives better.
- The model faced challenges in handling complex, high-dimensional song data, and remained sensitive to imbalanced datasets.

	Evaluation Metric	Original k-NN	Tuned k-NN
0	Train Accuracy	0.7952	0.9990
1	Test Accuracy	0.5378	0.5733
2	Precision	0.5495	0.5849
3	Recall	0.4868	0.5439
4	F1 Score	0.5163	0.5636

- Test Accuracy improved to 57.33%, but still lagged behind Decision Tree and Logistic Regression.
- Recall slightly improved to 54.39%, but was still lower than Decision Tree.
- Precision increased slightly to 58.49%, balancing false positives better.
- The model faced challenges in handling complex, high-dimensional song data, and remained sensitive to imbalanced datasets.

3.3 Logistic Regression

Logistic Regression was selected for its ability to produce interpretable models and its suitability for binary classification problems. The model requires independent features, which necessitated pre-processing steps before modeling.

Multicollinearity Check

A Variance Inflation Factor (VIF) analysis was conducted to assess multicollinearity among independent variables. Three variables were found to have high VIF values and were removed.

- Danceability- 27.34
- Energy-25.91
- Tempo- 19.80

Danceability, Energy, and Tempo were excluded from the model.

This adjustment improved model stability and mitigated multicollinearity.

	feature	VIF
0	artist	3.8445
1	danceability	27.3422
2	energy	25.9084
3	loudness	9.8316
4	dance	1.3365
5	rock	1.5274
6	hiphop	1.9316
7	pop	6.0379
8	valence	11.2967
9	instrumentalness	1.0992
10	acousticness	1.7644
11	tempo	19.7989

	feature	VIF
0	artist	3.5573
1	loudness	6.8132
2	dance	1.2606
3	rock	1.2952
4	hiphop	1.7764
5	pop	5.1596
6	valence	5.3442
7	instrumentalness	1.0956
8	acousticness	1.7365

```
<ipython-input-105-a20706bf9843
A value is trying to be set on
Try using .loc[row_indexer,col_
```

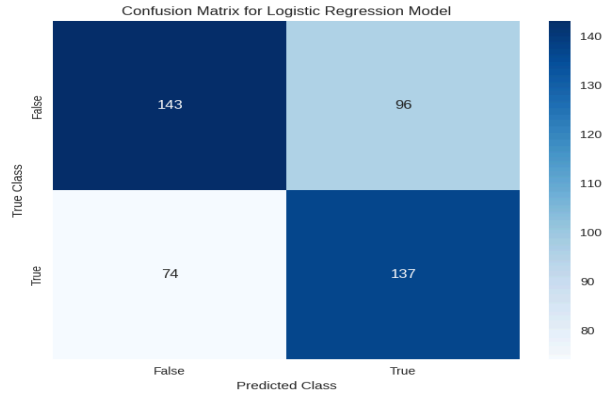
Initial Model Performance

The Logistic Regression model was then fitted to the processed data. The performance metrics are as follows:

	Evaluation Metric	Value
0	Train Accuracy	0.8429
1	Test Accuracy	0.6222
2	Recall	0.6493
3	Precision	0.5880
4	F1 Score	0.6171

- The difference between training and test accuracy was minimal, suggesting limited overfitting.
- The recall of 64.93% indicated that the model was able to correctly identify a significant proportion of popular songs.
- Precision at 56.19% suggested moderate success in avoiding false positives.
- The F1 Score of 57.61% reflected an overall balanced performance.

Confusion Matrix:



- True Positives (Popular songs correctly classified): 137
- True Negatives (Non-popular songs correctly classified): 143
- False Positives (Non-popular songs predicted as popular): 96
- False Negatives (Popular songs predicted as non-popular): 74

Cross-Validation Performance

A 10-Fold Cross-Validation procedure was conducted to validate model stability.

	Evaluation Metric	Value
0	Accuracy	0.5860
1	Recall	0.5910
2	Precision	0.5619
3	F1 Score	0.5761

- Accuracy remained relatively consistent, indicating the model's reliability across different splits of the dataset.
- Recall and Precision remained aligned with initial results, confirming model robustness.

3.4 Model Comparison:

Following the development and evaluation of all three models — Decision Tree, K-Nearest Neighbors (KNN) and Logistic Regression — a comparative analysis was conducted to determine the most suitable model for predicting song popularity.

Model	Test Accuracy	Recall	Precision	F1 Score
Decision Tree	0.5778	0.7675	0.5609	0.6481
K-Nearest Neighbors	0.5733	0.5439	0.5849	0.5636
Logistic Regression	0.5860	0.5910	0.5619	0.5761

Test Accuracy:
Logistic Regression recorded the highest test accuracy of **0.5860**, outperforming both Decision Tree (**0.5778**) and KNN (**0.5733**). Test accuracy is critical as it reflects the model's ability to generalize to unseen data.

Recall:
The Decision Tree achieved the highest recall of **0.7675**, indicating strong performance in correctly identifying popular songs. Logistic Regression recorded a moderate recall of **0.5910**, while KNN had the lowest recall at **0.5439**.

Precision:
KNN demonstrated the highest precision at **0.5849**, which means it made fewer false positive predictions when classifying songs as popular. Logistic Regression followed with **0.5619**, while Decision Tree recorded **0.5609**.

F1 Score:
Decision Tree achieved the best F1 Score of **0.6481**, showing a balanced performance between precision and recall. Logistic Regression achieved **0.5761**, and KNN recorded **0.5636**.

Final Model Selection Rationale

Although the Decision Tree exhibited strong recall and F1 score, its lower accuracy and higher variance between precision and recall highlight concerns regarding stability and generalization.

KNN, while demonstrating the highest precision, suffered from low recall and slightly lower overall test accuracy. Additionally, KNN's sensitivity to the dataset's imbalance limits its practical use in this context.

Logistic Regression, on the other hand, offered the highest test accuracy and a balanced performance across all metrics:

- Highest test accuracy (0.5860) ensures generalization to unseen songs.
- Balanced recall and precision help avoid extreme false positives or false negatives.
- Consistent performance in cross-validation (58.60%) demonstrates stability across different data splits.
- Interpretability of coefficients makes the model valuable for business applications and playlist curation strategies.

Based on a combination of accuracy, balance across evaluation metrics, stability, and interpretability, Logistic Regression was selected as the final model.
This model is best suited to support Spotify's recommendation system by effectively predicting popular songs while maintaining explainability for decision-making and marketing strategies.

4. Model Performance on Profitability

The company aims to maximize their profits while also utilizing an accurate model. In the above section, we reviewed and compared the models purely on performance and accuracy in identifying popular songs. Next, we need to review how those models will perform when considering the overarching effect on the company's P&L.

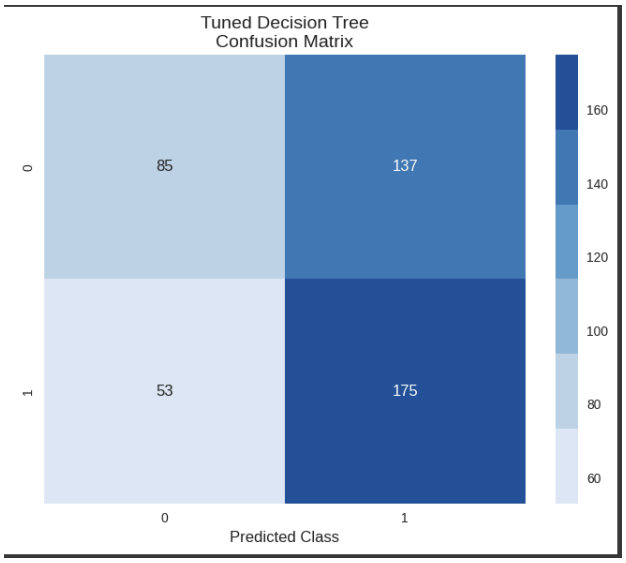
4.1 Breaking Down the Impact of Model Performance on the P&L

The company was able to provide us with a complete overview of how True Positives, False Positives, False Negatives, and True Negatives impact the P&L. When a model accurately predicts a popular song (true positive), we can expect $1,000 added in revenue. When a model falsely predicts a song is popular when it in fact is not (false positive), we can expect it to cost the business $700, and when the model falsely predicts that a song is not popular when it in fact is (false negative), we can expect it to cost the business $900. There is no impact to the P&L when the model accurately predicts a song is not popular.

To choose the best model for profitability, we needed to create a formula that would help us identify the best approach. Our calculation was as follows: Profit = (# of True Positive * $1,000) - (# of False Positives * $700) - (# of False Negatives * $900).

4.2 Decision Tree Accuracy and Profitability

To start, we will want to look at the confusion matrix for the final tuned decision tree model we created in the last section.



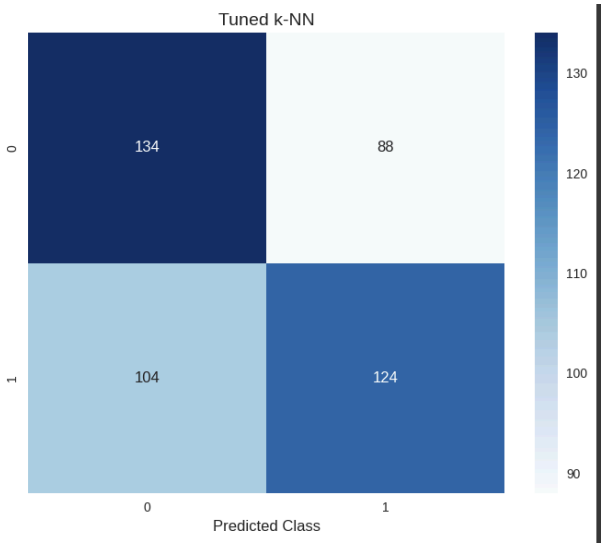
If we apply the formula we created above we will see the following:

Profit = (175 * $1,000) - (137 * $700) - (53 * $900)

Profit = $31,400

4.3 KNN Accuracy and Profitability

Similar to Decision Tree, we will look at the confusion matrix for our tuned KNN Model.



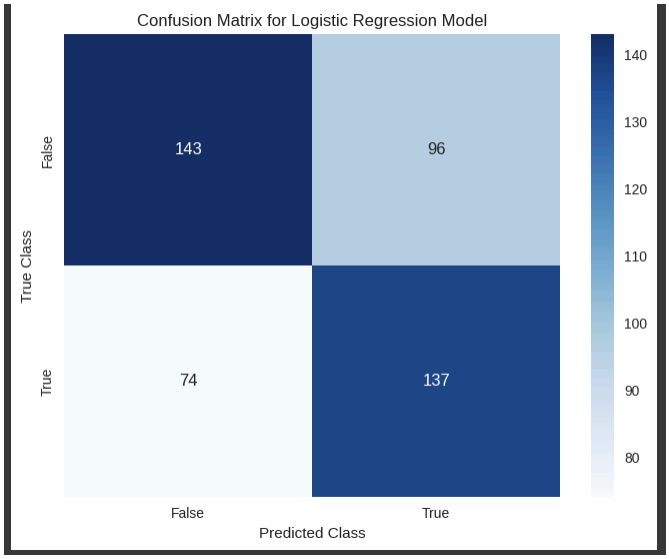
If we apply our formula:

Profit = (124 *$1,000) - (88 * $700) - (104 *$900)

Profit = -$31,200

4.4 Logistic Regression Accuracy and Profitability

Like the above, we will use the logistic regression confusion matrix to review profitability.



If we apply our formula:

Profit = (137 * $1,000) - (96 * $700) - (74 * $900)

Profit = $3,200

4.5 Conclusion and Recommendations

The above highlights that while some models may have a lower overall accuracy than others, profitability really depends on how many true positives the model gets vs how many false negatives. These two metrics have the highest positive and negative impact on the company's profitability.

In our first section, we made the conclusion that Logistic Regression would be the best predictive model based on accuracy and balance. While that is accurate, what we see from Logistic Regression, is that the model relatively accurately predicts when a song is truly popular as well as when it is truly not popular. As we know, when the model accurately predicts a song is popular (true positive) we can expect that there will be $1,000 added in revenue, there is no benefit to accurately predicting a song is not popular.

When we look at Logistic Regression compared to the tuned Decision Tree and tuned KNN we will see that while its overall accuracy is higher, when it comes to profitability its larger number of false negatives impacts the overall performance for the business.

Looking at all three models, we can come to the conclusion that the tuned Decision Tree would be the best overall model that meets the company's needs. While it did not predict true negatives as well as the other models, it did predict a higher number of true positives and had the least amount of false negatives. As we mentioned before, these two metrics can really make or break the decision on which model to utilize in the future. We see this also being true by the fact that the Decision Tree had the largest number of false positives, yet still was more profitable than the KNN and Logistic Regression.

Our recommendation to management would be to implement the tuned Decision Tree to maximize both accuracy and profitability moving forward.

5. Impact of Valence and Feature Combinations on Song Popularity: Insights for Playlist Optimization

The company aims to enhance playlist recommendations by identifying features and combinations of features that frequently associate with song popularity. This analysis will help predict song success based on the valence of the song and the interplay between other features, while also considering how these effects differ across music genres. The clustering and association rule mining methods were employed to provide insights into these relationships.

5.1 Clustering Songs by Danceability and Energy

For clustering the songs, K-means clustering was used, and the optimal number of clusters was determined to be 3, based on the Elbow Method and Silhouette Score. The following characteristics were observed in each cluster:

Cluster	Danceability	Energy	Count	Percentage
Cluster 1	0.76	0.77	670	44.67%
Cluster 0	0.53	0.83	509	33.93%
Cluster 2	0.64	0.52	321	21.40%

Overall	0.66	0.74	1500	100%

Table 4.1 Clustering

- **Cluster 0 (33.9%)**: This cluster includes songs with low danceability (0.53) but high energy (0.83). These songs likely represent genres such as rock or electronic, which have a high tempo but less rhythmic swing. Despite the lower danceability, this group showed the highest success rate, with 14.5% of songs classified as hot.

- **Cluster 1 (44.67%)**: Songs in this cluster are marked by high danceability (0.76) and high energy (0.77), typical of mainstream pop and dance genres. These tracks are suitable for party or workout playlists but had a slightly lower hot song ratio (12.5%) than Cluster 0.

- **Cluster 2 (21.4%)**: Featuring moderate danceability (0.64) and lower energy (0.52), this group is composed of songs suited for chill or acoustic playlists. These songs showed the lowest hot song ratio (10.6%).

5.2 Impact of Valence on Song Success

Cluster	Average valance	Hot ratio	Count
Cluster 0	0.52	14.54%	509
Cluster 1	0.67	12.54%	670
Cluster 2	0.45	10.59%	321

Table 4.2 Valance impact

The average valence and hot song ratio were analyzed within each cluster to evaluate how valence impacts the song's success:

- **Cluster 0:** Despite having a relatively low average valence (0.52), this cluster had the highest hot song ratio (14.54%), indicating that high energy, even with moderately positive emotional

tones, contributes to the popularity of songs.

- **Cluster 1:** With the highest average valence (0.67), this cluster still had a lower hot song ratio (12.54%), suggesting that while valence contributes to the song's appeal, other factors like novelty, genre, and lyrical content are also crucial.

- **Cluster 2:** This group showed the lowest average valence (0.45) and had the lowest hot song ratio (10.59%), suggesting that lower energy and a less positive emotional tone reduce the likelihood of success.

5.3 Identifying Popularity-Related Feature Combinations

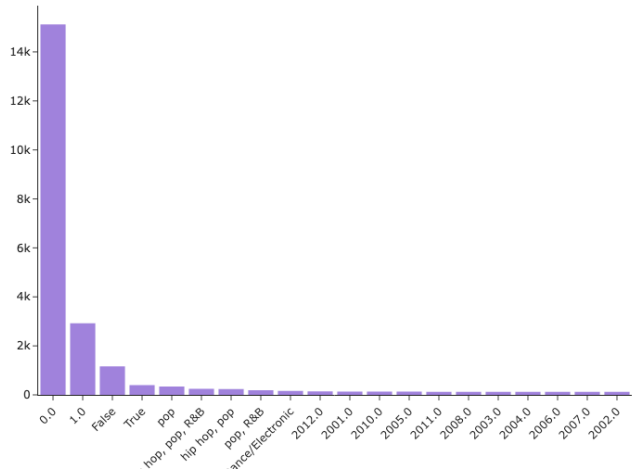
To identify feature combinations that correlate with song popularity, association rule mining was conducted. The top 20 frequent items and association rules were analyzed:

- The **Top 20 Items by Frequency** revealed that binary features such as "True" and "False" (indicating explicit content or popularity) are significant factors in engagement. Additionally, **pop** and **cross-genre combinations** like pop + hip hop or pop + Dance/Electronic were frequently seen, supporting the idea that genre blending enhances popularity.

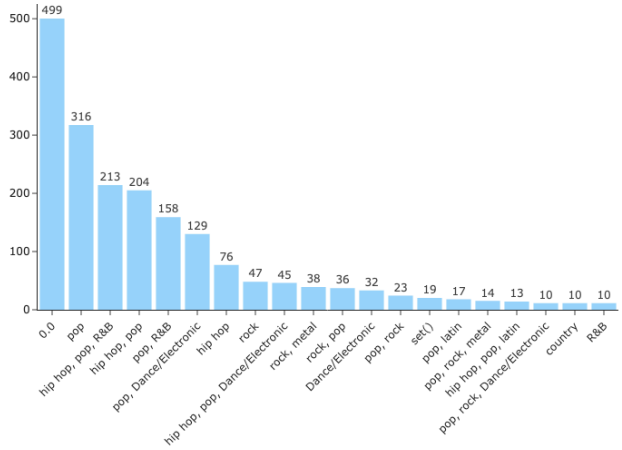
- **First Choice Items by Customers** also highlighted the dominance of pop-related genres, with popular combinations including **hip hop + pop** and **Dance/Electronic + pop**. These trends suggest that genre blending significantly impacts initial engagement.

- **Association Rules Ranked by Lift** emphasized the strong association between upbeat, expressive songs (with high valence and explicit content) and success. The highest lift values (>13) frequently involved **pop + Dance/Electronic** songs released around 2012-2013, confirming the role of genre, mood, and explicit content in driving popularity.

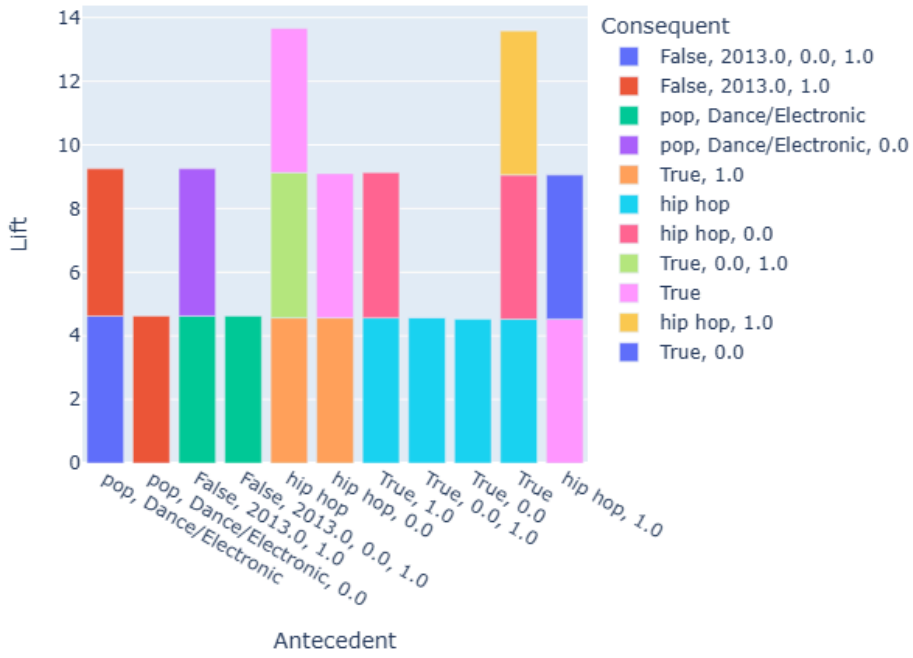
Top 20 Items by Frequency (Support Count)



Top 20 First Choice Items by Customers



Top 20 Association Rules by Lift



5.4 Managerial Insights

The clustering analysis revealed that songs in Cluster 0, with high energy and moderate valence, were most likely to be successful. Although Cluster 1 featured upbeat and positive songs, it did not outperform Cluster 0, indicating that other factors beyond valence, such as genre and lyrical content are key to a song's success. This insight suggests that playlist strategies should focus not only on energy levels but also on the emotional tone and context of the songs, adjusting playlists to match different listening environments like workouts or relaxation.

The association rule mining results underscore the importance of cross-genre blending and mood-enhancing features like high valence and explicit content. Songs that combine pop with hip hop, Dance/Electronic, and R&B are particularly successful, and these combinations should be prioritized in the company’s playlist recommendations. Additionally, the frequent appearance of binary features such as explicit labels and popularity indicators suggests that these factors should guide the configuration of dynamic and personalized playlists.

Based on these findings, the company should refine its playlist strategies by incorporating genre-blending and mood-enhancing features, particularly those related to valence and explicit content. These features are strongly associated with higher popularity, and their inclusion in playlists could increase user satisfaction and engagement.

In conclusion, by leveraging clustering and association rule mining, the company can fine-tune its playlist recommendations to align with listener preferences, particularly by focusing on emotional nuance, genre combinations, and the strategic use of explicit content to maximize engagement.

Appendix

Collab Link:
https://colab.research.google.com/drive/1aaw4tLKMnod0uGA3qYsO5YFvYfeWOf7o?usp=sharing