Team Members:

Andrew Rafael James, Doris Liang, Hunter Guo, Monica Ko, Pang Leesuravanich

Executive Summary

This white paper investigates the drivers of song popularity on Spotify by analyzing a dataset of 10,000 top songs spanning from 1950 to the present. Leveraging exploratory data analysis and K-Means clustering, the study identifies significant patterns in audio features and song similarities, grouping tracks into meaningful categories. Classification models, such as Random Forest and LightGBM, are employed to predict song popularity, achieving moderate success despite limitations posed by the absence of user interaction data. To enhance music discovery, a sophisticated recommendation system is developed using content-based filtering and cosine similarity, integrating audio features, sentiment analysis, lyrics, and artist information to deliver personalized song suggestions. These innovations aim to boost user engagement, refine playlist curation, and provide actionable market insights for artists and labels. By harnessing data analytics, this research offers a transformative framework for improving music streaming services and supporting industry stakeholders.

Problem Statement

With the rise of music streaming, platforms like Spotify, with over 600 million users, have reshaped how people discover music, making it crucial to understand what makes a song popular. By analyzing Spotify's top 10,000 songs from 1950 to now, this study explores how audio features like tempo, energy, and danceability influence popularity and whether trends have shifted across eras. Additionally, song analysis provides a deeper understanding of song features to create song categories and a recommendation engine that enhances the customer experience. Therefore, the final objective is to **recommend similar songs based on the user's favorite song features**, and we will achieve it by completing two milestones: (1) create various song categories for different user tastes (2) predict hit songs to increase platform engagement

These insights can enhance playlist creation, improve recommendations, and help artists and labels understand evolving music trends.

Data Source

Our dataset is sourced from Kaggle, which contains a list of 10,000 popular songs that have dominated ARIA and Billboard charts from 1950 until present. The dataset contains information about songs, genre, artist, album names, as well as, audio-related features, ranging from level of loudness and danceability to level of acoustic and instrumental.

1. Classifying Song Popularity
    a. we will use *Popularity* column as our target variable and use *song and artist's informations*, as well as, *audio-related features* to predict song popularity
2. Clustering Song Similarity

a. using *audio-related features* allows us to understand similar songs that can be clustered together for playlist creation

3. Recommending Songs to Users

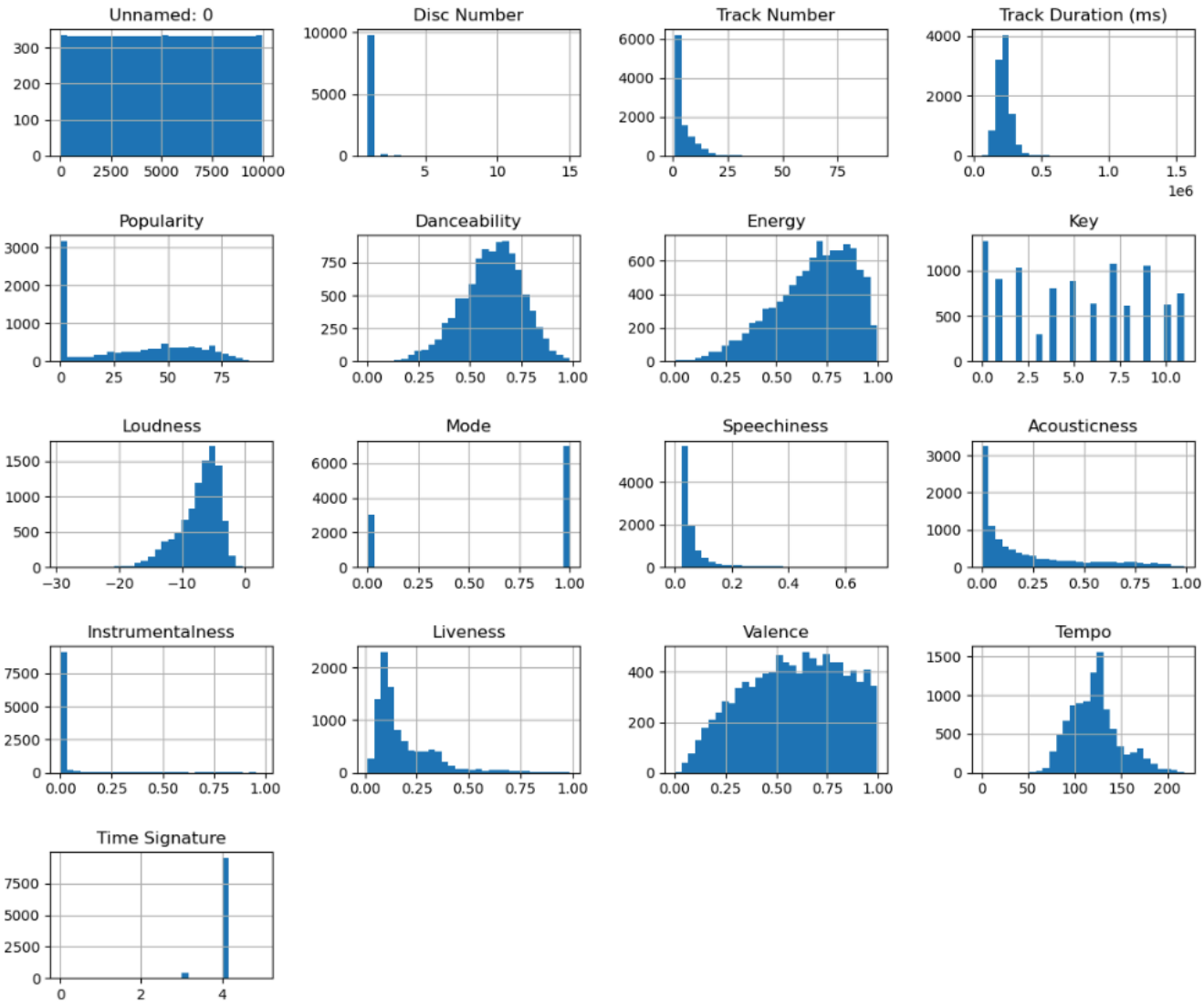
No	Column Name	Column Description	Column Description
1	Track.URI	character	Song URL
2	Track.Name	character	Song Name
3	Artist.URI.s.	character	Artist URL
4	Artist.Name.s.	character	Artist Name
5	Album.URI	character	Album URL
6	Album.Name	character	Album name
7	Album.Artist.URI.s.	character	Album Artist URL
8	Album.Artist.Name.s.	character	Album Artist Name
9	Album.Release.Date	date	Album Release Date
10	Album.Image.URL	character	Album Image URL
11	Disc.Number	double	Disc Number
12	Track.Number	double	Track Number
13	Track.Duration..ms.	double	Track Duration in milliseconds
14	Track.Preview.URL	character	Track Preview URL
15	Explicit	boolean	Whether or not the track has explicit lyrics (true = yes it does; false = no it does not OR unknown)
16	Popularity	character	The popularity of a track is a value between 0 and 100, with 100 being the most popular. The popularity is calculated by algorithm and is based, in the most part, on the total number of plays the track has had and how recent those plays are. Generally speaking, songs that are being played a lot now will have a higher popularity than songs that were played a lot in the past. Duplicate tracks (e.g. the same track from a single and an album) are rated independently. Artist and album popularity is derived mathematically from track popularity.
17	ISRC	character	International Standard Recording Code. It's a 12-digit code that uniquely identifies a sound recording or music video. ISRCs are used to manage rights, track sales, and collect royalties
18	Artist.Genres	character	Artist genre

19	Danceability	double	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
20	Energy	double	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
21	Key	double	The estimated overall key of the track. Integers map to pitches using standard Pitch Class notation . E.g. 0 = C, 1 = C♯/D ♭ , 2 = D, and so on. If no key was detected, the value is -1.
22	Loudness	double	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typical range between -60 and 0 db.
23	Mode	double	Mode indicates the modality (major or minor) of a track, the type of scale from which its melodic content is derived. Major is represented by 1 and minor is 0.
24	Speechiness	double	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
25	Acousticness	double	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.

26	Instrumentalness	double	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
27	Liveness	double	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.
28	Valence	double	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).
29	Tempo	double	The overall estimated tempo of a track in beats per minute (BPM). In musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration.
30	Time.Signature		An estimated time signature. The time signature (meter) is a notational convention to specify how many beats are in each bar (or measure). The time signature ranges from 3 to 7 indicating time signatures of 3/4, to 7/4
31	Album.Genres	character	Album Genre
32	Label	character	recording label
33	Copyrights	character	copyrights
34	Lyrics	character	Song lyrics

EDA (with Clustering)

1. **Histograms - Check Feature Distributions**



- Most songs are in the 1st disc according to Disc Number, which makes sense because most albums have only 1 disc, only few are multi-disc albums

- Most songs are the 1st song of their belonging album, and most albums have less than 25 songs according to Track Number

- Most songs are between 200,000 ms and 300,000 ms, which are between 200s and 300s, or between 3.5min and 5min according to Track Duration (ms)

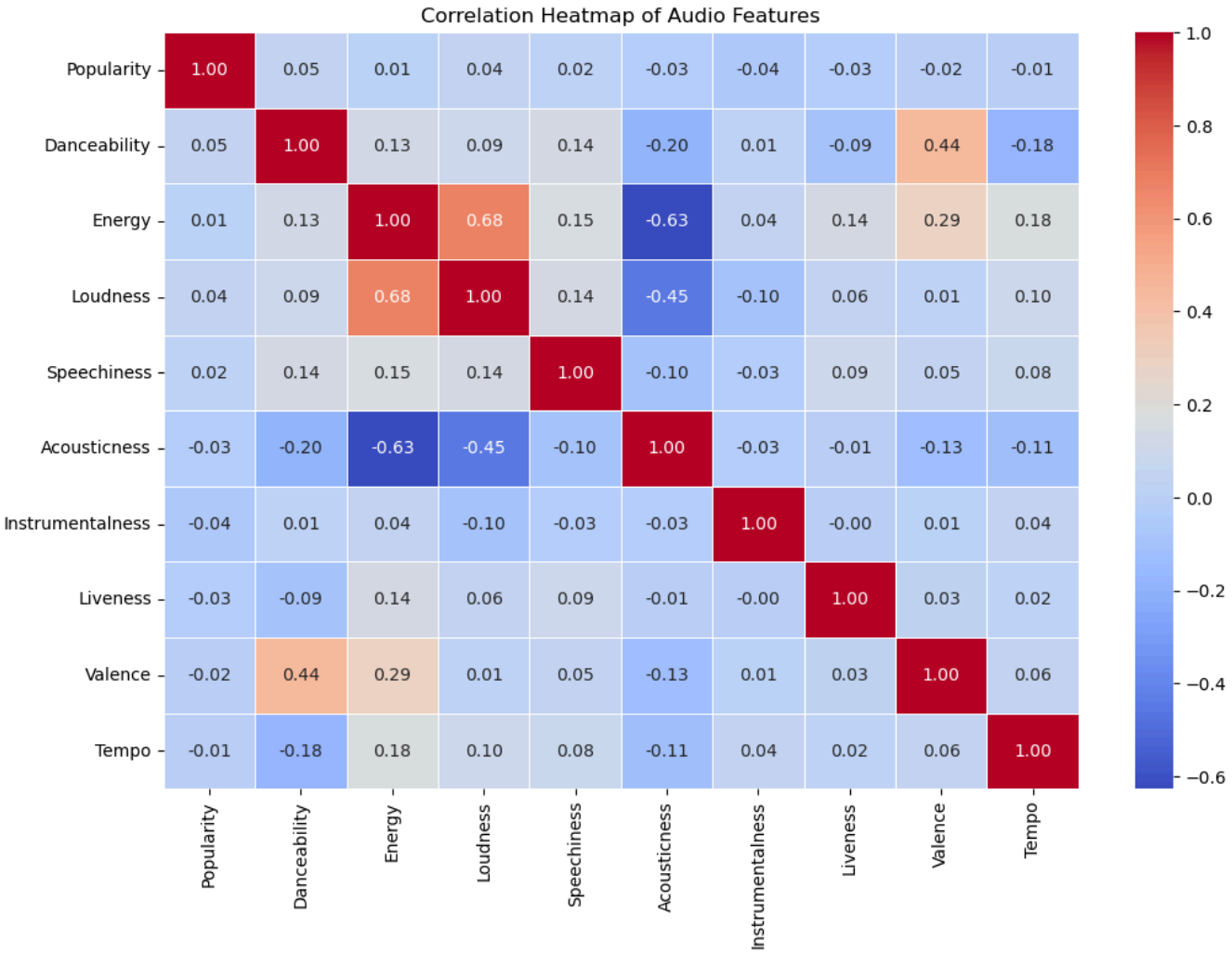
- Most songs have popular score close to 0, which means most songs in this dataset are not popular according to Popularity

- According to left-skewed distributions of Danceability, Energy, Loudness, Valence, Tempo, most songs are exciting and positive

- According to right-skewed distributions of Speechiness, Acousticness, Instrumentalness, Liveness, most songs are vocal (less rap, less pure music) and studio recordings (not live music)

- According to Key (evenly distributed), Mode (most 1), Time Signature (most 4/4), most songs have a wide variety of major-keys and are modern pop music

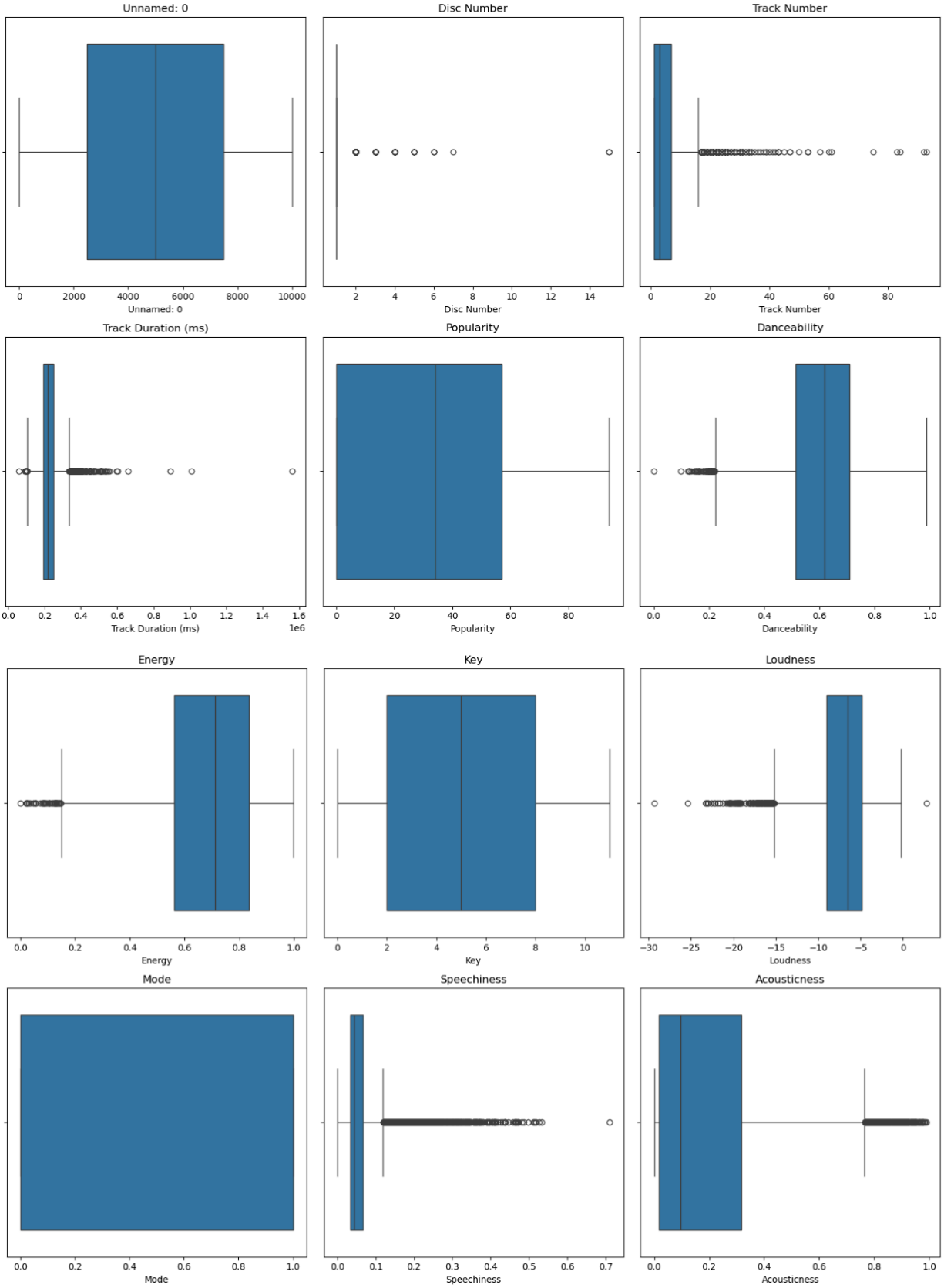
2. Correlation Matrix - Check Feature Relationships

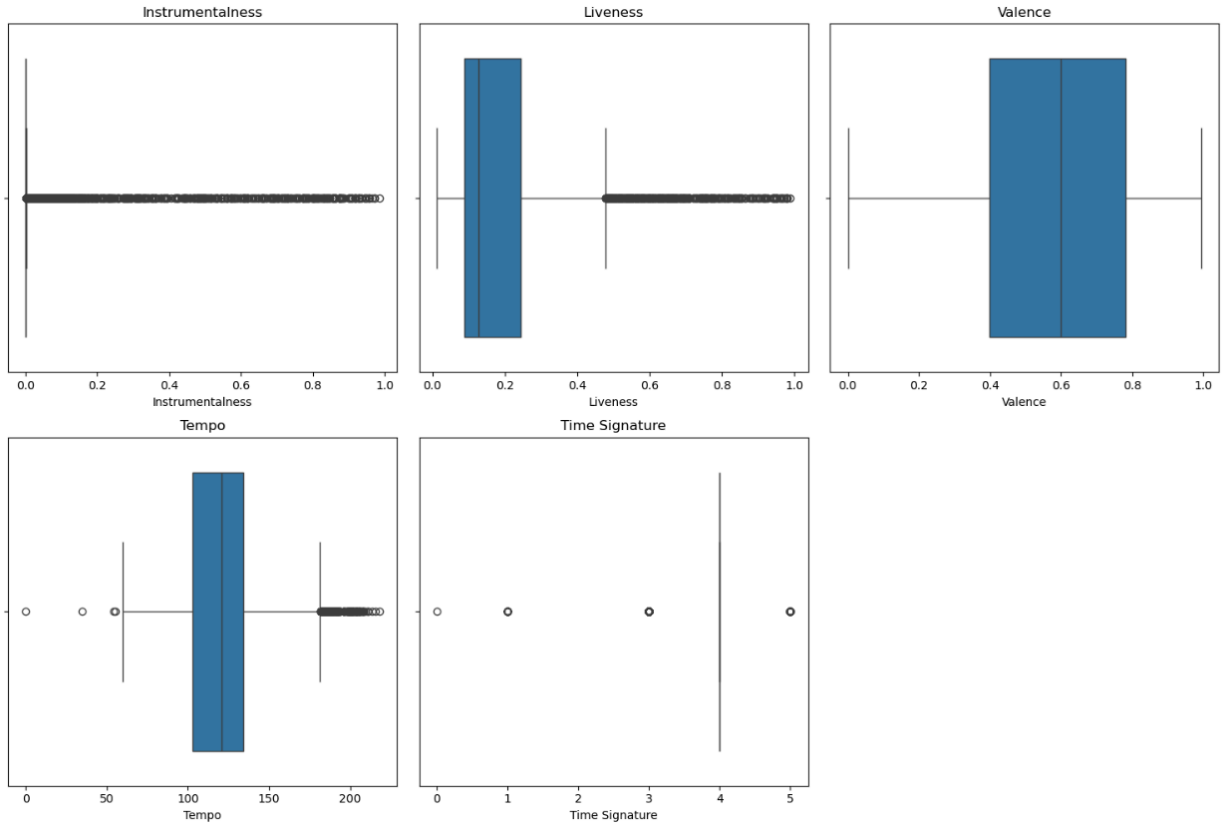


- Valence VS Danceability, Loudness VS Energy have highly positive correlation, which makes sense

- Acousticness VS Energy, Acousticness VS Loudness have highly negative correlation, which also makes sense

3. Boxplots - Check Feature Outliers





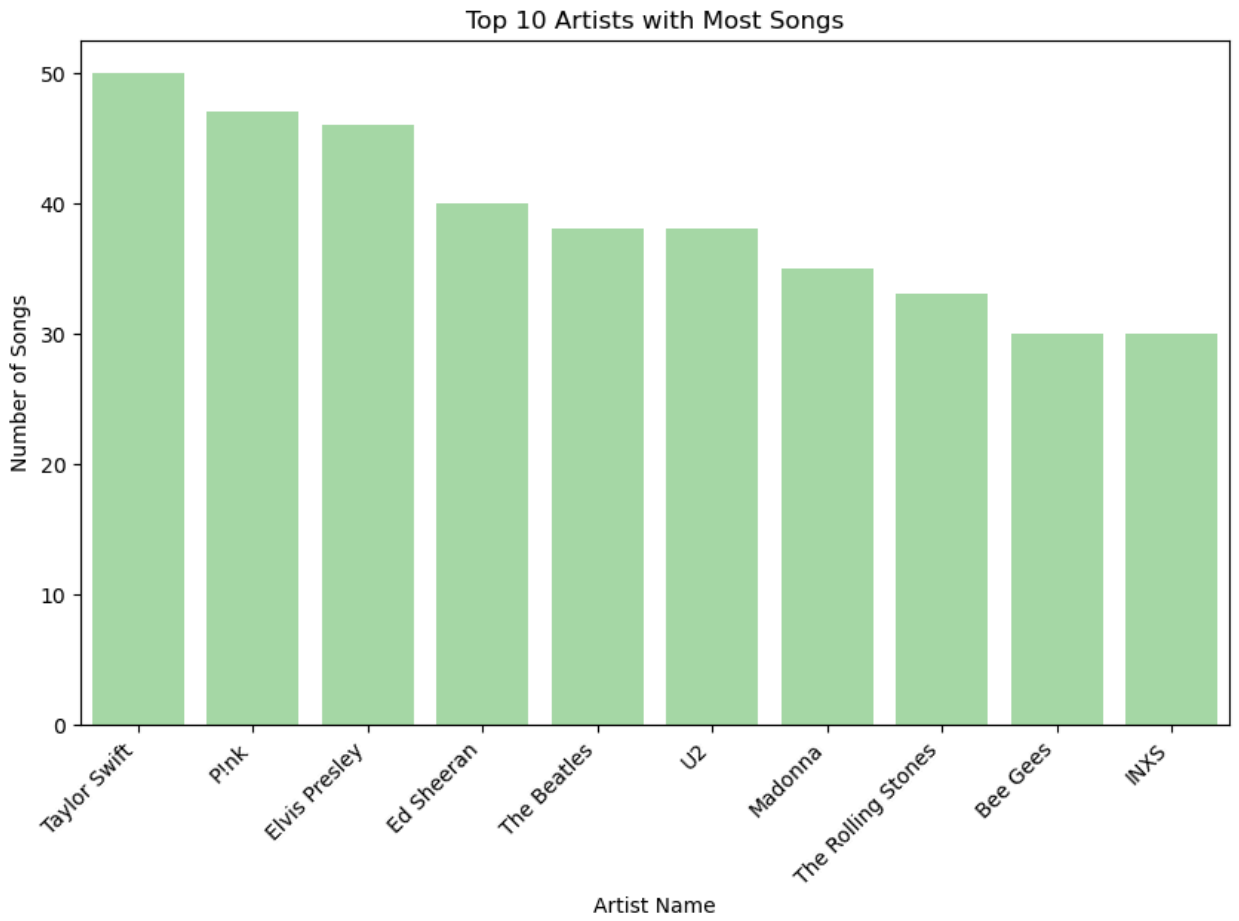
- The maximum disc in an album is around 15 according to Disc Number
- The maximum songs in an album is around 90 according to Track Number
- The maximum length of a song is around 1,600,000ms (= 1,600 seconds = 26 minutes = close to half an hour) according to Track Duration (ms)
- There are also extremely popular songs and extremely unpopular songs according to Popularity
- Loudness, Acousticness, Speechiness, Instrumentalness, and Liveness have lots of outliers
- Danceability, Energy have less outliers, and Valence don't have outliers
- Tempo also has a good number of extreme values, possibly very slow or very fast tracks
- Popularity, Mode, and Key do not have outliers, which makes sense as they have predefined ranges
- Time Signature has many outliers because most modern music only have 1 time signature of 4/4

Number of Outliers in Each Feature via IQR method:

- 'Unnamed: 0': 0,

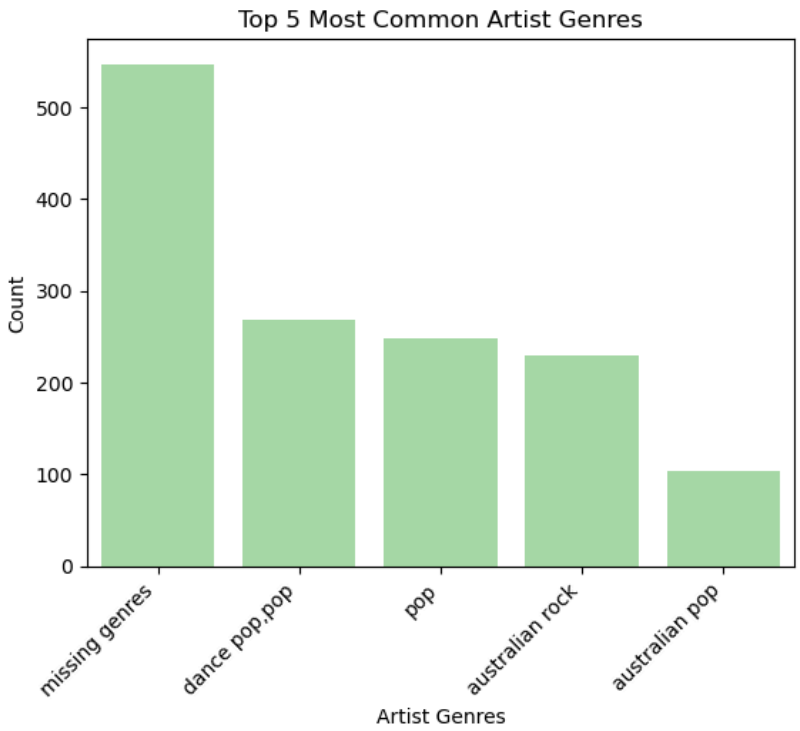
- 'Disc Number': 226,
- 'Track Number': 321,
- 'Track Duration (ms)': 284,
- 'Popularity': 0,
- 'Danceability': 71,
- 'Energy': 50,
- 'Key': 0,
- 'Loudness': 228,
- 'Mode': 0,
- 'Speechiness': 1043,
- 'Acousticness': 484,
- 'Instrumentalness': 2056,
- 'Liveness': 489,
- 'Valence': 0,
- 'Tempo': 237,
- 'Time Signature': 472

4. **Bar Graph - Top 10 Artists & Top 5 Genres**



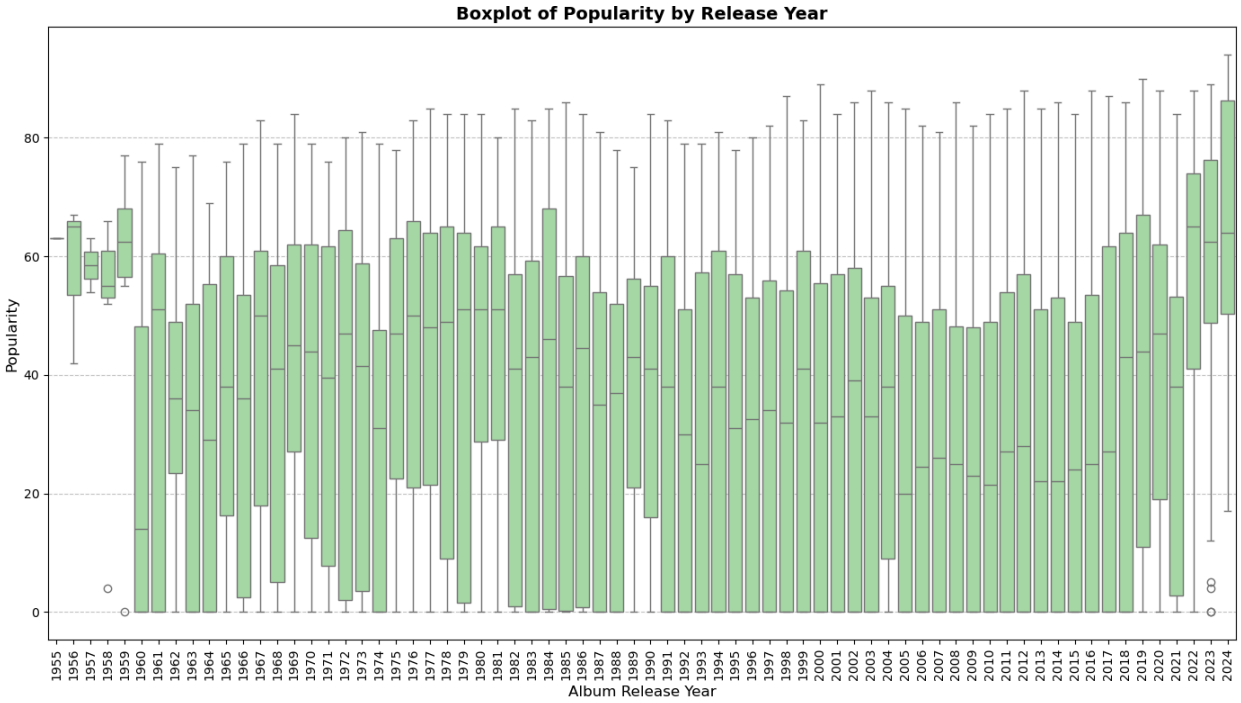
- The only two people I am familiar with from this list are Taylor Swift and Ed Sheeran…I need to have more fun rather than studying all day😀

  (PS: I know Elvis Presley, The Beatles, The Rolling Stones after translating into Chinese, and everyone knows Madonna as well. They all have their own famous time, and we just not live in their time unfortunately)



- We have over 500 songs missing their genres, the rest of them are most dance and pop

Bar Graph - Song Popularity Scores by Release Years

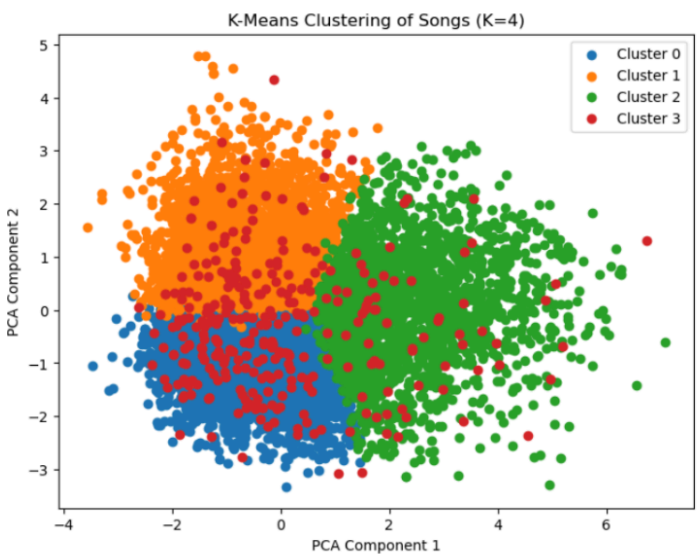
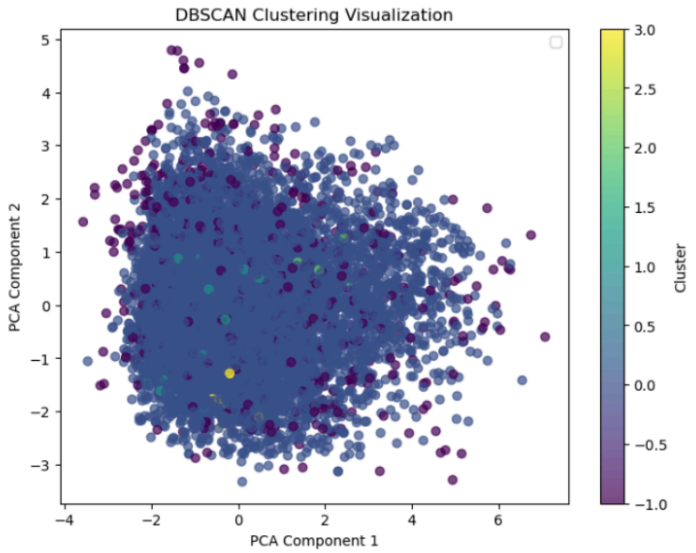


- This makes sense since most songs in 2024 (50-90) are more popular than songs in other years (0-70). Also, songs in 2022 and 2023 are also pretty popular among others (40-70)

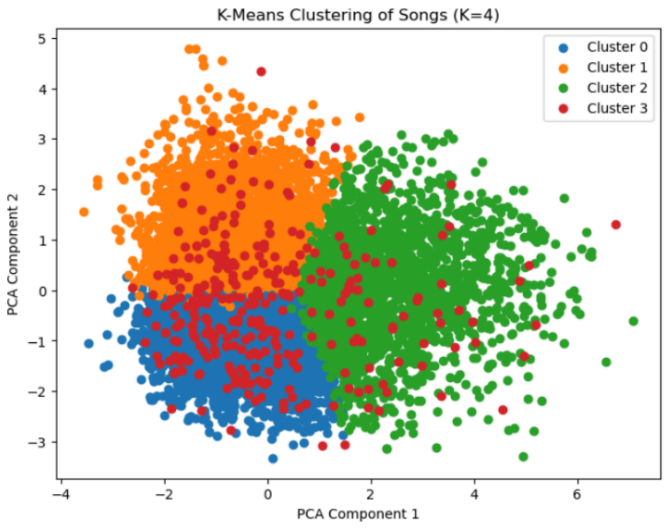
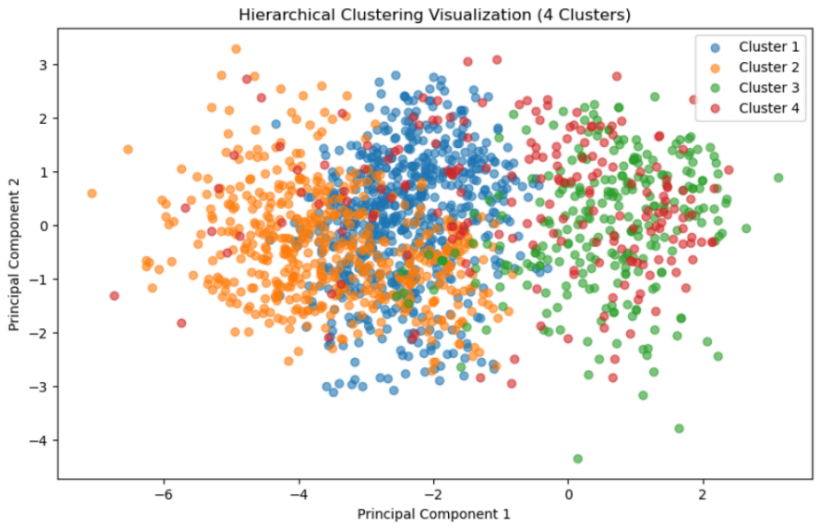
- Surprisingly, songs in 1956 to 1959 have very high popularity (50-70). From our research, it is due to the rise of rock and roll and the emergence of a new generation of artists, as well as the influence of television and radio on popular culture (It matches our Top 10 artists graph because one of the artists, Elvis Presley, was the product of that period)

5. K-Means Clustering VS DBSCAN Clustering VS Hierarchical Clustering

K-Means Clustering Results VS DBSCAN Clustering Results

Cluster	K-means	DBSCAN
		
0	Danceable & Vocal **Count:** 4251	Vocal **Count:** 9150
1	Energetic & Electric **Count:** 3256	Energetic & Loud & Instrumental **Count:** 85
2	Acoustic & Quiet **Count:** 2169	Quiet & Electric & Sad & Slow **Count:** 9
3	Instrumental & Pure Music **Count:** 317	Danceable & Happy & Studio **Count:** 18
-1	(N/A)	Noise / Outliers **Count:** 731

K-Means Clustering Results VS Hierarchical Clustering Results

Cluster	K-Means	Hierarchical
		
0	Danceable & Vocal **Count:** 4251	Danceable & Vocal **Count:** 7598
1	Energetic & Electric **Count:** 3256	Energetic & Happy **Count:** 759
2	Acoustic & Quiet **Count:** 2169	Acoustic & Quiet **Count:** 1172
3	Instrumental & Pure Music **Count:** 317	Instrumental & Electric **Count:** 464
N/A	Silhouette Score **Value:** 0.22	Silhouette Score **Value:** 0.17
N/A	Davies-Bouldin Index **Value:** 1.51	Davies-Bouldin Index **Value:** 1.70

- Comparing K-means to DBSCAN, K-means has more balanced cluster sizes than DBSCAN. DBSCAN has severe imbalanced clusters, the lowest size is 9 and the highest size is 9150. Therefore, K-means performs better

- Comparing K-means to Hierarchical clustering, Hierarchical still has imbalanced cluster issues but performs better than DBSCAN. However, according to evaluation metrics like Silhouette Score and Davies-Bouldin Index, K-means has more compact and well-separated clusters than Hierarchical.

- Therefore, we recommend K-means clustering (K=4) for this business problem and dataset

Popularity Prediction

Spotify offers a vast collection of songs for users to enjoy. From a business perspective, it is crucial for Spotify to predict whether a song will become popular. Accurate popularity prediction can help Spotify optimize recommendation systems, enhance user engagement, and optimize revenue through strategic marketing and playlist placements.
For our project, we decided to predict whether a song would be popular based on audio features, metadata, and artist information. Since popularity is a numerical variable, We transform the **popularity score** into a **binary classification problem** (with a threshold of 50.), apply **feature engineering**, handle **class imbalance** using **SMOTE**, and evaluate multiple **supervised learning models**.

Data Pre-processing
The following steps were taken to clean and prepare the data for further modeling use:
- **Dropped unnecessary columns** (e.g., URIs, preview URLs)
- **Removed duplicate entries** (final dataset: **9,946 songs**)
- **Extracted album release year** from the full release date
- **Encoded categorical variables**, including **explicit content and track characteristics**
- **Created additional binary indicators**:
- Whether the song is from a **famous artist** (e.g., Drake, Rihanna, BTS, who are top 20 popular artists)
- Genre categorization (e.g., Pop, Rock, Hip-Hop, R&B)

Feature Engineering
To enhance model performance, feature engineering techniques were applied:

- **Track characteristics**: Clustered songs into **Danceable & Vocal, Energetic & Electric, Acoustic & Quiet, Instrumental & Pure Music**
- **Genre Encoding**: Assigned songs into **12 major genres** (e.g., Pop, Rock, Jazz, Hip-Hop)
- **Decade-based grouping**: Mapped album release year to a **decade feature** (e.g., 2010s, 2000s)

Supervised Models
Supervised machine learning models were used to predict song popularity:
Binary Classification problem

- **Popularity Threshold: We classified songs as popular (≥50 popularity score) or not popular (<50)**
- **Class Imbalance: The dataset was imbalanced, with 3,334 popular songs vs. 6,612 non-popular songs**
- **SMOTE (Synthetic Minority Over-sampling Technique) was applied to balance the dataset before training models**

Supervised Models:

- **Random Forest**
- **Gradient Boosting**
- **Logistic Regression**
- **Support Vector Machine**
- **XGBoost**
- **LightGBM**

Models Evaluation

The models were evaluated using multiple performance metrics to ensure robustness:

Model	Accuracy	F1 Score	Precision	Recall	ROC-AUC
LightGBM	0.58	0.56	0.43	0.82	0.67
Gradient Boosting	0.53	0.56	0.41	0.88	0.66
XGBoost	0.56	0.55	0.42	0.82	0.66
SVM	0.57	0.54	0.42	0.76	0.64
Logistic Regression	0.58	0.42	0.39	0.45	0.59
Random Forest	0.67	0.29	0.50	0.21	0.66

- **LightGBM, Gradient Boosting, and XGBoost** perform similarly in F1 Score (~0.55–0.56) and **high recall** (~0.82–0.88). This indicates that these models effectively identify popular songs but may also misclassify some non-popular songs.
- **Random Forest** achieves the highest accuracy (0.67), but its **low recall (0.21) and F1 Score (0.29)** indicate that it struggles with classifying popular songs correctly.
- **Logistic Regression** has the lowest F1 Score (0.42), meaning it does not effectively balance precision and recall.
- **SVM** has a balanced performance but does not outperform tree-based models.
- Since the goal of the prediction is to **identify popular songs correctly**, **precision is the most critical metric**. LightGBM and XGBoost provide reasonable precision (0.43 and 0.42, respectively).

Given the focus on predicting whether a song will become popular, **LightGBM is the best choice** as it maintains a good balance between **precision (0.43), recall (0.82), and ROC-AUC (0.67)**. However, the

overall precision scores are relatively low, suggesting additional features, such as user behavior and demographic data, are needed to improve model performance.

Limitation

Despite the use of advanced models and data balancing techniques, some limitations exist:

- Prediction Performance: The prediction results were unsatisfactory because the dataset only provided song characteristics without user interactions, media trends, or promotional activities, which may not be sufficient for predicting popularity.
- Lack of External Information: To improve prediction accuracy, additional data sources such as user behavior, demographic data, and artist or song-specific information should be incorporated.
- Class imbalance: Even after applying **SMOTE**, models still **struggled with recall** for predicting popular songs.

Conclusion

Predicting song popularity using metadata and audio features alone is limited, as external factors like user engagement, social media trends, and playlist placements heavily influence success. While LightGBM and Gradient Boosting showed moderate predictive power (F1-score ~0.56), metadata alone is insufficient for accurate predictions.

For Spotify, integrating user behavior, streaming trends, and social media data would significantly improve prediction accuracy. Enhancing recommendation systems with these insights can optimize playlist curation, improve user engagement, and drive strategic marketing efforts to better anticipate hit songs.

Recommendation System

Song recommendation is integral to users' experiences at Spotify. Our goal is to connect users with relevant songs based on their preferences, taste, and current trends. To build Spotify's song recommendation system, we use **content-based filtering algorithm** and leverage on three components:

1. **Similarity of songs** based on **audio features, popularity score, and sentiment** - how similar songs based on audio-related components, such as loudness, danceability, instrumentality, whether songs have similar popularity score and sentiment (positive, negative, or neutral). For example, a user might be interested in a sad song with low level of loudness and full of instrumentality. Therefore, we would like to recommend songs with similar sentiment, audio features, as well as, popularity.
2. **Similarity of song** based on **lyrics** - whether songs have semantic similarities in their lyrics. For example, a song with lyrics about heartbreak, would be recommended with another song with similar context.
3. **Similarity of song** based on **artist information** - how similar artist is based on genre and artist name. For example, a user might be a fan of Taylor Swift and pop songs. Hence, we hope to recommend songs from the same artist or same/similar genres.

Similarity of songs based on audio features, popularity, and sentiment

We utilize audio features to analyze similarity of songs, including Danceability, Energy, Loudness, Speechiness, Acousticness, Instrumentalness, Liveness, Valence, Tempo, Key, Mode, and Time Signature. Additionally, we incorporate popularity scores as a key feature. Since popularity data may not be available when a song is first released, a popularity prediction model can be used to estimate this value. However, in our dataset, all songs already have popularity scores, so we directly use the available data.

To enhance song analysis, we also extract sentiment features from lyrics. The lyrics are preprocessed by removing stopwords and tokenizing before applying sentiment analysis using the **TextBlob library** in Python. The output is a polarity score, a float value ranging from -1.0 to 1.0:

- -1.0 → Highly negative sentiment (e.g., sad or angry lyrics)
- 0.0 → Neutral sentiment
- 1.0 → Highly positive sentiment (e.g., cheerful or happy lyrics)

	count	mean	std	min	25%	50%	75%	max
Danceability	5531.0	0.61	0.15	0.00	0.51	0.61	0.71	0.99
Energy	5531.0	0.68	0.19	0.02	0.56	0.71	0.84	1.00
Loudness	5531.0	-7.35	3.24	-29.37	-9.13	-6.61	-5.00	-0.28
Speechiness	5531.0	0.06	0.06	0.00	0.03	0.04	0.07	0.71
Acousticness	5531.0	0.21	0.25	0.00	0.02	0.10	0.34	0.99
Instrumentalness	5531.0	0.02	0.11	0.00	0.00	0.00	0.00	0.96
Liveness	5531.0	0.18	0.15	0.01	0.09	0.13	0.25	0.98
Valence	5531.0	0.59	0.24	0.00	0.40	0.60	0.79	0.99
Tempo	5531.0	121.25	26.60	0.00	101.97	120.06	135.00	217.91
Key	5531.0	5.16	3.59	0.00	2.00	5.00	8.00	11.00
Mode	5531.0	0.71	0.46	0.00	0.00	1.00	1.00	1.00
Time Signature	5531.0	3.96	0.25	0.00	4.00	4.00	4.00	5.00
Popularity	5531.0	36.24	27.67	0.00	0.00	42.00	60.00	90.00
sentiment	5531.0	0.10	0.20	-0.85	-0.02	0.09	0.23	1.00

Figure 1.0 Descriptive Statistics of Audio Features, Popularity, and Sentiment

After combining audio features and sentiment scores of songs, we observed that each feature had a different range of values. To ensure fair comparisons and prevent any single feature from dominating the similarity calculation, we applied **StandardScaler**, which standardizes all numeric features by transforming them to have a mean of 0 and a standard deviation of 1.

Besides, we use clustering labels from the K-means clustering model as categorical features to represent different song genres. These labels are converted into numerical form using get_dummies and then combined with the scaled numeric features using hstack. This integration ensures that both categorical and numerical features contribute effectively to the similarity calculation.

After combining all audio features, we compute the **cosine similarity**, generating an n x n matrix that captures the similarity between each pair of songs based on these features. To ensure compatibility with other

similarity scores, we **normalize the matrix** by shifting and scaling the values so that the similarity scores fall within a range of 0 to 1. This adjustment allows for a balanced combination with other cosine similarity features.

Similarity of song based on lyrics

We perform **keyword extraction** using the **RAKE library**, which tokenizes words from each email and performs word scoring based on frequency and co-occurrence patterns. Output is stored in the keywords_lyrics column. After extracting keywords using RAKE, we still found non-alphabetical characters, such as "?" and "(", hence we also **removed unnecessary characters** and store the cleaned version under **keywords_lyrics_clean**

lyrics	keywords_lyrics	keywords_lyrics_clean
I'm in transit\n Floating, stranded on this boat\n And I pledge myself allegiance\n To a better night's sleep at home\n And the sweet, sweet sun's coming down hard\n The sun's coming down hard, it burns the bones\n So hold a hand for cover, hold a hand for cover\n Hold a hand for cover from harm\n Talk don't change a thing\n Oh, it's fading fader\n Words don't sink, it swims\n Oh, it's fading fader\n Bless this mess, we tried our best\n That's all that we can do\n While the angels walk with the lonely ones\n In the cold rain to rescue you\n And this fable world's coming down hard\n The world's coming down hard on all our homes\n So hold a hand for cover, hold a hand for cover\n Hold a hand for cover from harm\n Talk don't change a thing\n Oh, it's fading fader\n Words don't sink, it swims\n Oh, it's fading fader\n ♫\n Talk don't change a thing\n Oh, it's fading fader (I'm in transit)\n Words don't sink, it swims (the world might be there, the world might be there)\n Oh, it's fading fader (I'm in transit)\n Oh, it's fading fader (the world might be there, the world might be there)\n Oh, it's fading fader (I'm in transit)\n	[transit, floating, stranded, boat, pledge, allegiance, better, night, sleep, home, sweet, sun, coming, hard, burns, bones, hold, hand, cover, harm, talk, change, thing, oh, fading, fader, words, sink, swims, bless, mess, tried, best, angels, walk, lonely, ones, cold, rain, rescue, fable, world, homes, ♫, might]	transit floating stranded boat pledge allegiance better night sleep home sweet sun coming hard burns bones hold hand cover harm talk change thing oh fading fader words sink swims bless mess tried best angels walk lonely ones cold rain rescue fable world homes might

Figure 1.1 Data Cleaning Process for Lyrics including Keywords Extraction, Removing Stopwords, and Tokenization

Then, we vectorized our **keywords_lyrics_clean** into matrices using three different word-embedding approaches: **TF-IDF, Word2Vec,** and **BERT**. Though **TF-IDF** is a simple word-embedding approach by penalizing frequently appeared words, it produces sparse word matrices and is not able to get semantic meaning of the lyrics. **Word2Vec** comes into handy here to capture semantic meaning by learning the relationships surrounding the word context. This is indicated by similar word-relationships whose words are located together in higher dimensional space, such as "sad" and "mad". However, some words are strangely located in a similar vector space, like "regular", "angry", and "devotion".

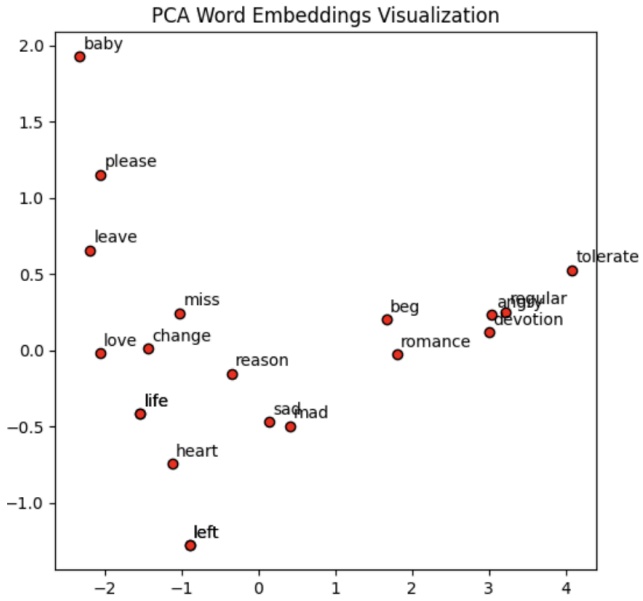


Figure 1.2 Visualization of Collections of Words from Lyrics in High-Dimension Vector Space

BERT embedding, fortunately, can capture semantic meaning at the sentence level and takes into account context, making it better for lyrics similarity. For example, "I feel blue today" vs "The sky is blue". Word2Vec considers these two sentences to be similar, whereas BERT refers to the first sentence to sadness and second sentence to color. For instance, we checked "**Sad**" by "Maroon 5" and found that songs with lyrics similarity are "**Crying for No Reason**" by "Katy B", "**Because of You**" by "Kelly Clarkson", and "**Tough**" by "Lewis Capaldi". By making educated guesses from the lyrics, these songs' themes are about heartbreak, regret, and emotional pain.

	track_name	artist_name	song_lyrics
0	Crying for No Reason	Katy B	I pushed all my problems to the back of my mind\n Then they surfaced in my dreams, they come alive\n I sweep all my issues to somewhere I can't find\n In hope that I'll forget, but there's just so many times\n Why can't I be strong, and just confront all my fears?\n When my fear is hurting you by being sincere\n But how many more days can I run? How many years?\n Emotions flooding, and now it's all seeming so clear\n Crying for no reason, feel the tears roll down\n I felt strong, but am I breaking now?\n Crying for no reason 'cause I buried it deep\n I made promises I could not keep\n 'Cause I never faced all the pain I caused\n Now the pain is hitting me full force\n .?\n I pushed all my problems to the back of my brain\n A darkness deep inside where I just can't find my way\n How can I walk with a smile? Get on with my day\n When I deceived myself pretending it's all okay\n I tried my best to hold it all together, I know\n The strings have worn away, and now I'm all exposed\n I tr...
1	Because of You	Kelly Clarkson	Ooh, ooh\n Ooh\n I will not make the same mistakes that you did\n I will not let myself cause my heart so much misery\n I will not break the way you did, you fell so hard\n I've learned the hard way to never let it get that far\n Because of you, I never stray too far from the sidewalk\n Because of you, I learned to play on the safe side, so I don't get hurt\n Because of you, I find it hard to trust not only me, but everyone around me\n Because of you, I am afraid\n I lose my way, and it's not too long before you point it out\n I cannot cry, because I know that's weakness in your eyes\n I'm forced to fake a smile, a laugh, every day of my life\n My heart can't possibly break when it wasn't even whole to start with\n Because of you, I never stray too far from the sidewalk\n Because of you, I learned to play on the safe side, so I don't get hurt\n Because of you, I find it hard to trust not only me, but everyone around me\n Because of you, I am afraid\n I watched you die, I heard you ...
2	Tough	Lewis Capaldi	Every little part of me\n Is holding on to every little piece of you\n Is holding on to every drop of blood you drew\n Is holding onto you\n And every waking hour I spend\n Holding out for reasons not to go to bed\n I'm holding out for someone else to hold instead\n If every hope of you is dead\n 'Cause every night since you cut and run\n I feel like the only one who's ever been the lonely one\n Trying to mend a heart that keeps breaking\n With every step that you're taking\n 'Cause you've been running circles 'round\n My mind, turnin' me inside out\n And I fell for you, but hit the ground\n 'Cause the only love I've known has let me\n Down, and I need liftin' up\n Now you ain't here I'm sleeping rough\n And I'm prayin' I can pray enough\n So wakin' up without you ain't so tough\n I find it hard to find my feet\n When I keep on stumbling over you and me\n But I keep on trying 'cause I know I need\n To outrun the memories\n And every day, I'm reminded of\n The way I let it come undo...
3	Malibu Nights	LANY	There's no reason, there's no rhyme\n I found myself blindsided by\n A feeling that I've never known\n I'm dealing with it on my own\n Phone is quiet, walls are bare\n I drink myself to sleep, who cares\n No one even has to know\n I'm dealing with it on my own\n .?\n I've got way too much time to be this hurt\n Somebody, help, it's getting worse\n What do you do with a broken heart?\n Once the light fades, everything is dark\n Way too much whiskey in my blood\n I feel my body giving up\n Can I hold on for another night?\n What do I do with all this time?\n .?\n Every thought when it gets late\n Put me in a fragile state\n I wish I wasn't going home\n Dealing with it on my own\n I'm praying but it's not enough\n I'm done, I don't believe in love\n Learning how to let it go\n Dealing with it on my own\n I've got way too much time to be this hurt\n Somebody, help, it's getting worse\n What do you do with a broken heart?\n Once the light fades, everything is dark\n Way too much whiskey i...
4	Amnesia	5 Seconds of Summer	I drove by all the places we used to hang out getting wasted\n I thought about our last kiss\n How it felt, the way you tasted\n And even though your friends tell me you're doing fine\n Are you somewhere feeling lonely\n Even though he's right beside you?\n When he says those words that hurt you\n Do you read the ones I wrote you?\n Sometimes I start to wonder, was it just a lie?\n If what we had was real, how could you be fine?\n 'Cause I'm not fine at all\n I remember the day you told me you were leaving\n I remember the make-up running down your face\n And the dreams you left behind, you didn't need them\n Like every single wish we ever made\n I wish that I could wake up with amnesia\n And forget about the stupid little things\n Like the way it felt to fall asleep next to you\n And the memories I never can escape\n 'Cause I'm not fine at all\n .\n The pictures that you sent me\n They're still living in my phone\n I'll admit I like to see them, I'll admit I feel alone\n And all my...

Figure 1.3 Sample of Songs with Highest Lyrics Similarity with "Sad" by "Maroon 5"

Afterwards, we compute **cosine similarity** from the song vectors for each word-embedding method. Finally, we normalize the similarity matrix by adjusting and scaling the values to fall within a range of 0 to 1, ensuring compatibility when comparing it with other cosine similarity features.

	0	1	2	3	4	...	379	380	381	382	383
0	-0.006774	-0.051937	0.062708	-0.001512	0.052102	...	-0.005658	0.032090	-0.010281	-0.007009	-0.005329
1	-0.075876	-0.031299	0.055969	0.020134	-0.048129	...	0.046712	0.003198	0.080382	-0.019965	-0.104067
2	-0.041455	-0.085146	0.087218	0.009781	0.049353	...	0.038653	0.033622	-0.036462	-0.026269	0.007393
3	-0.087284	0.006863	0.077970	0.048862	0.009378	...	-0.030187	0.106418	0.028516	-0.052849	0.021415
4	-0.021077	-0.051282	0.038390	0.060887	0.028547	...	0.029120	0.041625	0.029327	-0.007715	-0.084739
...	...	...	...	...	...	...	...	...	...	...	...
5526	-0.026670	-0.039442	-0.018629	0.015092	-0.008629	...	-0.013314	0.046375	0.008684	-0.038554	-0.015158
5527	-0.018773	0.021080	0.019311	0.007747	0.035917	...	0.003651	0.107160	-0.002361	0.050047	0.055357
5528	-0.008873	0.051540	-0.022936	0.002498	-0.024291	...	-0.003787	0.048069	-0.051757	0.008707	-0.043163
5529	-0.078631	-0.050005	0.107541	0.051721	0.024217	...	-0.057873	0.055417	-0.066109	0.025904	-0.009667
5530	-0.034219	-0.048376	0.061712	0.006054	-0.030641	...	-0.015165	-0.035990	-0.004509	-0.019952	-0.055800

Figure 1.4 BERT Embedding Matrices Used for Cosine Similarity Computation

The plot below shows distribution of lyrics similarity scores using the aforementioned three word-embedding approaches. Similarity score using TF-IDF is highly left-skewed, indicating that most songs do not have similar lyrics. Similarity score using Word2Vec is extreme to the right, which means most songs are similar to each other. The results do not quite make sense, as we have completely different context and meaning of lyrics. Moreover, Word2Vec might not work well because it fits a relatively small dataset (5k songs). On the other hand, BERT produces a more balanced distribution, suggesting that it does a better job at

differentiating between songs. Therefore, we decided to use BERT embedding to compute lyrics similarity scores.

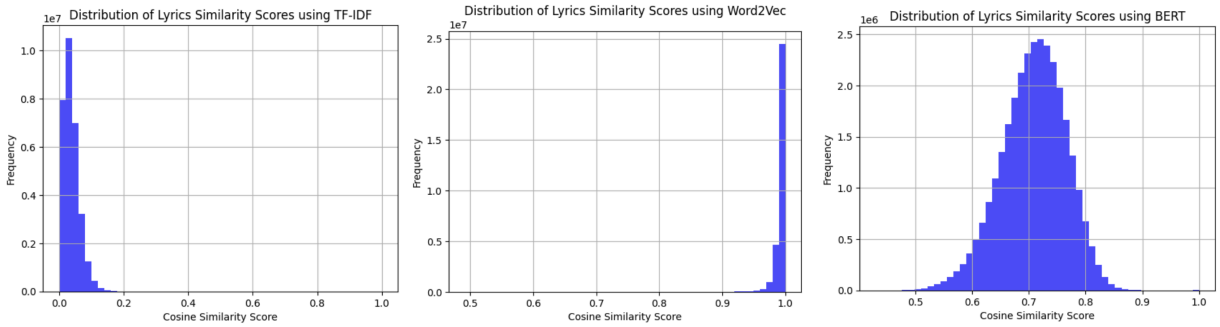


Figure 1.5 Distribution of Lyrics Similarity Scores using TF-IDF, Word2Vec, and BERT

Similarity of song based on artist information

We also conducted the same text-processing for **Artist Name** and **Artist Genres** by removing **stopwords**, such as "about", "who", "what", "your" and non-alphabetical characters. After cleaning the data, we combine **artist_name_clean** and **artist_genre_clean** into one combined column, called **artist_info.**

We used **CountVectorizer** to translate text of artist info into matrices and then we computed **cosine similarity** from these matrices.

Artist Name(s)	Artist Genres	artist_name_clean	artist_genre_clean	artist_info
The Temper Trap	indietronica,modern rock,shimmer pop	tempertrap	indietronica modernrock shimmerpop	tempertrap indietronica modernrock shimmerpop
Frankie Valli & The Four Seasons	adult standards,bubblegum pop,doo-wop,lounge,n...	frankievallifourseasons	adultstandards bubblegumpop doo-wop lounge nor...	frankievallifourseasons adultstandards bubbleg...
Foxes	electropop,metropopolis,uk pop	foxes	electropop metropopolis ukpop	foxes electropop metropopolis ukpop
Captain & Tennille	mellow gold,soft rock,yacht rock	captaintennille	mellowgold softrock yachtrock	captaintennille mellowgold softrock yachtrock
Rita Ora	dance pop,pop,uk pop	ritaora	dancepop pop ukpop	ritaora dancepop pop ukpop
...	...	...	...	...
Olivia Newton-John	adult standards,australian dance,disco,mellow ...	olivianewtonjohn	adultstandards australiandance disco mellowgol...	olivianewtonjohn adultstandards australiandanc...
Alanis Morissette	canadian pop,canadian singer-songwriter,lilith...	alanismorissette	canadianpop canadiansinger-songwriter lilith n...	alanismorissette canadianpop canadiansinger-so...
The Who	album rock,british invasion,classic rock,hard ...		albumrock britishinvasion classicrock hardrock...	albumrock britishinvasion classicrock hardroc...
New Kids On The Block	boy band	newkidsblock	boyband	newkidsblock boyband
Taco	missing genres	taco	missinggenres	taco missinggenres

Figure 1.6 Data Cleaning Process for Artist Information

Compute Final Similarity Score

We apply different weights to three different similarity scores, allowing for more flexibility to adjust for business needs. Higher weights are set to songs with similar audio features, sentiment, and popularity, followed by lyrics similarity and artist information (same artist or similar genres).

$$Final\ Similarity\ Score\ =\ 0.1\ *\ Artist\ Info\ Similarity\ Score\ +\ 0.25\ *\ Lyrics\ Similarity\ Score\ +\ 0.65\ Audio\ Similarity\ Score$$

Song recommender function

We use Track Name as the index and select the cosine similarity matrix, which contains similarity scores between the input song and all other songs. Then, we sort the similarity scores in descending order to find the most similar songs. After identifying the highest similarity scores, we extract the corresponding song details from the dataset. Finally, we visualize the recommendations in a DataFrame, displaying key attributes such as: song name, artist name, artist genres, artist similarity, lyrics similarity, audio similarity and final Similarity score. This output helps to understand which features contribute the most to the recommendation, making the system more explainable.

Sample of Song Recommendations

Input: "**When I Was Your Man**" by "**Bruno Mars**"

	Recommended Songs	Artist Name(s)	Artist Genres	Artist Similarity	Lyrics Similarity	Audio Similarity	Final Similarity
0	Count on Me	Bruno Mars	dance pop,pop	1.000000	0.740722	0.956688	0.907028
1	I'm Not a Girl, Not Yet a Woman	Britney Spears	dance pop,pop	0.666667	0.748794	0.946925	0.869366
2	Frozen	Madonna	dance pop,pop	0.666667	0.772427	0.925558	0.861386
3	Everytime	Britney Spears	dance pop,pop	0.666667	0.807453	0.896579	0.851306
4	Too Good At Goodbyes	Sam Smith	pop,uk pop	0.333333	0.814065	0.943483	0.850114
5	Dancing On My Own	Calum Scott	pop	0.408248	0.741483	0.959656	0.849972
6	Nothing Like Us	Justin Bieber	canadian pop,pop	0.333333	0.827766	0.937113	0.849398
7	Happier	Ed Sheeran	pop,singer-songwriter pop,uk pop	0.258199	0.794072	0.958930	0.847643

Figure 1.7 Song Recommendation from Input "When I Was Your Man" by "Bruno Mars"

The first recommended song similar to **When I Was Your Man** is **Count on Me** by Bruno Mars. This song falls under the pop/acoustic pop category, closely matching the pop/soul style of When I Was Your Man. Additionally, both songs explore themes of romantic relationships, and since they were produced by the same artist, all three main similarity features score highly, making Count on Me the top recommendation.

Another recommended song is **I'm Not a Girl, Not Yet a Woman** by Britney Spears. While its lyrics similarity is lower compared to other recommendations due to its theme not focusing on romantic relationships, the audio and artist style are quite similar, making it the second-best recommendation. In contrast, **Nothing Like Us** by Justin Bieber shares strong lyrics similarities, focusing on themes of love, heartbreak, and emotional longing. This high lyrics similarity contributes to Nothing Like Us becoming one of the recommended songs on the list.

Input: "**Fast Car**" by "**Tracy Chapman**"

	Recommended Songs	Artist Name(s)	Artist Genres	Artist Similarity	Lyrics Similarity	Audio Similarity
0	I Can't Make You Love Me	Bonnie Raitt	country rock,electric blues,folk,folk rock,mellow gold,singer-songwriter,soft rock	0.377964	0.692354	0.927467
1	Do You Really Want To Hurt Me	Culture Club	new romantic,new wave,new wave pop,soft rock,synthpop	0.000000	0.735582	0.967086
2	50 Ways to Leave Your Lover	Paul Simon	classic rock,folk,folk rock,mellow gold,permanent wave,rock,singer-songwriter,soft rock	0.358569	0.774192	0.889007
3	Fire and Rain - 2019 Remaster	James Taylor	classic rock,folk,folk rock,mellow gold,singer-songwriter,soft rock	0.400892	0.762920	0.884904
4	Bloodstream	Ed Sheeran	pop,singer-songwriter pop,uk pop	0.169031	0.725135	0.930705
5	Carolina in My Mind	James Taylor	classic rock,folk,folk rock,mellow gold,singer-songwriter,soft rock	0.400892	0.736410	0.888739
6	Baby Can I Hold You	Tracy Chapman	folk,lilith,singer-songwriter,women's music	1.000000	0.608349	0.833887
7	Walk On the Wild Side	Lou Reed	classic rock,glam rock,permanent wave,rock,singer-songwriter	0.285714	0.786005	0.868282

Figure 1.8 Song Recommendation from Input "Fast Car" by "Tracy Chapman"

Most of the song recommendations were released in the 1980s and 1990s, during which Fast Car was also released in 1982. The first recommended song similar to **Fast Car** is **I Can't Make You Love Me** by Bonnie Raitt. This song falls under the country rock, blues, and folk category, resembling quite closely with the folk style of Fast Car. The majority of other songs also share similar artist genres, which revolve around rock and country.

Model Evaluation

Since the recommendation system is an unsupervised learning task, traditional evaluation metrics like accuracy or F1-score cannot be used. Instead, we evaluate the model using the following approaches:

1. Model-based evaluation
2. User-based evaluation

Model-based evaluation

From the cosine similarity distribution, the distribution is approximately normal and centered around 0.5, indicating a balanced spread of similarity scores. Additionally, the scores are not concentrated near 0, meaning the model is unlikely to provide irrelevant recommendations. At the same time, the scores are not overly concentrated near 1, demonstrating that the features effectively distinguish differences between songs.

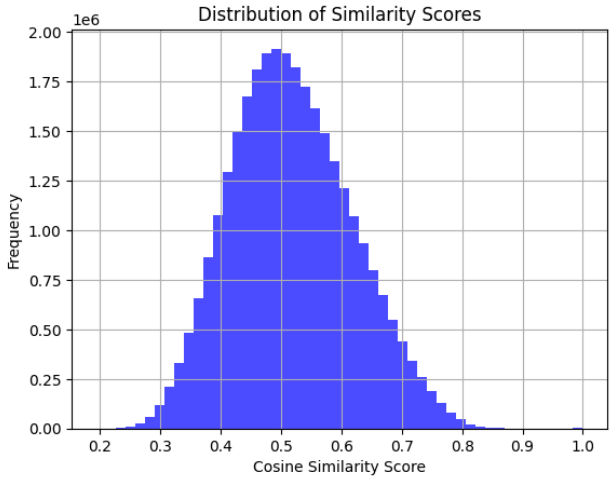


Figure 1.9 Distribution of Final Similarity Score

User-based evaluation

Our current data does not allow for user-based evaluation. However, once the model is implemented, **A/B testing** with real users can be conducted, using metrics such as Skip Rate to assess the effectiveness of the recommendations.

Business Value

This project analyzes Spotify's top 10,000 songs from 1950 to the present to improve music discovery and user experience. It enhances user engagement by recommending similar songs based on listening history, creating diverse song categories, and predicting potential hit songs, which can increase traffic on the platform. The insights gained from this analysis can improve playlist creation, enhance song recommendations, and help artists and labels understand evolving music trends.

1. Personalized Recommendations
   By utilizing cosine similarity with content-based approaches, recommendations are enhanced to be more personalized, resulting in increased user engagement and more relevant playlists.
2. Playlist Curation
   Clustering techniques like K-Means, DBSCAN, Hierarchical clustering enable streaming services to better understand the data by adding characteristics label to each song, and use the results as new data features and input for personalized recommendations to do further analysis
3. Predicting Hit Songs and Market Trends
   Classification models such as Random Forest and XGBoost are used to predict hit songs, allowing artists and labels to enhance their song production and marketing strategies. This also helps streaming platforms prioritize trending tracks. Furthermore, the findings from this project assist analysts in understanding long-term music trends and informing future decisions in the industry.

Conclusion

By fulfilling objectives, this project refines recommendations, improves playlist curation, and predicts hits, ultimately enhancing the user experience and supporting industry stakeholders.