Song Popularity Prediction

SC_23

Department	Section	ID	Name
SC	Sec 5	2021170423	مارك جمال انور فهمي
SC	Sec 7	2021170557	مهند أحمد عبد الحميد محمد
SC	Sec 5	2021170442	مايكل نادر عماد الدين
SC	Sec 7	2021170536	مكاريوس موسي سليمان موسي
SC	Sec 8	2021170650	يوسف ناصر محمد علي

Introduction

The objective of our project is to develop a predictive model that can forecast the popularity of songs on Spotify. To achieve this, we will leverage a comprehensive dataset containing various features related to each song. These features encompass details such as song attributes, information, and specific metrics from the Hot 100 chart. Here is an explanation of the features:

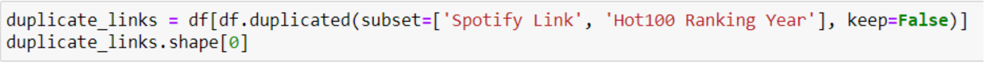
- Song Information: Including song title, album name, album release date, artist name, and genre.
- Song Attributes:
- Song Length: Duration of the song in milliseconds.
- Loudness: Volume and intensity of the song.
- Acousticness: Proportion of real instruments used in the song versus electronic sounds.
- Danceability: Measure indicating the ease of dancing to the song.
- Energy: Represents the intensity and activity of the song.
- Instrumentalness: Indicates the presence of vocals versus instrumental components.
- Liveness: Likelihood of the song being a live performance based on audience sounds.
- Speechiness: Presence of spoken words or speech-like elements.
- Tempo: Speed of the song measured in beats per minute (BPM).
- Valence: Indicates the positivity conveyed by the song, such as cheerfulness.
- Key: Musical key or tonality influencing the song's mood.
- Time Signature: Defines the rhythmic structure of the song by specifying the number of beats per bar and the note type.
- Hot100 Ranking Year: The year when the song achieved its Billboard Hot 100 ranking.
- Hot100 Rank: The song's specific ranking on the Billboard Hot 100 chart for a given year.

With this dataset, we aim to implement and fine-tune a predictive model that can effectively identify the key factors contributing to a song's popularity on Spotify.

Data Preprocessing

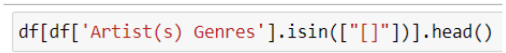
- Initial exploration using functions like head(), info(), and describe() provided an overview of the data structure, column names, data types, and summary statistics.
- Missing Values: The dataset was examined for missing values using functions like ‘isnull().sum()’ and ‘duplicated().sum()0’. While no obvious nulls or fully duplicated rows were found, there were indeed hidden nulls and duplicates, which we investigated using the following techniques.
- In the dataset, many songs had duplicate entries, although not all of their links were duplicated. When a link appeared more than once but with different Hot 100 ranking years, it was considered potentially non-problematic, as the song could have been featured in multiple significant song lists over various years. However, when the Hot 100 ranking years were the same for duplicated links, it posed a challenge for the model. This situation was unusual, as a song typically shouldn't have multiple rankings in the same year. Rows where both the link and

Hot 100 ranking year were duplicated were investigated. Given that there were only 12 such rows, they were removed from the dataset.



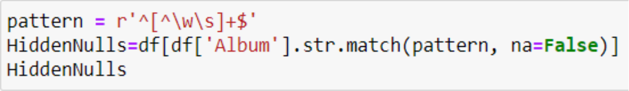
```
duplicate_links = df[df.duplicated(subset=['Spotify Link', 'Hot100 Ranking Year'], keep=False)]
duplicate_links.shape[0]
```

- Hidden missing values in 'Artist(s) Genres' (‘[]’) were replaced with the mode value of the column, ensuring data completeness without compromising the integrity of the dataset.



```
df[df['Artist(s) Genres'].isin(["[]"])].head()
```

- Hidden nulls in the 'Album' column (‘?’), identified using regex patterns, were deleted to maintain data consistency as they were only 8 rows.



```
pattern = r'^[^\w\s]+$'
HiddenNulls=df[df['Album'].str.match(pattern, na=False)]
HiddenNulls
```

EDA

The df.describe() function was used to extract valuable insights from the dataset:

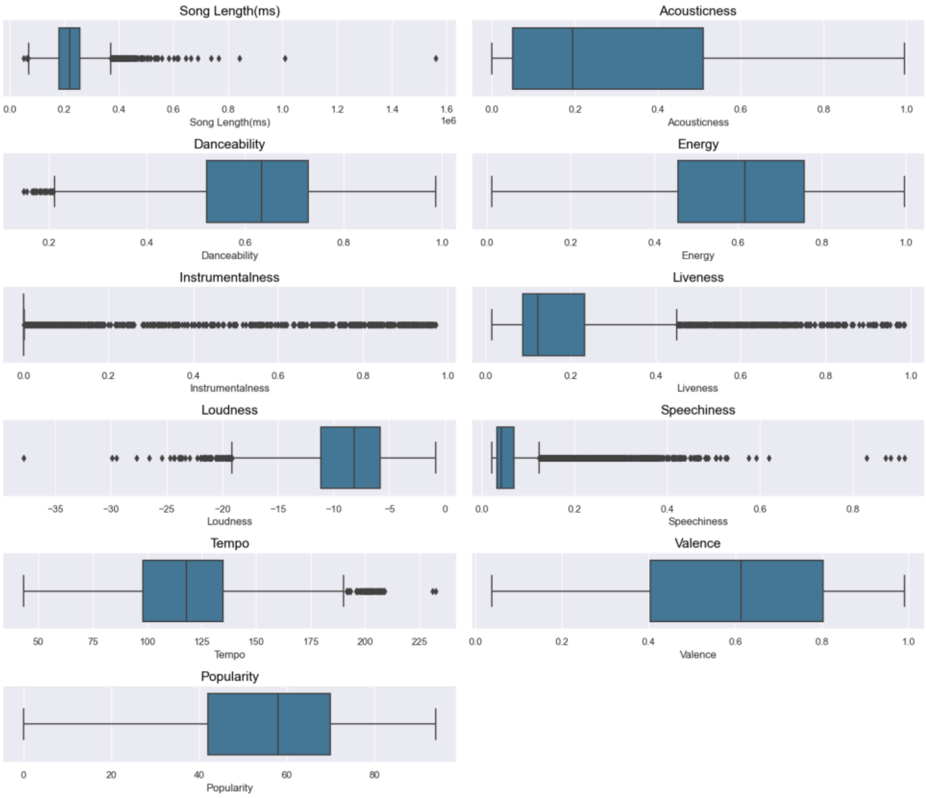
- Hot100 Ranking Year: The ranking years span from 2010 to 2020.
- Hot100 Rank: The average rank is 50.5, and there are no missing values.
- Popularity: The average popularity score stands at 66.4.
- Energy: Energy values range from a minimum of 0.0112 to a maximum of 0.996.
- Instrumentalness: The mean instrumentalness value is 0.117.
- Liveness: Liveness values vary between 0.0158 and 0.995.
- Loudness: The loudest song has a loudness value of 5.787, while the quietest is at -14.874.
- Acousticness: Songs with an acousticness score of 0.8 or higher are predominantly acoustic.
- Danceability: Danceability scores exhibit an approximately normal distribution, slightly skewed to the right.
- Key: The key scores have an approximately uniform distribution.
- Speechiness: Speechiness scores are right-skewed in their distribution.
- Tempo: Songs with a tempo score of 120 or higher are considered fast-paced.
- Valence: Valence scores range from a minimum of 0.0337 to a maximum of 0.982.
- Time Signature: The average time signature is 4.0.
- Mode: The dataset contains two modes: "Major" and "Minor". The prevalent mode is "Major", appearing 61 times in the dataset.

Insights from Exploring Unique Values and Value Counts

- Most Prolific Artists: 'The Karaoke Channel' tops the list with a whopping 42 songs, showcasing their significant contribution to the dataset. Close behind are iconic artists like 'Madonna' and 'Janet Jackson', with notable song counts of their own.

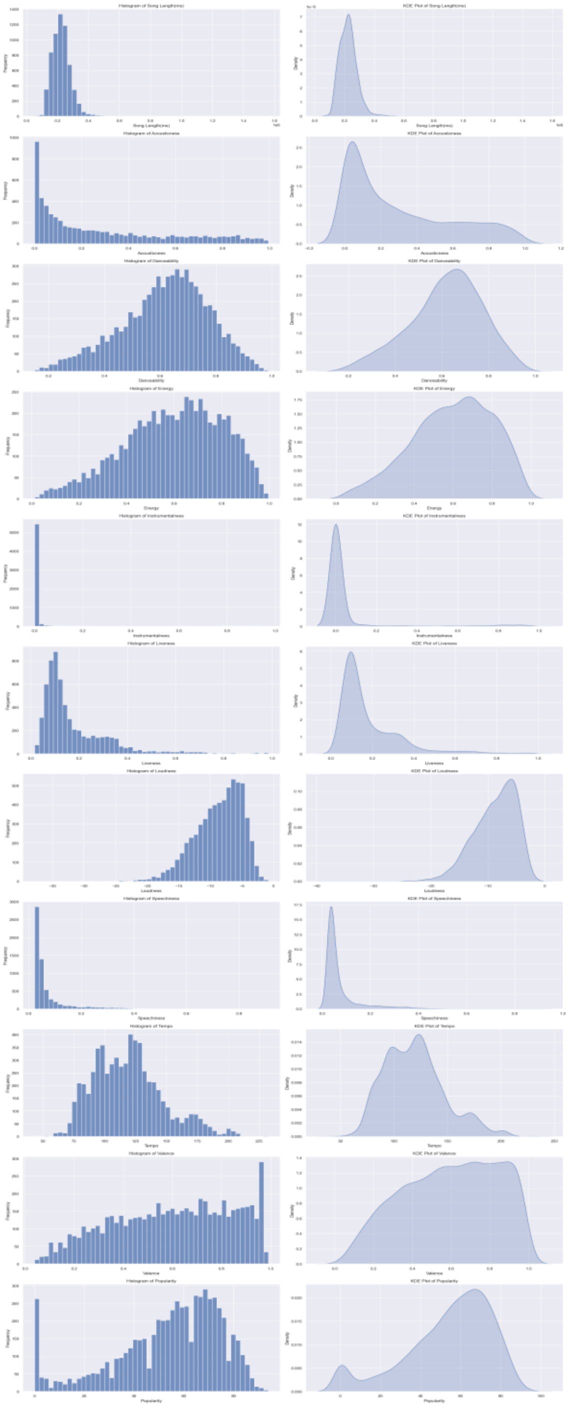
- Prolific Albums: The album 'Greatest Hits' stands out as the most prolific with an impressive 48 entries. Following closely behind are 'Super Hits' and '16 Most Requested Songs', each boasting their own respectable counts.
- Popularity Rankings: The majority of songs in the dataset have rankings ranging from mid to high 90s. Notably, the years 2017 and 1974 emerge as standout years with higher song rankings. This suggests that these particular years might have witnessed a surge in popular or significant songs.
- Earlier Years: In contrast, earlier years such as 1955, 1950, and 1952 exhibit lower rankings. This could imply either a scarcity of popular songs during these years or potentially a smaller dataset representation for those specific years.
- There are over 1760 rows, accounting for more than 28% of the data, with a 'rank year' of 0 even before the song's release year. This suggests that the 'rank year' column is largely inaccurate.

Boxplots were employed to visualize the distribution of the data and identify any outliers. They provided a clear graphical representation of the data's central tendency, spread, and potential outliers.



- Outliers were found in most of the features; however, they were left for the feature engineering phase to determine whether they were normal or abnormal before any removal.

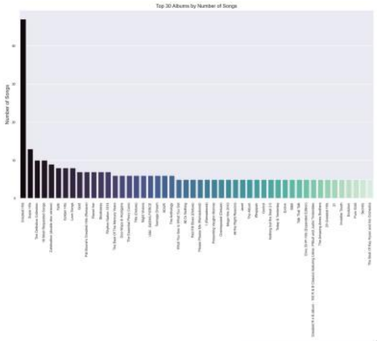
Histplots and **kdeplots** were utilized to examine the distributions of the numerical features in the dataset. Histplots displayed the frequency distribution of a single variable, while kdeplots provided a smoothed representation of the data's distribution.

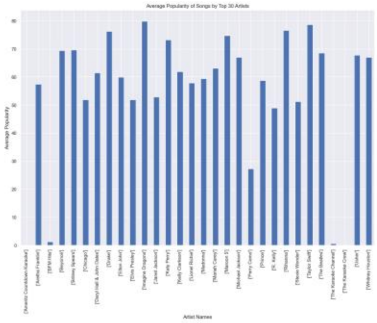


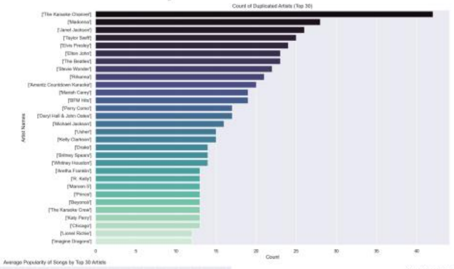
Many features exhibited right-skewness, indicating non-normal distribution. Despite this, they were retained for the feature engineering phase to undergo appropriate transformations.

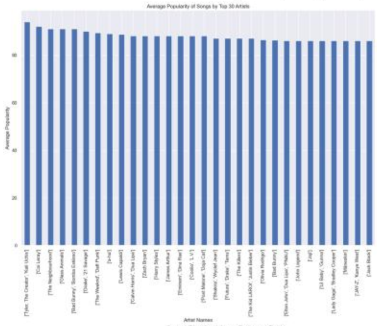
Count plots:

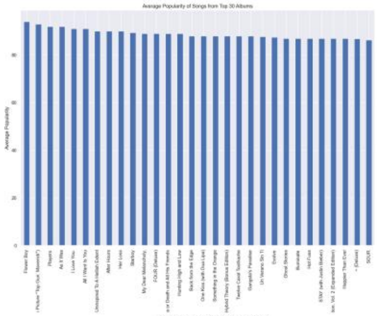
- Top Albums: Highlighting albums with the most songs.
- Frequent Years: Showing years with the highest song counts.
- Popular Albums: Identifying albums with the highest popular songs.
- Top Artists: Revealing artists with multiple popular songs.
- Average Popularity: Calculating the average popularity score for the top 30 artists.
- These plots offer insights into album, year, and artist distributions and their respective popularities in the dataset.

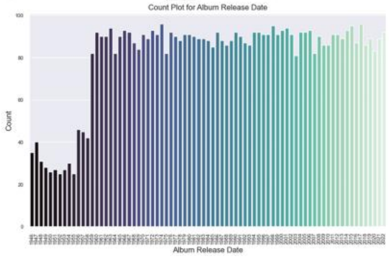


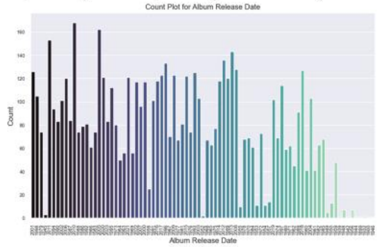






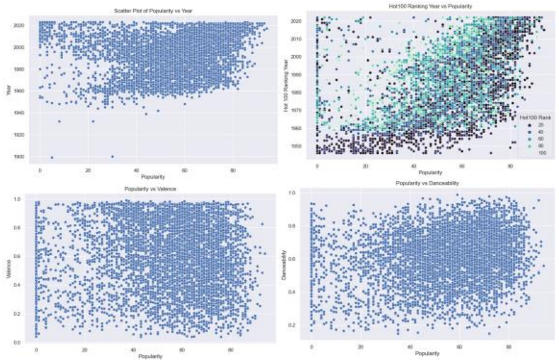


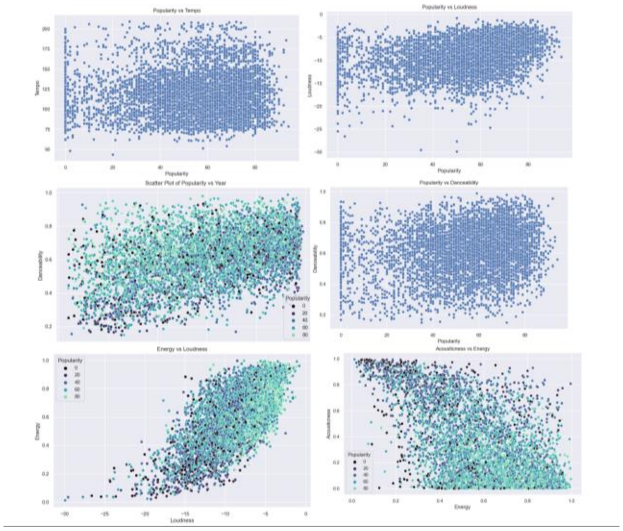




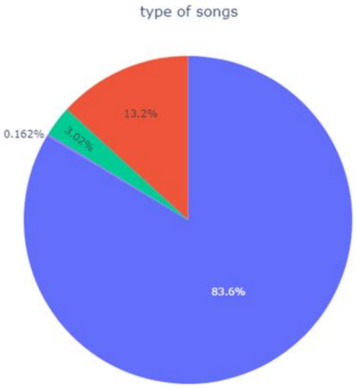
Scatterplot Insights:

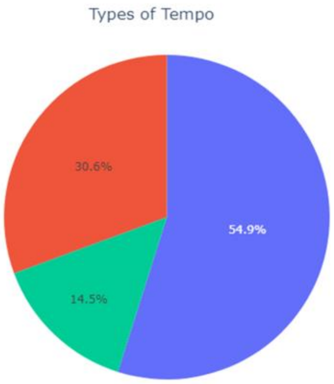
- Hot Ranking Year & Popularity: A direct relationship exists between the hot ranking year and popularity.
- Valence & Popularity: No clear relationship is observed between valence and popularity, loudness, or tempo.
- Valence & Danceability: Valence and danceability show a moderate direct relationship; higher danceability often corresponds to higher popularity.
- Energy, Loudness & Popularity: Energy and loudness are positively correlated, with higher levels indicating increased popularity.
- Acousticness & Energy: An inverse relationship is seen between acousticness and energy; lower acousticness tends to correlate with higher popularity.
- Song Length & Popularity: No direct correlation exists between song length and popularity, but songs with excessive length generally have lower popularity.

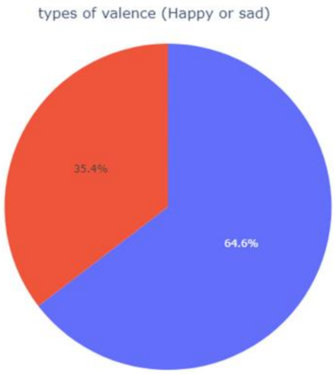




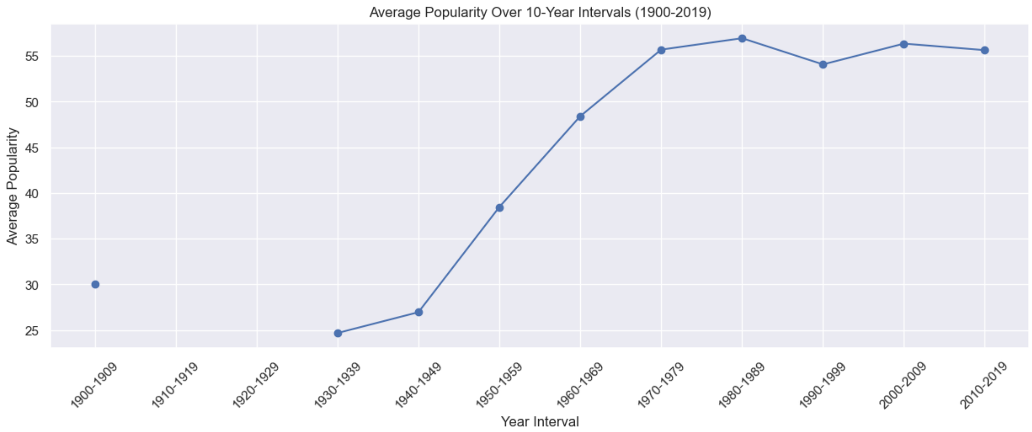
Pie Charts:







- Speechiness: Most of the dataset has very low speechiness.
- Tempo & Speed: Over 54% of the data indicates songs that are neither too fast nor too slow. Fast songs with high tempo are twice as common as slow songs.
- Mood & Popularity: More than 64% of the dataset comprises sad songs. Interestingly, sad songs seem to have higher popularity.



Higher popularities tend to be in more recent years. As we look further back in time, the popularities of songs generally decrease.

Feature Engineering

Encoding

To enhance the utility of the 'artist name' and 'artist genres' columns, which contained lists of strings, the following data transformations were conducted:

1. **Splitting Lists into Individual Rows**:
   - Each list within the 'artist name' and 'artist genres' columns was divided into multiple rows, with each element assigned to its own row.

2. **Encoding Categorical Data**:
   - Upon splitting, both **target** and **label** encoding techniques were applied to the transformed features. Target encoding converted categorical values into numerical representations based on the target variable, while label encoding allocated a distinct numerical value to each category.

3. **Combining Encoded Value**s:
- Encoded values were aggregated back into a single cell for lists containing more than one string by summing them. This approach maintained the crucial information while reducing data redundancy.

Creating new features:

Two new features were introduced to the dataset: 'Hype' and 'Happiness'.

The 'Hype' feature is calculated as the sum of 'Loudness' and 'Energy'.
The 'Happiness' feature represents the sum of 'Danceability' and 'Valence'.

Handling Outliers:

Upon investigating the outliers, it was observed that all outliers represented normal and genuine values. No abnormal or unusual values were identified, and their presence was not found to adversely impact the model. Therefore, the decision was made to retain these outliers in the dataset.

Removing unnecessary columns:

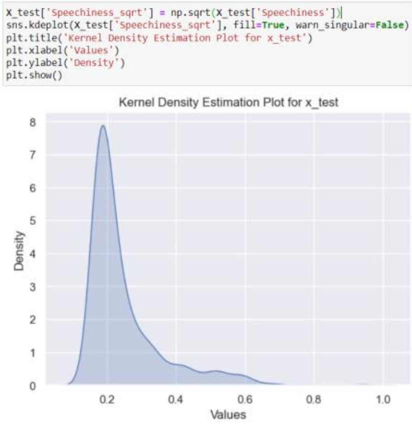
- The following columns were dropped from the dataset as they were deemed not useful for our predictive model: ['Song', 'Album', 'Spotify Link', 'Song Image', 'Spotify URI']

Splitting the data:

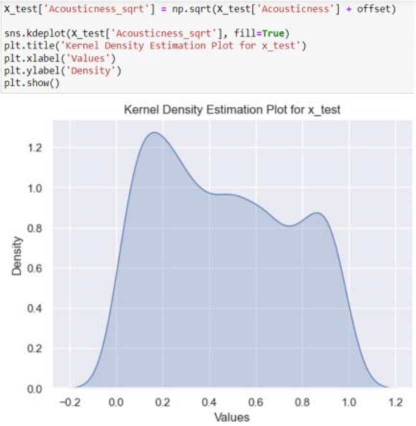
- Before performing any feature scaling or transformations to avoid data leakage, the dataset was split into 80% for training and 20% for testing.
- Although a separate validation set was not created, k-fold cross-validation was utilized during the modeling process to ensure robustness and assess the model's generalization performance.

Transformations:

- To correct the skewed distributions, a log transformation was applied to the 'Liveness' feature, a square root transformation to the 'Acousticness' feature, and a square root transformation to the 'Speechiness' feature.

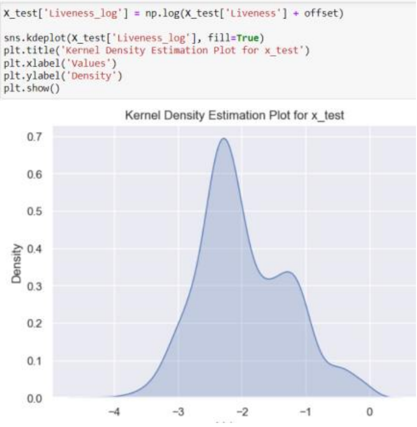


```
X_test['Speechiness_sqrt'] = np.sqrt(X_test['Speechiness'])
sns.kdeplot(X_test['Speechiness_sqrt'], fill=True, warn_singular=False)
plt.title('Kernel Density Estimation Plot for x_test')
plt.xlabel('Values')
plt.ylabel('Density')
plt.show()
```



```
X_test['Acousticness_sqrt'] = np.sqrt(X_test['Acousticness'] + offset)

sns.kdeplot(X_test['Acousticness_sqrt'], fill=True)
plt.title('Kernel Density Estimation Plot for x_test')
plt.xlabel('Values')
plt.ylabel('Density')
plt.show()
```



```
X_test['Liveness_log'] = np.log(X_test['Liveness'] + offset)

sns.kdeplot(X_test['Liveness_log'], fill=True)
plt.title('Kernel Density Estimation Plot for x_test')
plt.xlabel('Values')
plt.ylabel('Density')
plt.show()
```

Scaling:

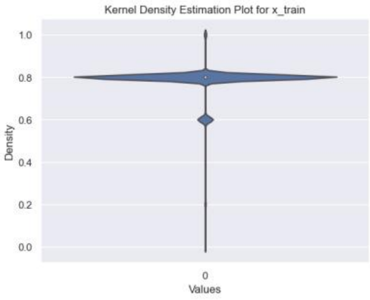
- During the scaling process, the dataset was initially partitioned to avoid data leakage. Min-Max scaling was employed as it useful when data is not Gaussian, while Standard scaling was not utilized. Standard scaling is typically applied to normally distributed data, making Min-Max scaling the preferred choice for this dataset.

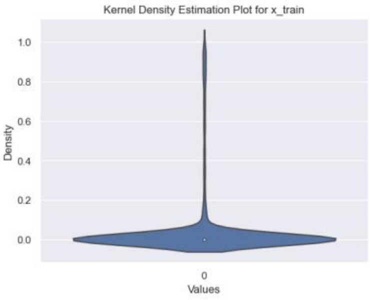
Feature Selection

In the feature selection phase, two primary methods were employed: the correlation matrix and filter methods using kBest.

Instrumentalness & Time Signature:

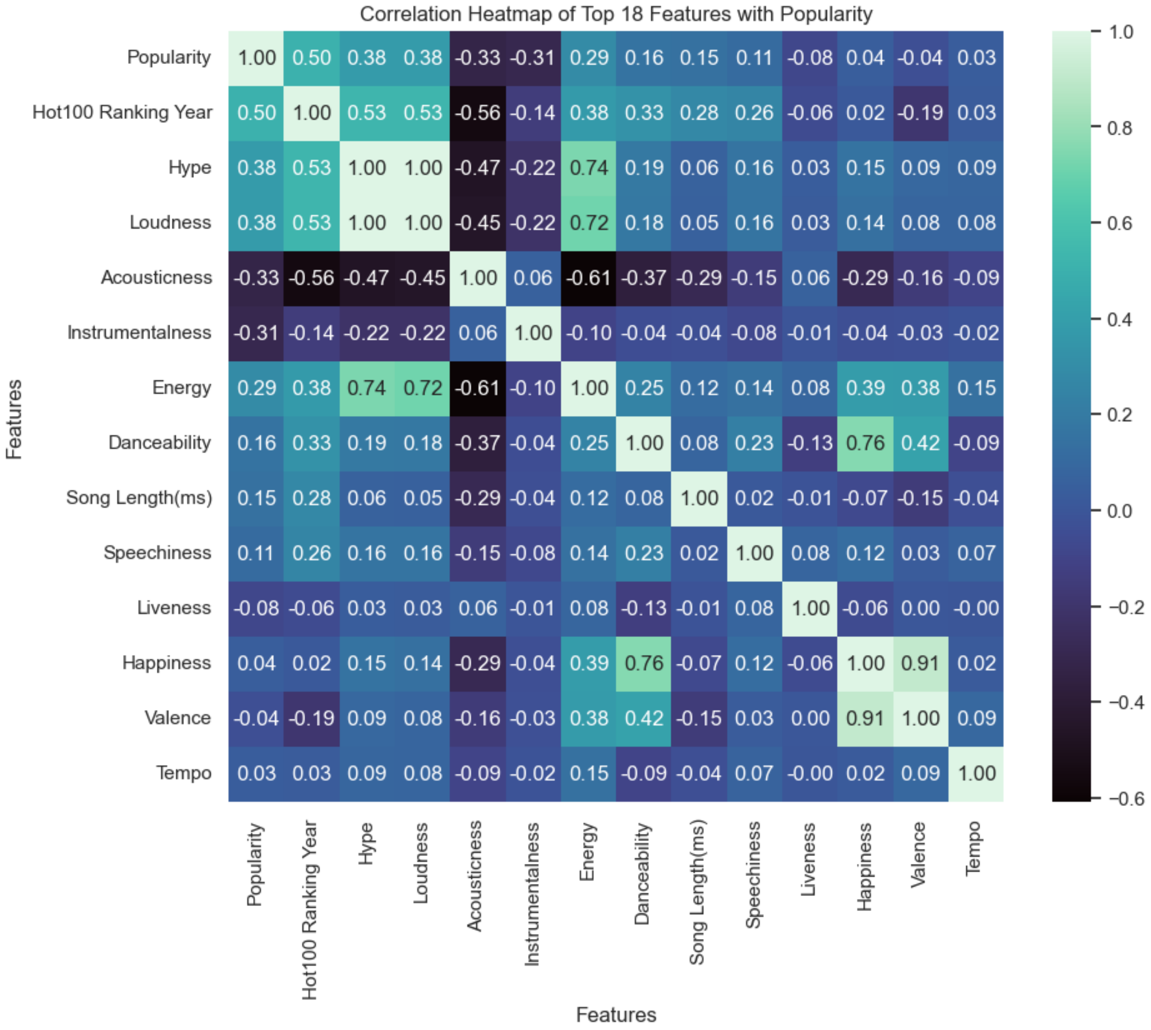
Instrumentalness and Time Signature were removed from the dataset due to their negligible variance, indicating that they would not significantly influence the model. Specifically, over 99% of the Time Signature values were '4', and nearly half of the Instrumentalness values were zeros.





Correlation Matrix Heatmap:

For the correlation matrix approach, the data was first split to only include continuous features. Subsequently, a correlation heatmap was generated to identify relationships between these features. The obtained output from this analysis is as follows:



Correlation Heatmap of Top 18 Features with Popularity

	Popularity	Hot100 Ranking Year	Hype	Loudness	Acousticness	Instrumentalness	Energy	Danceability	Song Length(ms)	Speechiness	Liveness	Happiness	Valence	Tempo
Popularity	1.00	0.50	0.38	0.38	-0.33	-0.31	0.29	0.16	0.15	0.11	-0.08	0.04	-0.04	0.03
Hot100 Ranking Year	0.50	1.00	0.53	0.53	-0.56	-0.14	0.38	0.33	0.28	0.26	-0.06	0.02	-0.19	0.03
Hype	0.38	0.53	1.00	1.00	-0.47	-0.22	0.74	0.19	0.06	0.16	0.03	0.15	0.09	0.09
Loudness	0.38	0.53	1.00	1.00	-0.45	-0.22	0.72	0.18	0.05	0.16	0.03	0.14	0.08	0.08
Acousticness	-0.33	-0.56	-0.47	-0.45	1.00	0.06	-0.61	-0.37	-0.29	-0.15	0.06	-0.29	-0.16	-0.09
Instrumentalness	-0.31	-0.14	-0.22	-0.22	0.06	1.00	-0.10	-0.04	-0.04	-0.08	-0.01	-0.04	-0.03	-0.02
Energy	0.29	0.38	0.74	0.72	-0.61	-0.10	1.00	0.25	0.12	0.14	0.08	0.39	0.38	0.15
Danceability	0.16	0.33	0.19	0.18	-0.37	-0.04	0.25	1.00	0.08	0.23	-0.13	0.76	0.42	-0.09
Song Length(ms)	0.15	0.28	0.06	0.05	-0.29	-0.04	0.12	0.08	1.00	0.02	-0.01	-0.07	-0.15	-0.04
Speechiness	0.11	0.26	0.16	0.16	-0.15	-0.08	0.14	0.23	0.02	1.00	0.08	0.12	0.03	0.07
Liveness	-0.08	-0.06	0.03	0.03	0.06	-0.01	0.08	-0.13	-0.01	0.08	1.00	-0.06	0.00	-0.00
Happiness	0.04	0.02	0.15	0.14	-0.29	-0.04	0.39	0.76	-0.07	0.12	-0.06	1.00	0.91	0.02
Valence	-0.04	-0.19	0.09	0.08	-0.16	-0.03	0.38	0.42	-0.15	0.03	0.00	0.91	1.00	0.09
Tempo	0.03	0.03	0.09	0.08	-0.09	-0.02	0.15	-0.09	-0.04	0.07	-0.00	0.02	0.09	1.00

Hot 100 ranking year, Hype, Loudness, Acousticness, Energy were some of the continuous features that correlated with the target variable, Popularity. Following these, Danceability, Length, and Speechiness also showed correlation.

However, Liveness, Happiness, Valence, and Tempo did not exhibit strong correlation with Popularity, suggesting they will most likely be removed from the feature set.

Filter methods (SelectKBest):

SelectKBest was utilized to reduce the dimensionality of the model by selecting optimal features. Initially, the features were categorized into continuous and categorical sets. For the categorical features, a subset was chosen using ANOVA due to its compatibility with numerical output. Meanwhile, for the continuous features, subsets were selected using both Pearson and Spearman correlation methods, given their suitability for numerical input-output relationships.

- The categorical features, including ‘Hot100 Ranking Year',' Artist Names', 'Artist(s) Genres', 'Year', 'Hot100 Rank', 'Key', 'Mode', and 'Time Signature', were split to perform ANOVA (f_regressor). The resulting scores from the ANOVA analysis are as follows:

Feature	Score
Hot100 Ranking Year	1685.089278
Artist(s) Genres	589.849021
Mode	59.832321
Year	43.159868
Hot100 Rank	38.086366
Time Signature	32.804640
Artist Names	2.609181
Key	2.274452

The continuous features, including 'Acousticness', 'Loudness', ‘Hype’, ‘Happiness’, 'Energy', 'Danceability', 'Instrumentalness', 'Song Length(ms)', 'Speechiness', 'Liveness', 'Valence', and 'Tempo', were split to perform Spearman correlation analyses. Here is the output:

Feature	Score
Loudness	0.411262
Hype	0.409064
Acousticness	0.309086
Energy	0.262168
Instrumentalness	0.211925
Song Length(ms)	0.201803
Danceability	0.156096
Speechiness	0.152260
Valence	0.076781

Liveness	0.043025
Tempo	0.030857
Happiness	0.009169

Selected Features:

- Acousticness
- Artist(s) Genres
- Hot100 Ranking Year
- Mode
- Year
- Hot100 Rank
- Song Length(ms)
- Hype

The features listed in the table exhibited the highest scores in both the Pearson correlation matrix and the SelectKBest selection using both ANOVA and Spearman methods**.**

Modeling and Hyperparameter tuning

Model Optimization

1. Hyperparameter Tuning

Hyperparameter tuning was conducted using grid search, which systematically combined all possible combinations of hyperparameters provided. This approach aimed to identify the optimal hyperparameter values that yielded the best-performing models in terms of accuracy. The tuning process significantly enhanced the accuracy of the models by optimizing the hyperparameters.

2. Cross-Validation

Cross-validation was employed to assess the generalization capability of the models and mitigate overfitting. This provided a more reliable estimate of a model's performance by averaging the evaluation results across multiple validation sets, thereby offering a more robust evaluation metric compared to a single train-test split.

Evaluation Metrics

To evaluate the performance of the optimized models, the following metrics were used:

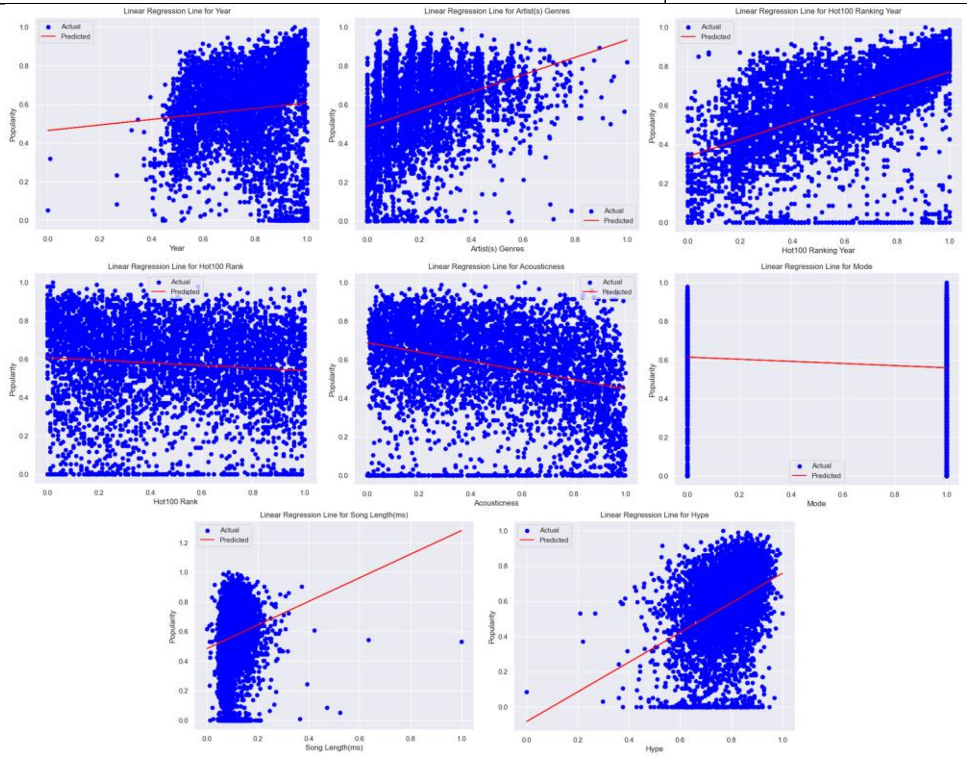
- Mean Cross-Validation Score
- Mean Squared Error (MSE)
- Root Mean Squared Error (RMSE)
- Mean Absolute Error (MAE)
- R^2 Score

Models & Results

1. Linear Regression:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

Mean Cross-Validated MSE	0.03064286084857142
Training Set Mean Squared Error (MSE)	0.030530084172362366
Test Mean Squared Error (MSE)	0.030767509891288524
Root Mean Squared Error	0.1754066985359696
Mean Absolute Error	0.12796408516909635
R-squared Score	0.42652560110677795



1. Support Vector Machine (SVR) Regression:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

Mean cross-validation error	0.02679188011142495
Test Mean squared error	0.02364065098569238
Root mean squared error	0.15375516572035028
Mean Absolute Error	0.11139626870966297

R-squared score	0.5593628421225276

2. Decision Tree Regressor:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

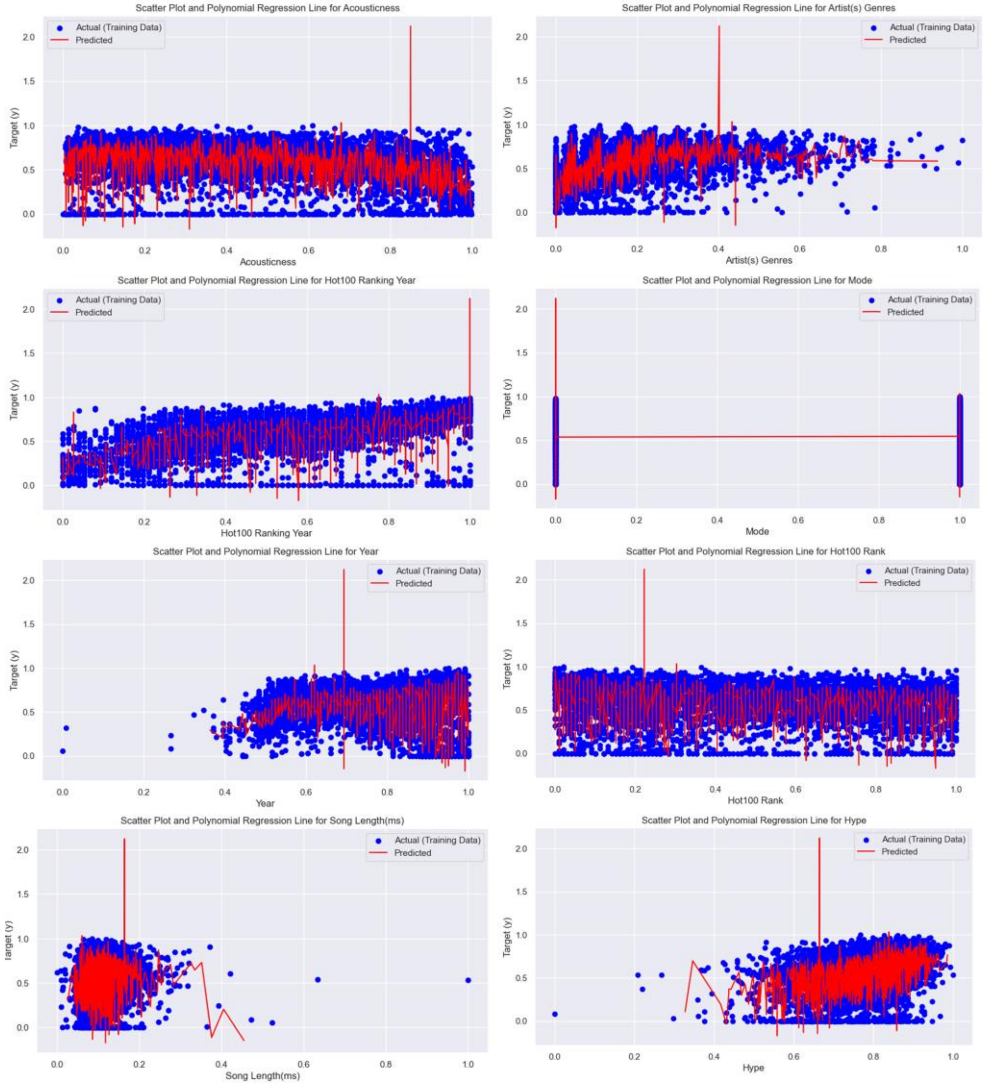
Mean cross-validation error	0.02300366021603647
Mean squared error	0.022309474807185516
Root Mean Squared Error	0.14936356586258082
Mean Absolute Error	0.10894726894194527
R-squared score	0.5841745822174368

3. Polynomial Regression:

Degree = 3

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

Mean squared error (CV)	0.025753112894161535
Test Mean squared error	0.023946517278389555
Root Mean Squared Error	0.15474662283355187
Mean Absolute Error	0.11240978380328935
R-squared score	0.5536618124010461



Degree = 2

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

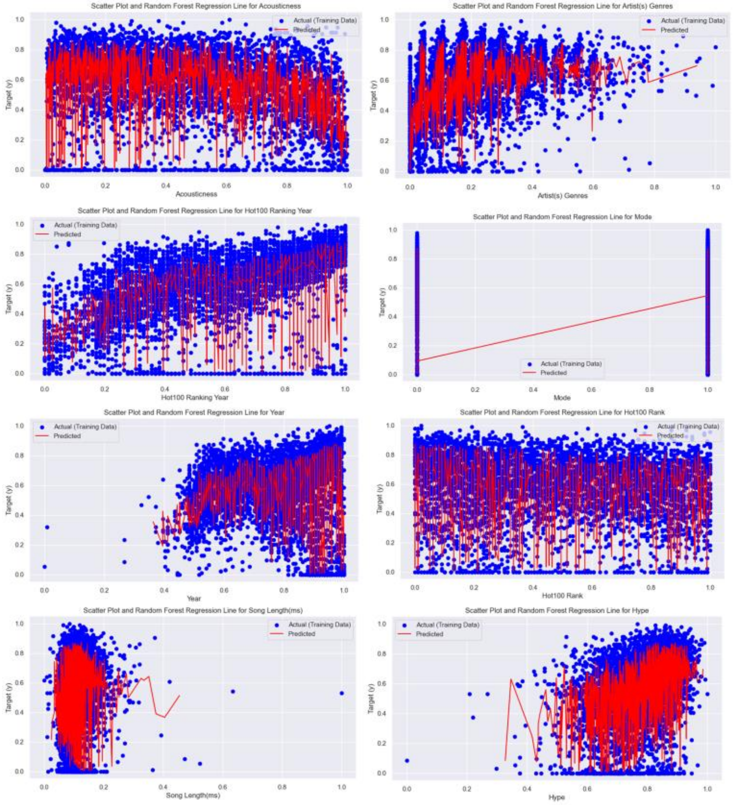
Mean squared error (CV)	0.027041647374290288
Test Mean squared error	0.02581685892686747
Root Mean Squared Error	0.16067625501880317
Mean Absolute Error	0.12014215938373975
R-squared score	0.5188005884548884

4. Random Forest Regressor:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

Mean cross-validation error	0.019599309537988087
Mean squared error	0.018050108976868182
Root Mean Squared Error	0.13435069399473967

Mean Absolute Error	0.09869859715548565
R-squared score	0.663564733316378



5. XGBoost Regression:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (16%) of training data.

Mean cross-validation error	0.01951636057772324
Mean squared error	0.01802356187746808
Root Mean Squared Error	0.13425185986595523
Mean Absolute Error	0.09999520046725531
R-squared score	0.6640595436511962

6. Lasso Regression:

Training set: 80%, Test set: 20%, 5-fold cross-validation for the training set (8%) of training data.

Cross validation Mean Score	0.030007579795205946
Standard Deviation Score	0.003520597805276883
Test Mean Squared Error	0.028992806775098037
Root Mean Squared Error	0.1702727423139066
Mean absolute Error	0.1259128146258776
R2_score	0.4596042222356015

7. Ridge Regression:

Training set: 80%, Test set: 20%, 10-fold cross-validation for the training set (8%) of training data.

Cross validation Mean Score	0.030694200932455847
Standard Deviation Score	0.0025640225125444704
Test Mean Squared Error	0.030770364610363327
Root Mean Squared Error	0.17541483577612052
Mean Absolute Error	0.1279716694869285
R2 Score	0.4264723921109502

Conclusion

In our initial exploration of the dataset, we observed a wide range of values and high variance across its features. Our primary intuition was that the model might generalize predictions due to this variability.

Following extensive preprocessing and exploratory data analysis (EDA), we applied various feature engineering techniques and employed multiple methods to select the most influential features for our model.

Among the models evaluated, the XGBoost model yielded the highest R-squared (R^2) score of 0.664.

This outcome proved our initial intuition, suggesting that the model's ability to generalize predictions was influenced by the wide range of features encompassed in the dataset. All models achieved an average or above-average R^2 score, indicating good predictive capability without exceptionally high accuracy.