ECHOES OF POPULARITY: AI-DRIVEN PREDICTION OF MUSIC STREAMING SUCCESS

DONE BY
VENKAT ROHAN SEETEPALLI

OVERVIEW

- Introduction
- Literature Review
- Data Preprocessing
- Exploratory Data Analysis
- Terminology
- Data Modelling and Evaluation
- Summary

INTRODUCTION

Music streaming has revolutionized how we access and enjoy music. Song popularity is driven by musical traits (tempo, energy) and external factors (artist reputation, social trends). Using Machine learning and Deep learning, we analyze these influences to predict streaming success. Our insights empower artists, producers, and platforms to optimize music reach.

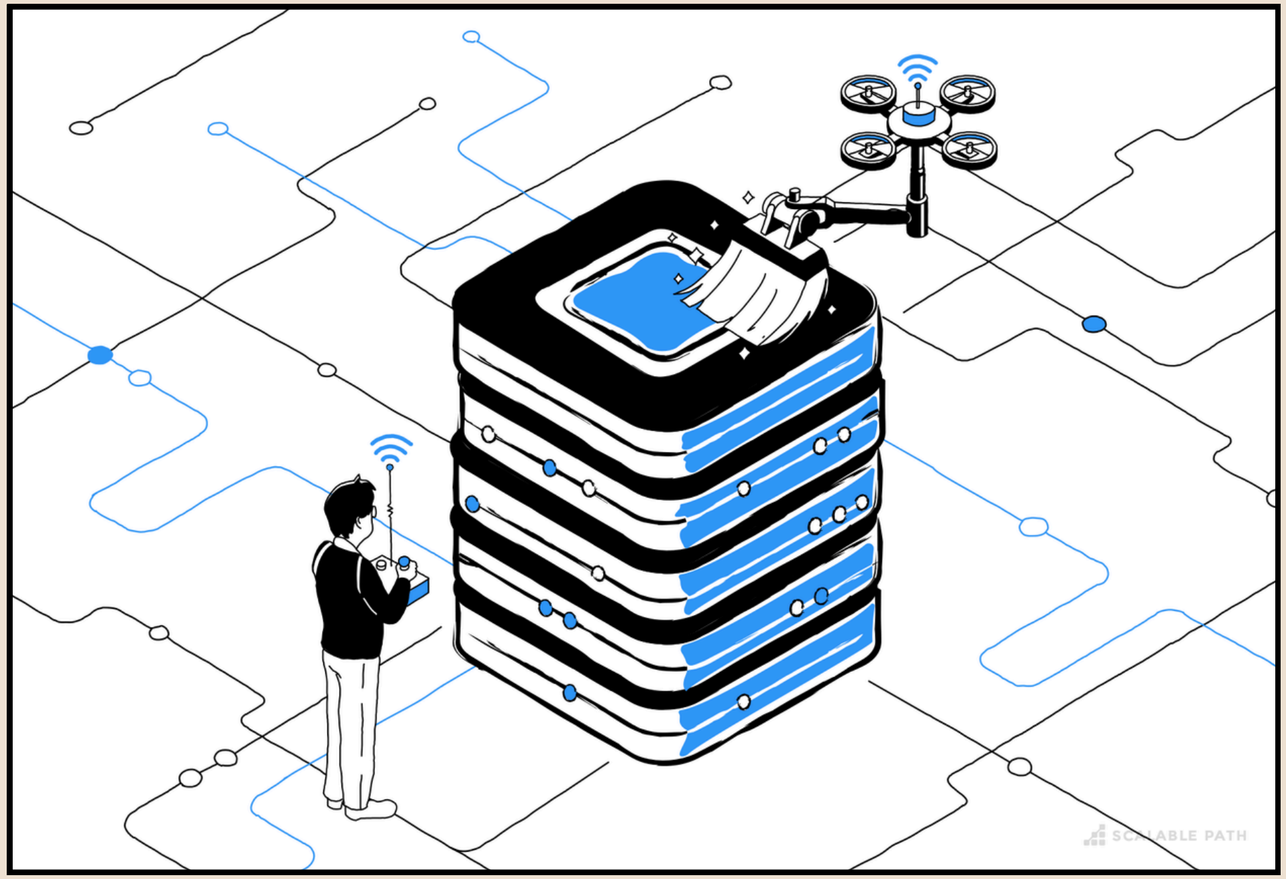
LITERATURE REVIEW

1. In the *2021* paper "*Catching the Earworm: Understanding Streaming Music Popularity Using Machine Learning Models*," *Andrea Gao* examines how audio features and artist information can predict music popularity on streaming platforms. The study employs machine learning models to identify key factors influencing a song's success, offering valuable insights for the music industry.

2. "*Predicting Song Popularity" by James Pham, Edric Kyauk, and Edwin Park (2012)* explores the use of machine learning algorithms to forecast a song's success. Utilizing the Million Song Dataset, the study evaluates various classification and regression models to identify key features influencing song popularity.

3. *"Hit Songs Prediction: A Review on Machine Learning Perspective" by Kah Yee Yap and Mafas Raheem (2024)* examines the evolution of Hit Song Science, focusing on the application of machine learning techniques to predict music popularity. The paper highlights the integration of audio features, lyrics analysis, and social media data, suggesting that future research should combine these elements to enhance prediction accuracy.

DATA PREPROCESSING



DATA

DATASET

My Dataset consists of 25 variables and 953 records

SOURCE

https://www.kaggle.com/datasets/abdulszz/spotify-most-streamed-songs/datat

ABOUT THE DATSET:

Our Kaggle dataset of 953 songs with 25 attributes captures both musical and market-driven factors behind streaming success. It includes audio features like energy, danceability, and valence, along with playlist presence and chart rankings. This unique blend of data provides a holistic view of what makes a song a hit. By analyzing these factors, we unlock patterns that drive music popularity

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
rack_nam	artist(s)_na	artist_cour	released_y	released_r	released_c	in_spotify_	in_spotify_	streams	in_apple_p	in_apple_c	in_deezer_	in_deezer_	in_shazam	bpm	key	mode	danceabili	valence_%	energy_%	acousticne	instrument	liveness_%	s
even (fea	Latto, Jung	2	2023	7	14	553	147	1.41E+08	43	263	45	10	826	125	B	Major	80	89	83	31	0	8	
ALA	Myke Towe	1	2023	3	23	1474	48	1.34E+08	48	126	58	14	382	92	C#	Major	71	61	74	7	0	10	
ampire	Olivia Rodr	1	2023	6	30	1397	113	1.4E+08	94	207	91	14	949	138	F	Major	51	32	53	17	0	31	
Cruel Sum	Taylor Swif	1	2019	8	23	7858	100	8.01E+08	116	207	125	12	548	170	A	Major	55	58	72	11	0	11	
WHERE SH	Bad Bunny	1	2023	5	18	3133	50	3.03E+08	84	133	87	15	425	144	A	Minor	65	23	80	14	63	11	
Sprinter	Dave, Cent	2	2023	6	1	2186	91	1.84E+08	67	213	88	17	946	141	C#	Major	92	66	58	19	0	8	
Ella Baila S	Eslabon Ar	2	2023	3	16	3090	50	7.26E+08	34	222	43	13	418	148	F	Minor	67	83	76	48	0	8	
Columbia	Quevedo	1	2023	7	7	714	43	58149378	25	89	30	13	194	100	F	Major	67	26	71	37	0	11	
ukumean	Gunna	1	2023	5	15	1096	83	95217315	60	210	48	11	953	130	C#	Minor	85	22	62	12	0	28	
a Bebe - P	Peso Plum	2	2023	3	17	2953	44	5.54E+08	49	110	66	13	339	170	D	Minor	81	56	48	21	0	8	
n x100to	Bad Bunny	2	2023	4	17	2876	40	5.06E+08	41	205	54	12	251	83	F#	Minor	57	56	72	23	0	27	
Super Shy	NewJeans	1	2023	7	7	422	55	58255150	37	202	21	5	168	150	F	Minor	78	52	82	18	0	15	
Flowers	Miley Cyrus	1	2023	1	12	12211	115	1.32E+09	300	215	745	58	1,021	118		Major	71	65	68	6	0	3	
Daylight	David Kush	1	2023	4	14	3528	98	3.88E+08	80	156	182	24	1,281	130	D	Minor	51	32	43	83	0	9	
As It Was	Harry Style	1	2022	3	31	23575	130	2.51E+09	403	198	863	46		174	F#	Minor	52	66	73	34	0	31	
Kill Bill	SZA	1	2022	12	8	8109	77	1.16E+09	183	162	161	12	187	89	G#	Major	64	43	73	5	17	16	
Cupid - Tw	Fifty Fifty	1	2023	2	24	2942	77	4.97E+08	91	212	78	6	0	120	B	Minor	78	76	59	43	0	34	
What Was	Billie Eilish	1	2023	7	13	873	104	30546883	80	227	95	24	1,173	78		Major	44	14	9	96	0	10	
Classy 101	Feid, Youn	2	2023	3	31	2610	40	3.35E+08	43	100	54	14	187	100	B	Major	86	67	66	14	0	12	
Like Crazy	Jimin	1	2023	3	24	596	68	3.63E+08	8	104	23	2	29	120	G	Major	63	36	73	0	0	36	
LADY GAG	Gabito Ball	3	2023	6	22	332	26	86444842	11	163	10	4	0	140	F	Minor	65	87	74	22	0	42	
Can See Y	Taylor Swif	1	2023	7	7	516	38	52135248	73	119	42	1	150	123	F#	Major	69	82	76	6	0	6	
Wanna Be	Arctic Mon	1	2013	1	1	12859	110	1.3E+09	24	98	582	2	73	135		Minor	48	44	42	12	2	11	
Peso Plum	Bizarrap, P	2	2023	5	31	1313	40	2.01E+08	17	152	32	11	139	133	F	Minor	85	81	67	26	0	12	
Popular (w	The Weekr	3	2023	6	2	1945	87	1.15E+08	74	182	87	14	1,093	99	C#	Major	85	83	68	7	0	36	

VARIABLES

CONTINUOUS	CONTINUOUS	CONTINUOUS
artist(s)_count	in_deezer_charts	acousticness_%
released_year	in_apple_charts	liveness_%
released_month	streams	speechiness_%
released_day	bpm	in_shazam_charts
in_spotify_playlists	danceability_%	
in_deezer_playlists	valence_%	
in_spotify_charts	instrumentalness%	
in_apple_playlists	energy_%	

CATEGORICAL
track_name
artist(s)_name
key
mode
cover_url

DATA CLEANING

Data Cleaning **: Data cleaning is the process of identifying and correcting or removing inaccurate, incomplete, or irrelevant data from a dataset .The process performed for data cleaning are as follows:**

- Filled missing values in the “key” column with the mode and “in_shazam_charts” with mean.
- Replaced “mode” column into 0's and 1's.
- Converted “streams” column to numeric and created a new variable called “***popular_category1***” (**target variable**) and mapped into 0's, 1's and 2's.
- Converted “in_deezer_playlists” and “in_shazam_charts” to numeric
- Conducted Label Encoding for 2 columns- “artist_name” and “track_name”
- Removed the column- “cover_url”

key	0
in_shazam_charts	0

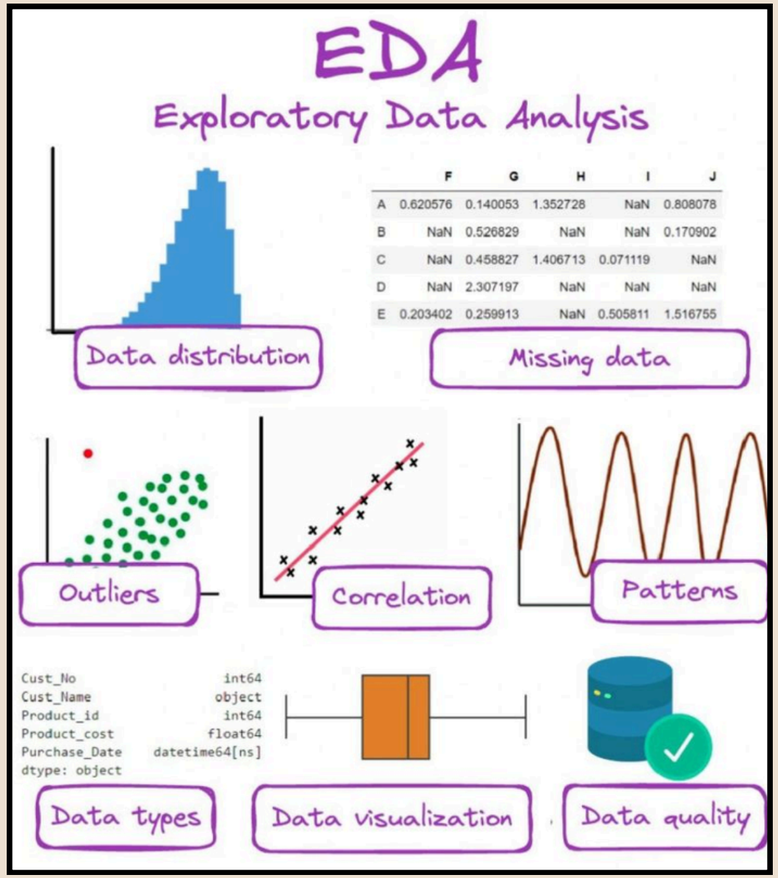
popular_category1	count
Highly Popular	324
Low Popular	315
Moderately Popular	314

dtype: int64

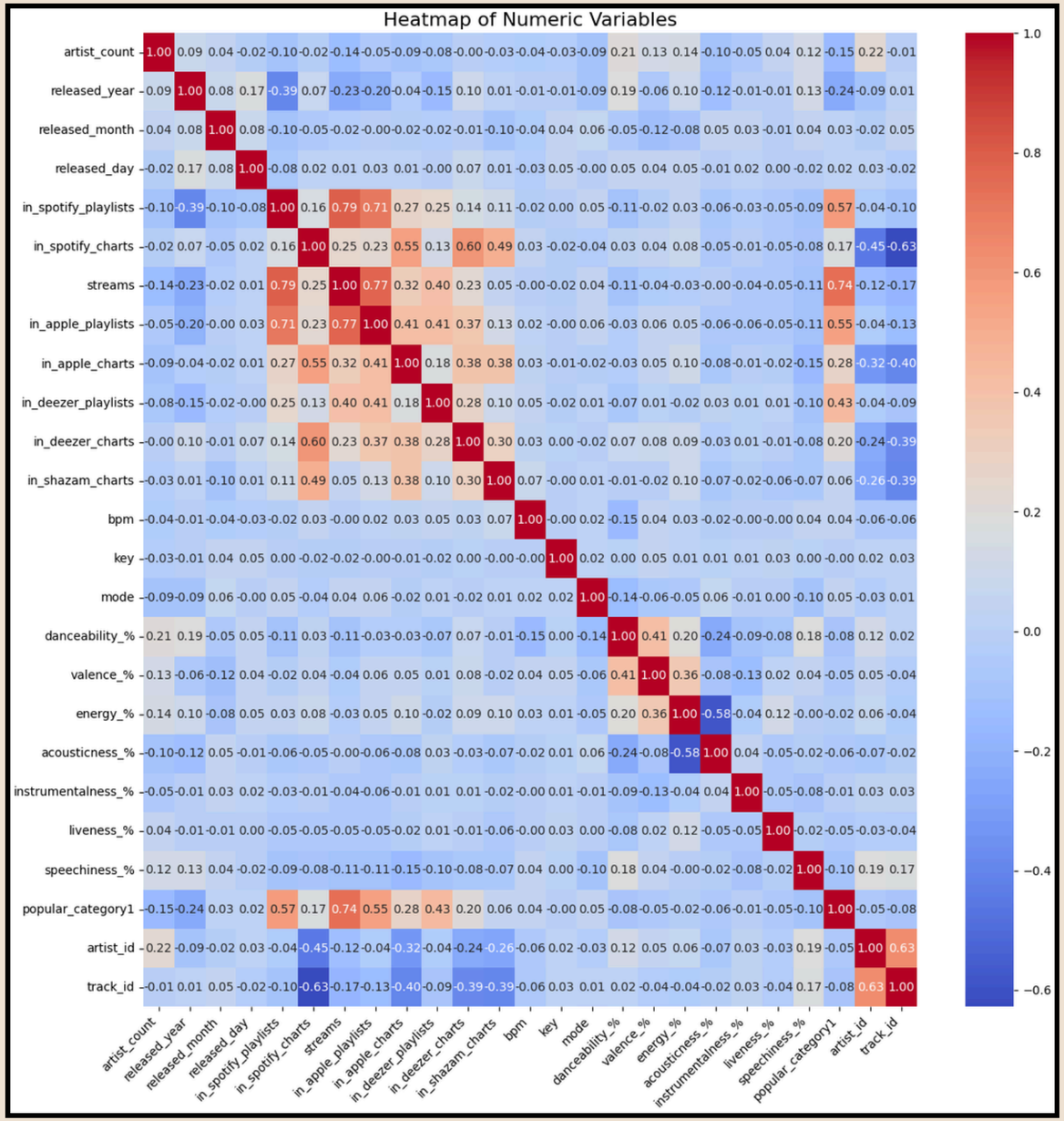
	key
0	2
1	3
2	7
3	0
4	0
...	...
948	0
949	8
950	3
951	3

	artist_id
0	0
1	1
2	2
3	3
4	4
...	...
948	642
949	3
950	643

EXPLOARATORY DATA ANALYSIS

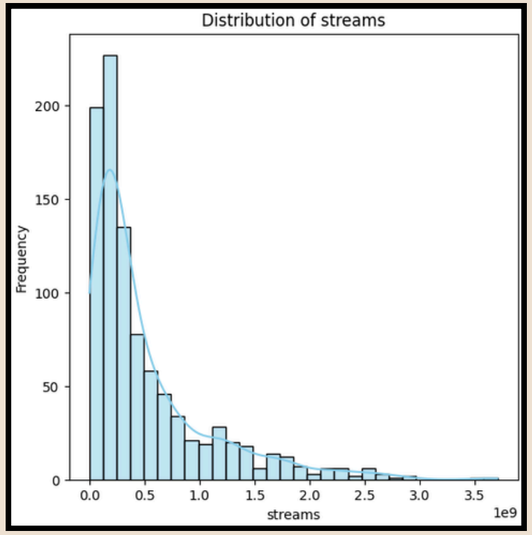


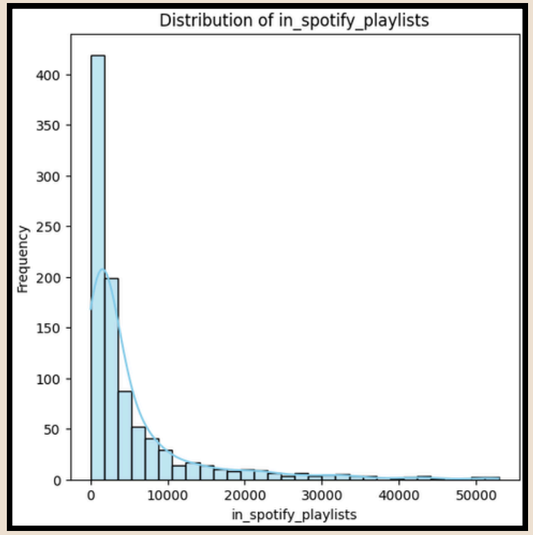
CORRELATION HEATMAP

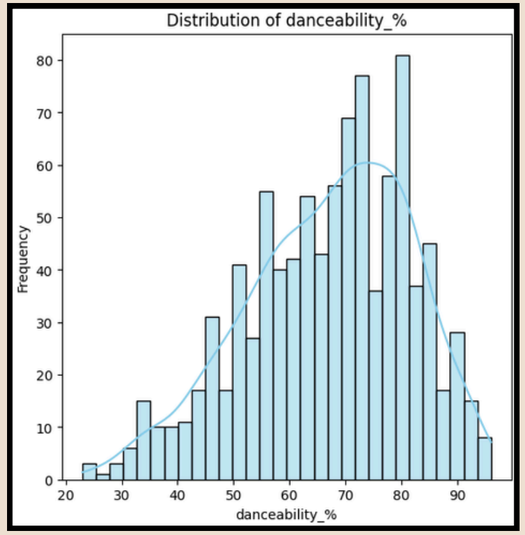


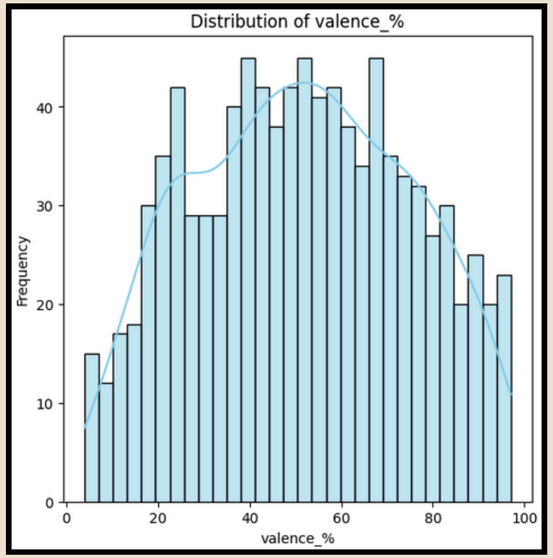
This visualizes the relationships between numerical features. It highlights correlations, helping identify features that might be redundant or strongly associated.

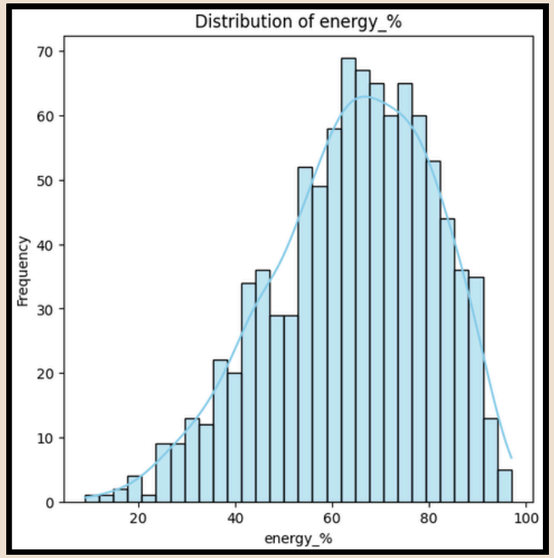
HISTOGRAM







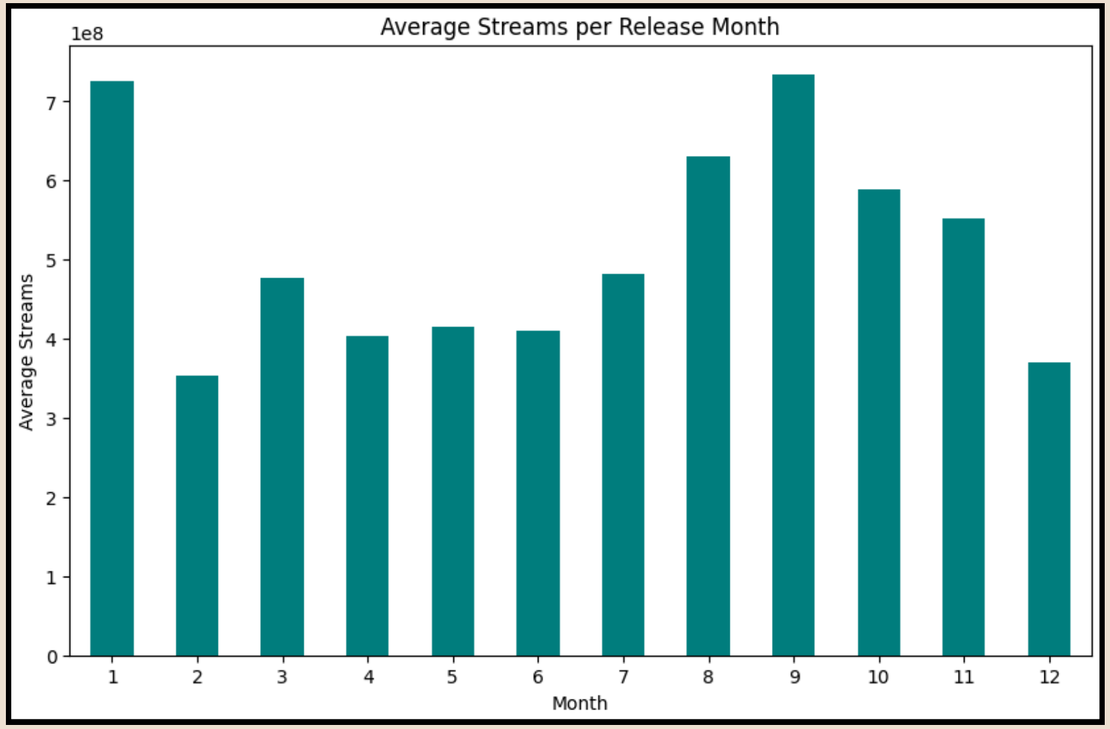


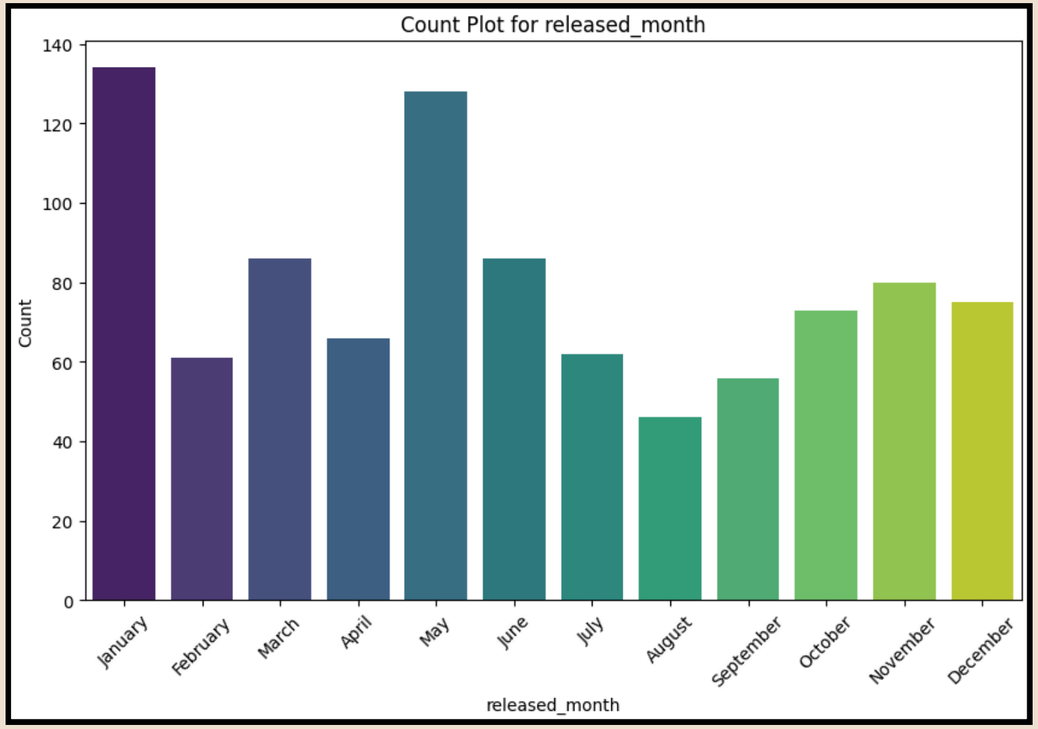


Histograms show how the distribution of variables, helping us understand the frequency of different values for each feature.

COUNT-PLOT

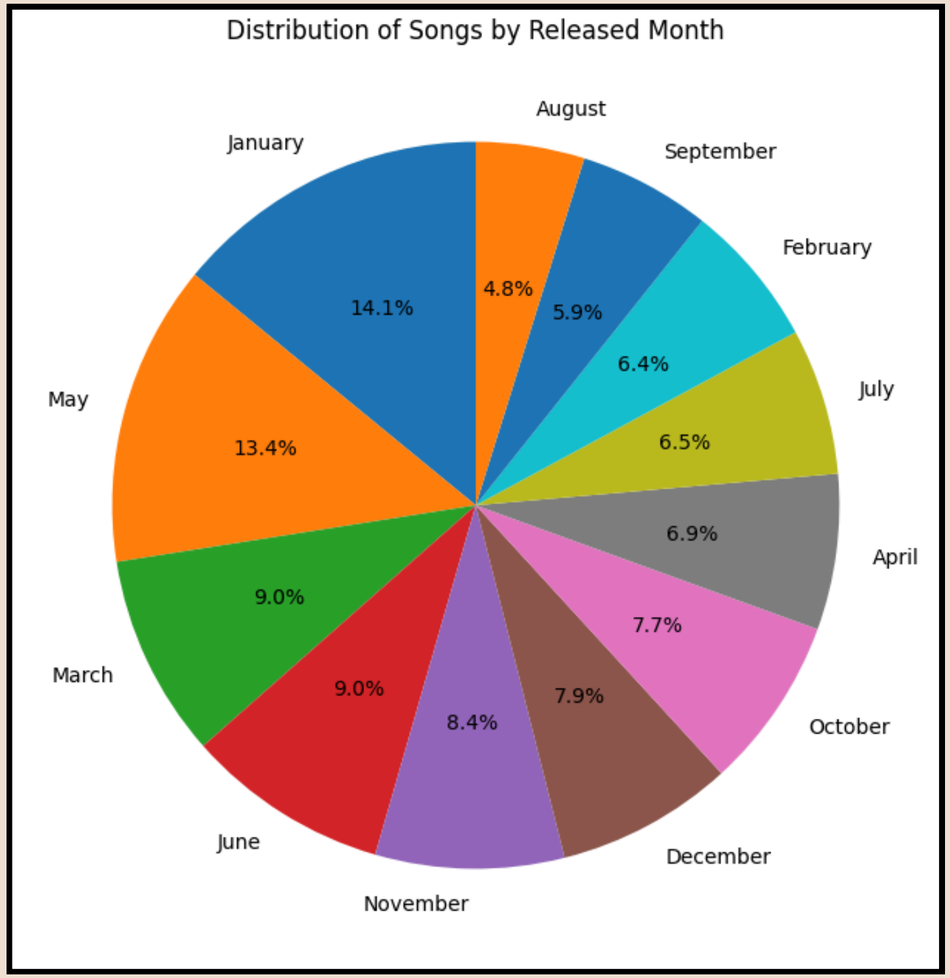
Count Plots give an idea of the categorical variables and their distributions. For example, for released_month.

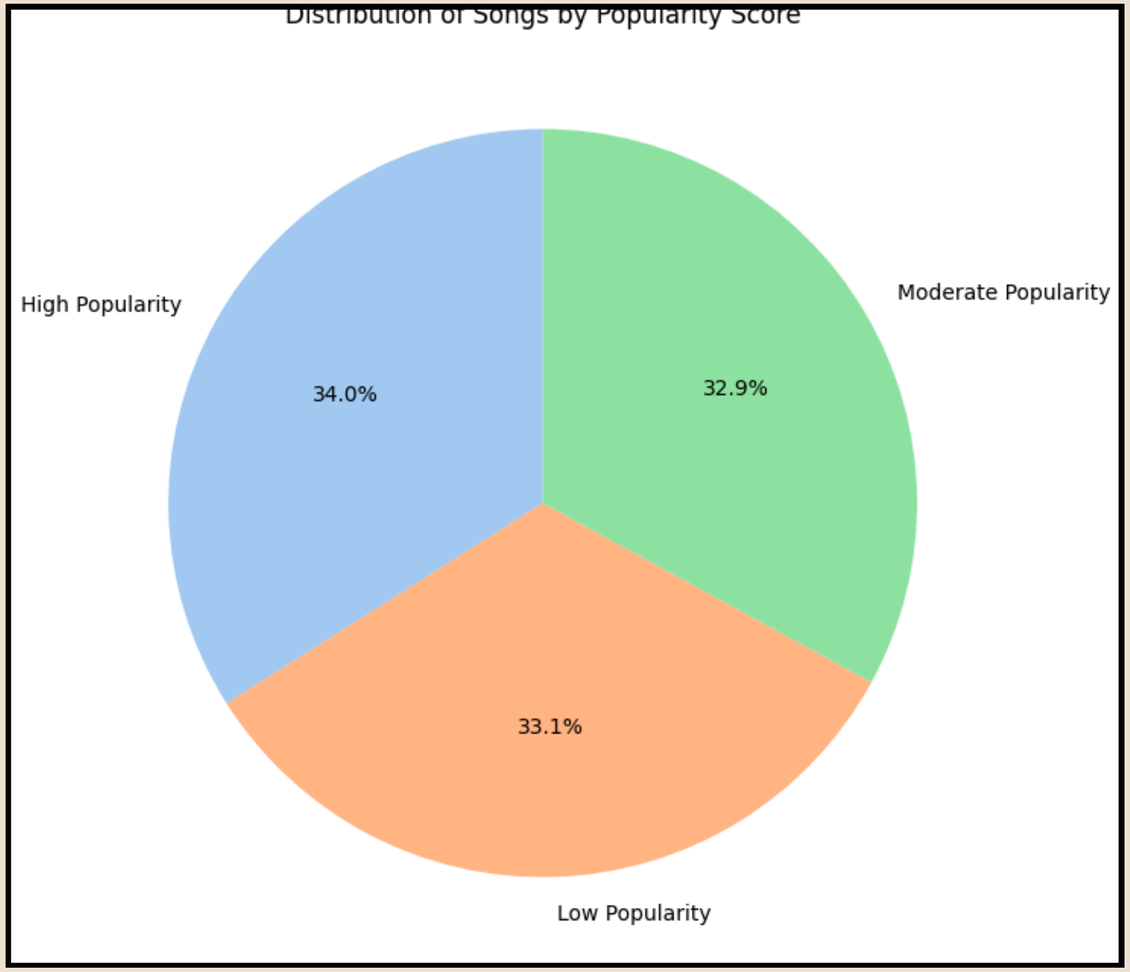




PIE-PLOT

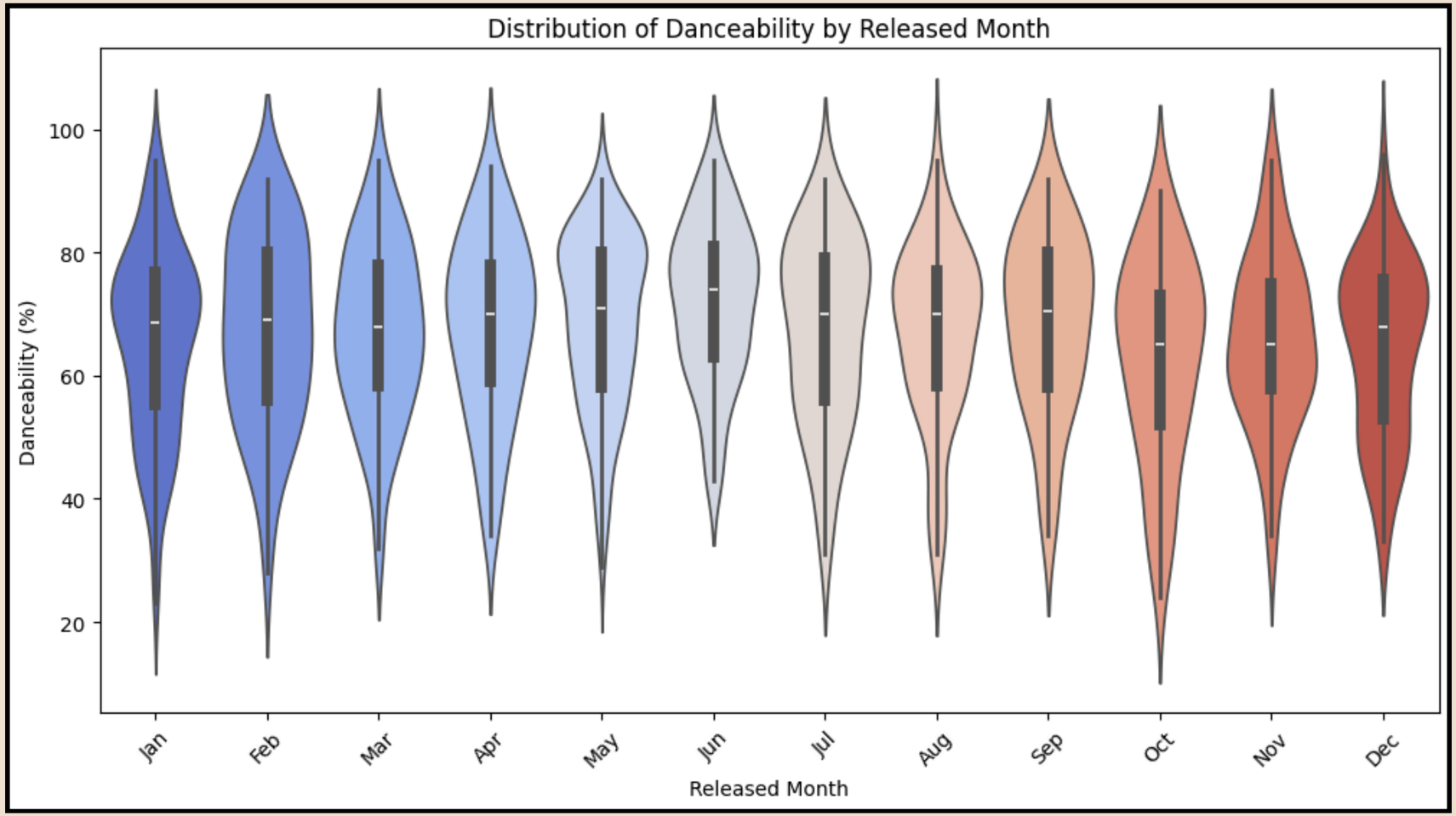
Pie plots visually break down proportions of a variable.





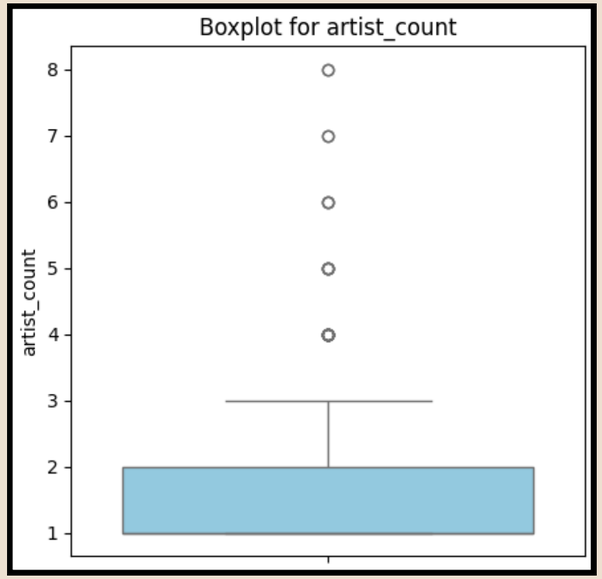
VIOLIN-PLOT

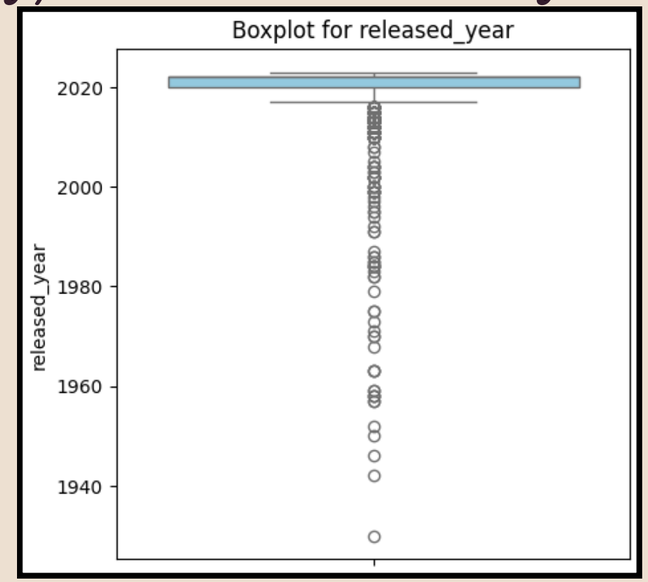
These plots show the spread and potential outliers for features. Violin plots combine box plot information with the distribution shape.

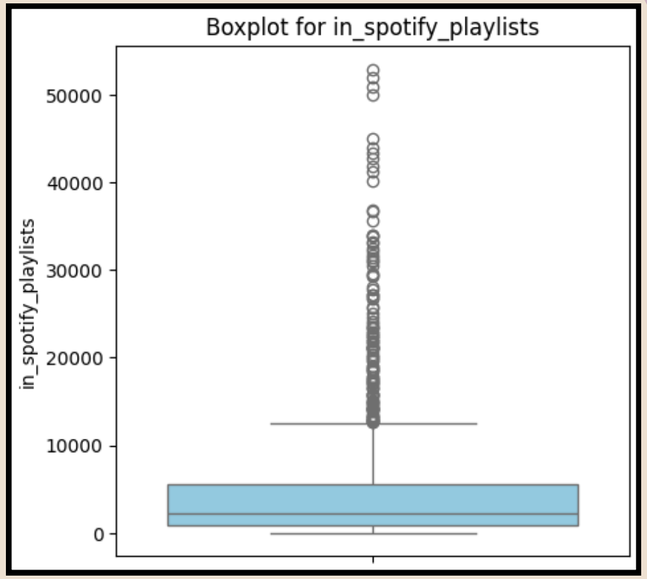


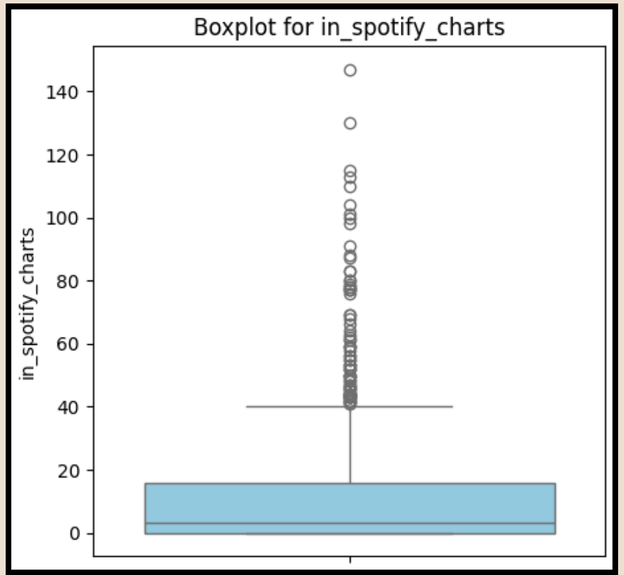
BOX-PLOT

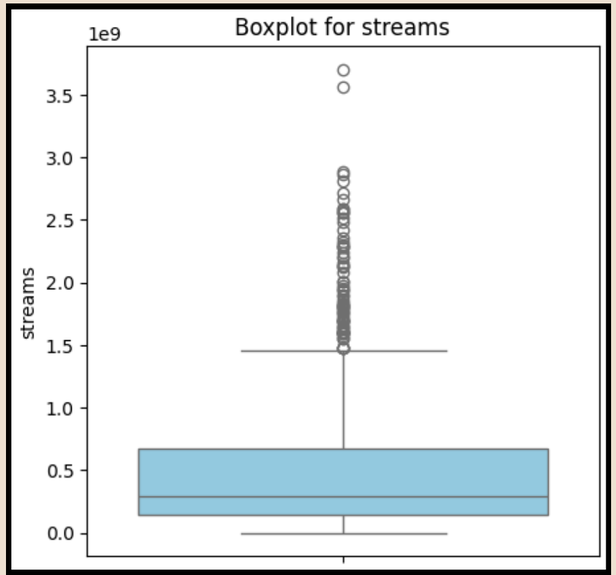
Box plots summarize the distribution of data, highlighting the median, variability, and outliers for easy comparison.











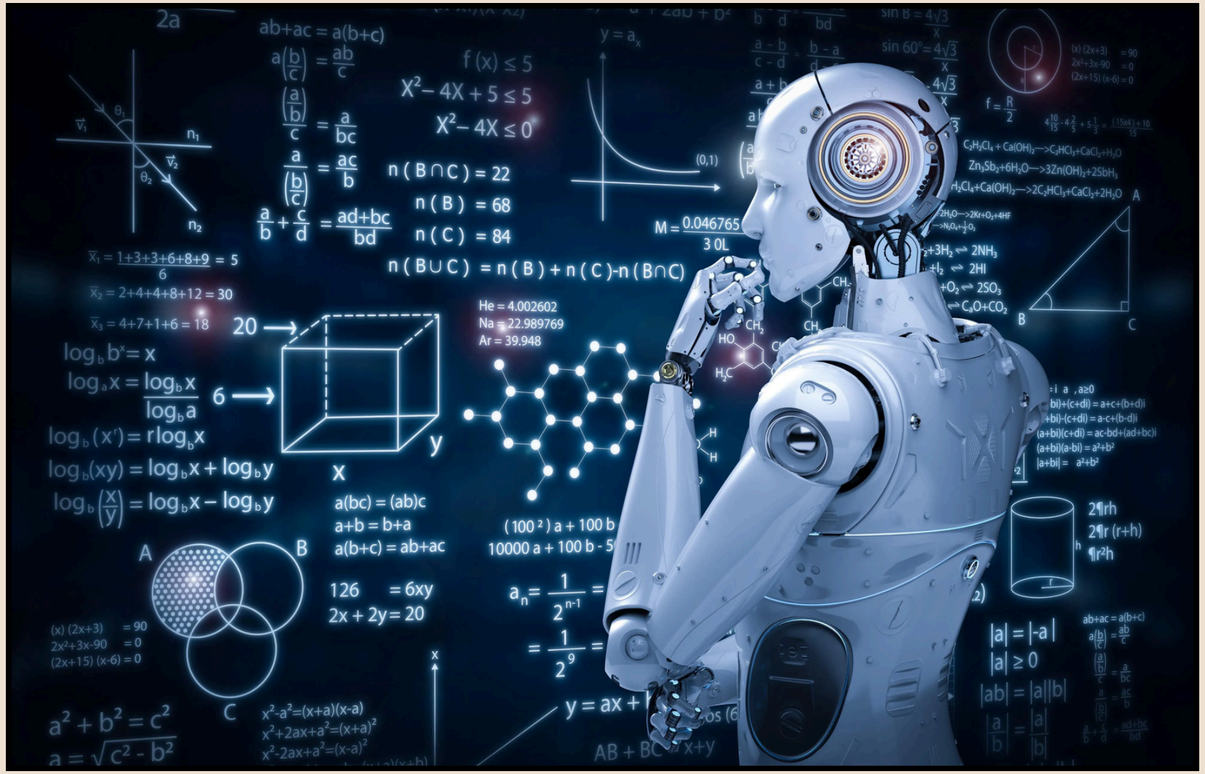
MULTI-COLLINEARITY CHECK

Multicollinearity occurs when independent variables in a model are too closely related or highly correlated , making it difficult to determine their individual effects on the dependent variable.

```
Removing feature: released_year (VIF: 116.5)
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
Removing feature: energy_% (VIF: 24.1)
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
Removing feature: danceability_% (VIF: 22.6)
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
Removing feature: bpm (VIF: 13.3)
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
/usr/local/lib/python3.11/dist-packages/statsmodels
  return 1 - self.ssr/self.uncentered_tss
Removing feature: track_id (VIF: 9.6)
Removing feature: in_spotify_playlists (VIF: 7.0)
Removing feature: artist_count (VIF: 6.5)
Removing feature: valence_% (VIF: 5.3)
```

```
Remaining features after multicollinearity removal:
               variables  VIF
0         released_month  3.8
1           released_day  3.4
2      in_spotify_charts  2.1
3                streams  4.4
4     in_apple_playlists  4.6
5        in_apple_charts  2.4
6    in_deezer_playlists  4.4
7       in_deezer_charts  1.4
8       in_shazam_charts  1.5
9                    key  3.1
10                  mode  2.2
11       acousticness_%  2.1
12   instrumentalness_%  NaN
13           liveness_%  2.5
14        speechiness_%  2.3
15            artist_id  3.3
```

MACHINE LEARNING ALGORITHMS



MACHINE LEARNING ALGORITHMS

Logistic Regression: A classification algorithm that models the probability of a binary outcome using a logistic function. It predicts probabilities and applies a threshold to classify data points.

K-Nearest Neighbors (KNN): A non-parametric algorithm that classifies data based on the majority class of the nearest neighbors. It calculates the distance between data points to make predictions.

Support Vector Machine (SVM): A classification algorithm that finds the optimal hyperplane to separate classes in a highdimensional space. It uses support vectors to maximize the margin between classes.

Decision Tree: A tree-like model that splits the data into branches based on feature values to predict outcomes. Each internal node represents a decision, and each leaf node represents a final class or value.

MACHINE LEARNING ALGORITHMS

Random Forest: An ensemble learning method that creates multiple decision trees and aggregates their results to improve accuracy. It reduces overfitting by averaging the predictions of individual trees.

AdaBoost: An ensemble technique that combines weak classifiers to form a strong classifier by assigning higher weights to misclassified instances. It iteratively adjusts the weights to improve prediction accuracy.

XGBoost: A gradient boosting algorithm that improves model performance by combining multiple weak learners. It is optimized for speed and efficiency, often used for structured/tabular data.

CatBoost : A high-performance gradient boosting algorithm developed by Yandex, optimized for categorical data and known for handling missing values efficiently.

MACHINE LEARNING ALGORITHMS

Ridge Regression : A linear regression technique that applies L2 regularization to reduce overfitting by penalizing large coefficients and improving model generalization

Bagging : A machine learning ensemble method that improves accuracy by training multiple models on bootstrapped subsets of data and averaging their predictions.

Artificial Neural Networks (ANNs) : Computational models inspired by the human brain, consisting of interconnected layers of neurons used for complex pattern recognition and deep learning tasks.

80-20 SPLIT

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
LOGISTIC REGRESSION	0.841	0.890
RIDGE REGRESSION	0.902	0.940
KNN	0.853	0.841
SVM	0.829	0.853
DECISION TREE	0.926	0.902
RANDOM FOREST	0.902	0.940
ADABOOST	0.902	0.902
GRADIENT BOOST	0.902	0.902
XG BOOST	0.902	0.902

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
CAT BOOST	0.902	0.878
BAGGING	0.902	0.890
ARTIFICIAL NEURAL NETWORKS	0.878	0.914

Model-1: Before VIF | Model-2: After VIF

75-25 SPLIT

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
LOGISTIC REGRESSION	0.854	0.912
RIDGE REGRESSION	0.912	0.912
KNN	0.805	0.854
SVM	0.844	0.864
DECISION TREE	0.941	0.893
RANDOM FOREST	0.902	0.903
ADABOOST	0.902	0.903
GRADIENT BOOST	0.902	0.903
XG BOOST	0.902	0.903

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
CAT BOOST	0.902	0.903
BAGGING	0.902	0.912
ARTIFICIAL NEURAL NETWORKS	0.893	0.883

Model-1: Before VIF | Model-2: After VIF

70-30 SPLIT

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
LOGISTIC REGRESSION	0.846	0.910
RIDGE REGRESSION	0.886	0.902
KNN	0.764	0.878
SVM	0.861	0.861
DECISION TREE	0.926	0.910
RANDOM FOREST	0.911	0.934
ADABOOST	0.911	0.910
GRADIENT BOOST	0.911	0.910
XG BOOST	0.911	0.910

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
CAT BOOST	0.911	0.910
BAGGING	0.911	0.870
ARTIFICIAL NEURAL NETWORKS	0.894	0.926

Model-1: Before VIF | Model-2: After VIF

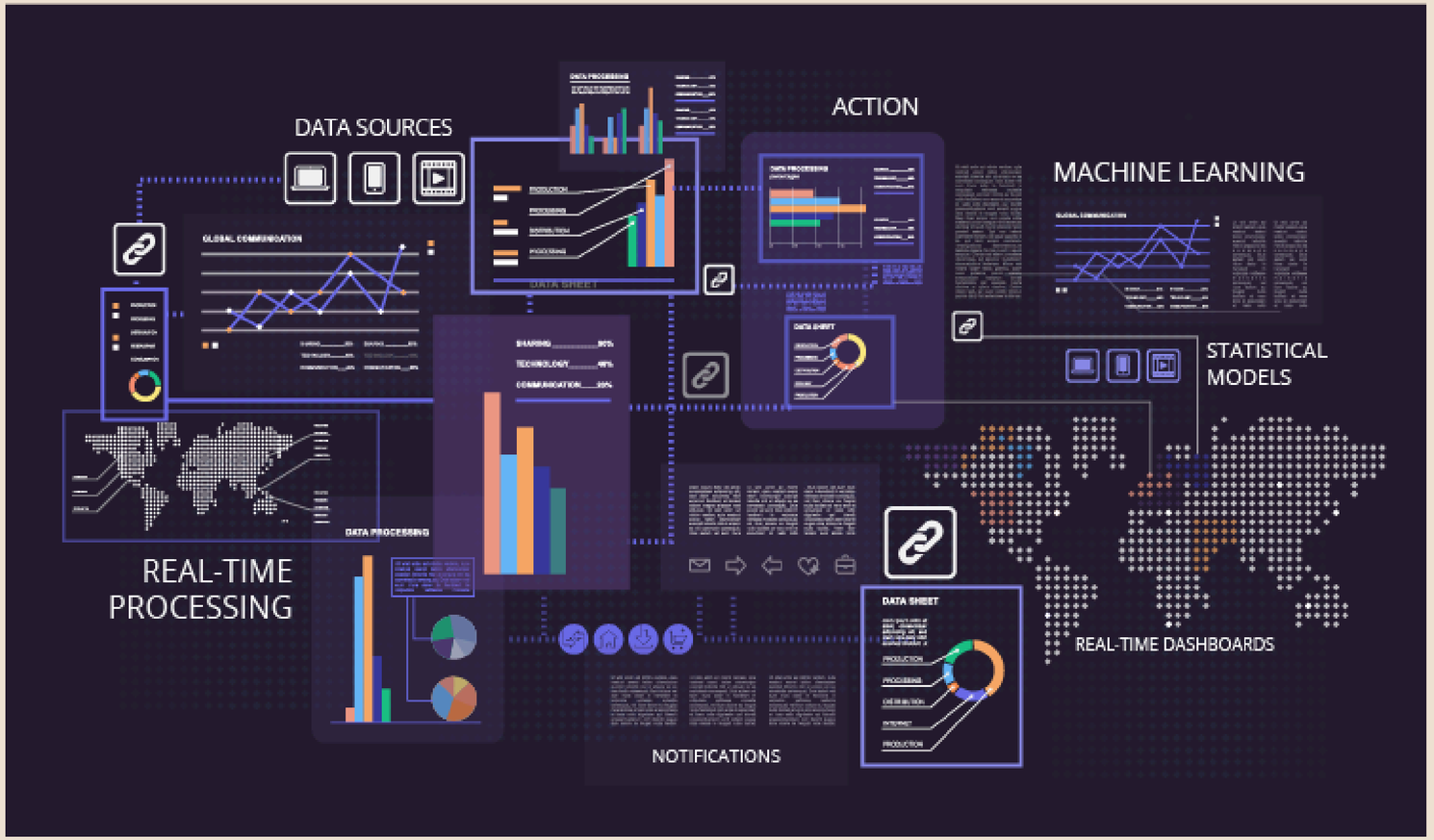
60-40 SPLIT

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
LOGISTIC REGRESSION	0.830	0.902
RIDGE REGRESSION	0.853	0.926
KNN	0.774	0.817
SVM	0.835	0.853
DECISION TREE	0.945	0.902
RANDOM FOREST	0.902	0.932
ADABOOST	0.902	0.902
GRADIENT BOOST	0.902	0.902
XG BOOST	0.902	0.902

ALGORITHMS	MODEL-1 Accuracy	MODEL-2 Accuracy
CAT BOOST	0.902	0.884
BAGGING	0.860	0.872
ARTIFICIAL NEURAL NETWORKS	0.945	0.940

Model-1: Before VIF | Model-2: After VIF

REAL-TIME PREDICTION



REAL-TIME PREDICTION

- **Our real-time song popularity prediction system analyzes an uploaded song file and predicts its potential success based on its unique audio characteristics.**
- **When a user uploads an MP3 file, the system automatically extracts key features.**
- **These extracted features are matched with patterns from our dataset, and a popularity score is calculated using our trained prediction model.**
- **Based on the computed score, the song is categorized as:**

1. **Low Popular (Under 0.4)**
2. **Moderately Popular (Between 0.4 and 0.7)**
3. **Highly Popular (Above 0.7)**

- **This AI-driven approach helps music producers, analysts, and streaming platforms understand a song's potential impact, making data-driven decisions more effective.**

REAL-TIME PREDICTION

```
Please upload the 'Blue.mp3' file
Choose Files No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving yung kai  blue Official Music Video.mp3 to yung kai  blue Official Music Video (2).mp3
Extracted Features: [[ 9.93840144e+01  1.26852268e+03  1.80033720e+03  2.40321030e+03
   5.19627871e-02  1.27780650e-05  1.56510632e-05 -1.60041626e+02
   1.45832565e+02  7.92200470e+00  2.09332981e+01  8.68595123e+00
  -2.72054940e-01 -4.21857786e+00 -6.55196190e+00 -6.00272274e+00
   9.47222888e-01 -9.76637304e-01  5.76532722e-01 -7.08096027e+00]]
Extracted Features: [[ 9.93840144e+01  1.26852268e+03  1.80033720e+03  2.40321030e+03
   5.19627871e-02  1.27780650e-05  1.56510632e-05 -1.60041626e+02
   1.45832565e+02  7.92200470e+00  2.09332981e+01  8.68595123e+00
  -2.72054940e-01 -4.21857786e+00 -6.55196190e+00 -6.00272274e+00
   9.47222888e-01 -9.76637304e-01  5.76532722e-01 -7.08096027e+00]]
Predicted Popularity Score for "Blue" by Yung Kai: [0.83868346]
Popularity Category: Highly Popular
```

```
Please upload the 'Birds of a Feather.mp3' file
Choose Files No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving Billie Eilish  BIRDS OF A FEATHER Official Music Video.mp3 to Billie Eilish  BIRDS OF A FEATHER Official Music Video (3).mp3
Extracted Features: [ 1.03359375e+02  1.68858705e+03  2.27329464e+03  3.97847855e+03
  4.82545431e-02 -4.21629420e-06  1.70529495e-06 -1.39110245e+02
  9.39605637e+01  2.43194103e+01  2.69064674e+01  7.80438185e+00
  1.28665638e+01  3.15756130e+00  5.19996405e-01  2.40761256e+00
  3.82024884e+00  4.84614468e+00  3.52545166e+00 -7.17523003e+00]
Predicted Popularity Score for "Birds of a Feather" by Billie Eilish: [0.84538815]
Popularity Category: Highly Popular
```

REAL-TIME PREDICTION

```
<ipython-input-47-09a33cbdbdae>:9: UserWarning: PySoundFile failed. Trying audioread instead.
  y, sr = librosa.load(audio_path)
/usr/local/lib/python3.11/dist-packages/librosa/core/audio.py:184: FutureWarning: librosa.core.audio.__audioread_load
        Deprecated as of librosa version 0.10.0.
        It will be removed in librosa version 1.0.
  y, sr_native = __audioread_load(path, offset, duration, dtype)
Extracted Features: [ 7.38281250e+01  1.88412316e+03  2.27968011e+03  4.31382005e+03
  6.88029124e-02 -2.29521338e-05 -2.53455219e-05 -1.13385040e+02
  1.02587097e+02  1.43517189e+01  3.91144538e+00  8.90279961e+00
  1.04355850e+01 -3.30585694e+00  1.68036234e+00 -3.46079612e+00
  4.62371254e+00 -1.90318978e+00  1.26805186e+00 -7.31377840e+00]
Predicted Popularity Score for "Love Me Again" by V: 0.7969816050135103
Popularity Category: Highly Popular
```

```
Please upload the 'Kabira.mp3' file
Choose Files No file chosen    Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.
Saving _Kabira Full Song_ Yeh Jawaani Hai Deewani _ Pritam _ Ranbir Kapoor, Deepika Padukone [ ezmp3.cc ].mp3 to _Kabira Full Song_ Yeh Jawaani H
Extracted Features: [ 8.61328125e+01  1.77846622e+03  2.22037746e+03  3.82968475e+03
  7.16578678e-02 -8.67706547e-07 -4.54700312e-05 -1.70235153e+02
  1.14973183e+02  2.77608204e+00  2.05547924e+01  9.30363369e+00
  3.28663754e+00  2.11672115e+00  1.85090911e+00 -1.58632803e+00
  5.03596067e+00 -3.00546765e+00  4.52584743e+00 -2.75342989e+00]
Predicted Popularity Score for "Kabira" from Yeh Jawaani Hai Deewani: 0.8148676451903529
Popularity Category: Highly Popular
```

SUMMARY

- The goal of the project was to classify a song based on its popularity into 3 categories- Low popular, Moderately Popular and Highly Popular.
- A 60-40 split was identified as optimal and offering a perfect balance between sufficient training data for model learning and reliable evaluation on unseen data.
- For the given dataset, the Artifical Neural Networks Algorithm demonstrated the highest accuracy of 94.5%, outperforming other models
- However, after removing high multicollinearity variables, the Artificial Neural Networks Algorithm delivered the best results with an accuracy of 94%, highlighting its ability to adapt effectively to reduced feature sets.
- The real-time prediction system successfully extracts audio features from any song and classifies its popularity instantly, demonstrating practical applicability in music analytics and streaming insights.

FUTURE SCOPE

- ***Integration with Streaming Platforms*** **– The model can be integrated with Spotify, Apple Music, or YouTube Music to analyze real-time streaming trends and improve recommendation systems.**
- ***Artist & Label Insights*** **– Helping music producers and record labels evaluate a song's potential success before release, optimizing marketing strategies.**
- ***Multilingual & Genre-Specific Models*** **– Training models for different languages and genres (e.g., K-pop, Indie, EDM) to provide customized popularity predictions.**
- ***Real-Time Music Trend Prediction*** **– Expanding the model to predict future popularity trends based on historical data and current listening patterns.**

THANK YOU!!

COLAB NOTEBOOK

Presented By :

VENKAT ROHAN SEETEPALLI