Introduction:

In this project, I investigate a dataset of songs sourced from Spotify to predict the popularity score of songs based on their audio features and metadata. The dataset provides a rich collection of features, including acoustic properties (e.g., danceability, energy, acousticness), metadata (e.g., song name, artist, release date and year), and popularity metrics. By analyzing these features, I seek to understand how musical characteristics have changed over decades and identify the factors that influence a song's popularity. This analysis is particularly relevant in the era of digital streaming, where understanding listener preferences can inform music production and marketing strategies.

The methodology employed includes data preprocessing to clean and standardize the dataset, exploratory data analysis to visualize trends and relationships, and regression models to predict song popularity. Key findings reveal intriguing trends in musical attributes, such as shifts in energy levels and tempo over time, as well as insights into the features most correlated with popularity. By combining historical analysis with predictive modeling, this project not only highlights the evolution of music but also provides actionable insights for stakeholders in the music industry.

Data Description:

The dataset used in this project is sourced from the following github repo (https://raw.githubusercontent.com/gabminamedez/spotify-data/master/data.csv), containing over 160,000 tracks from 1921 to 2020 spanning multiple decades. It contains 169909 rows and 19 columns representing their different features, attributes and metadatas, which I can use to help predict IMDb score.The 19 features for each track are: id, name, artists, duration_ms, release_date, year, acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo, valence, mode, key, popularity, and explicit.

The dataset is loaded using pandas from a CSV file hosted on GitHub . Initial examination shows that the dataset contains a mix of numerical and categorical variables, with some features normalized between 0 and 1, while others (like duration_ms and loudness) are on different scales. It will require some cleaning and munging, given the presence of potentially unnecessary columns and null values within the dataset. Challenges may arise in constructing an

accurate regression model due to the inherent volality of the data. However, I believe that certain predictive variables will be able to at least somewhat predict populariy scores with some degree of accuracy.

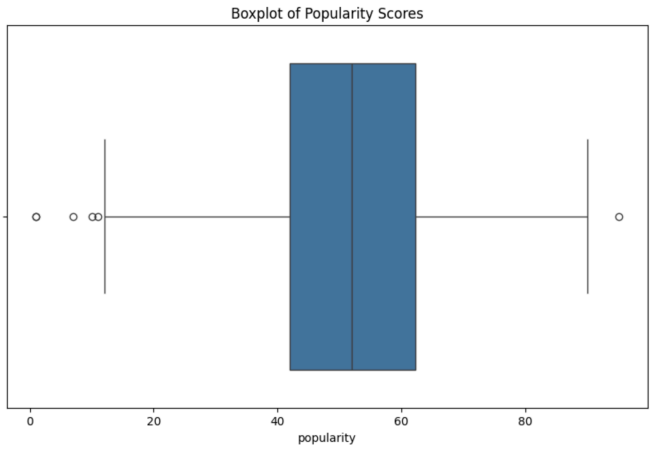
Models and Methods:

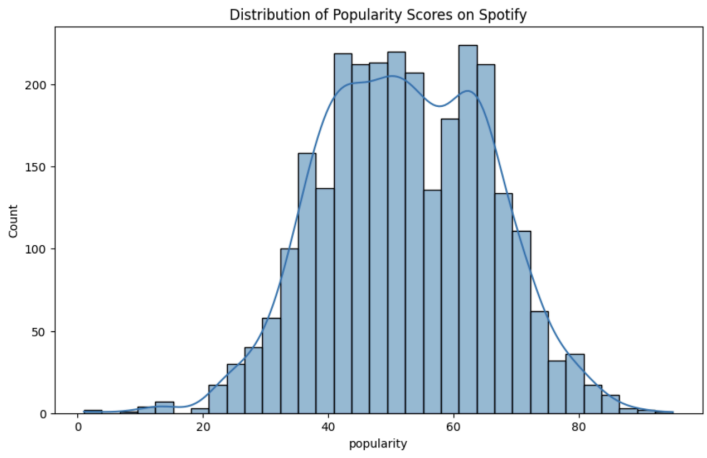
Data Preprocessing:

The data preprocessing phase involved several crucial steps to ensure the dataset's quality and relevance for analysis. First, all rows containing null values were removed, eliminating incomplete records that could potentially skew the results. Next, rows with zero values across all columns were filtered out, as these likely represented erroneous or placeholder entries. Finally, irrelevant columns that were not necessary for the analysis were dropped. These included 'id', 'name', 'artists', and 'release_date', which were primarily identifying or descriptive fields rather than numerical features useful for modeling. The resulting dataset now contains only relevant, non-zero, and complete entries, providing a cleaner and more focused set of data for subsequent analysis and modeling tasks. The revised dataset that I will be working with contains 2788 spotify tracks released from 1929 to 2020

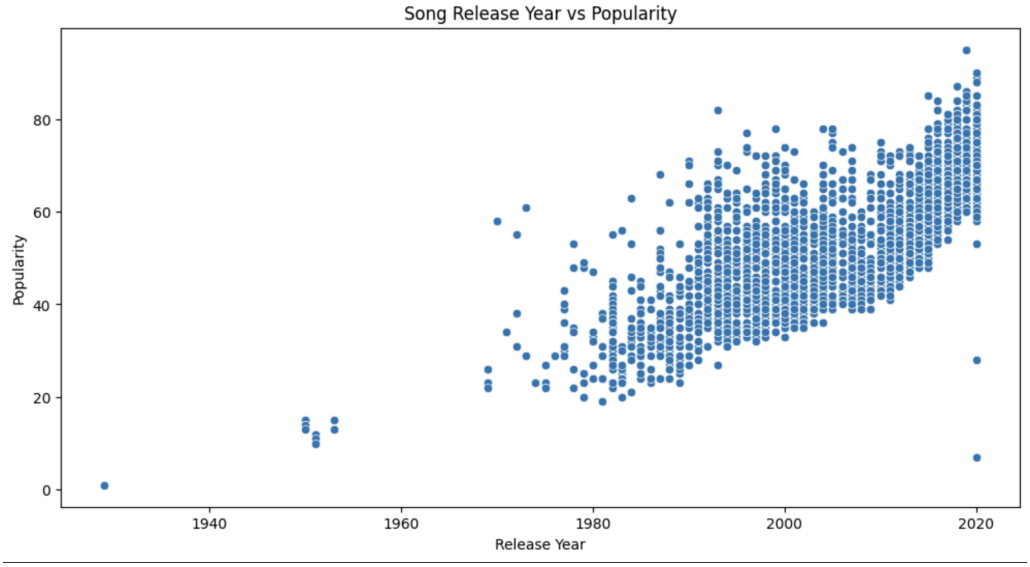
Exploratory Data Analysis:

The exploratory data analysis of the Spotify dataset reveals several interesting patterns and distributions across different variables. The descriptive statistics show that the dataset spans from 1929 to 2020, with tempo ranging from 45.518 to 215.669 BPM. The popularity scores range from 1 to 95, with a mean of approximately 52.33 and a median of 52.0, suggesting a relatively symmetric distribution.

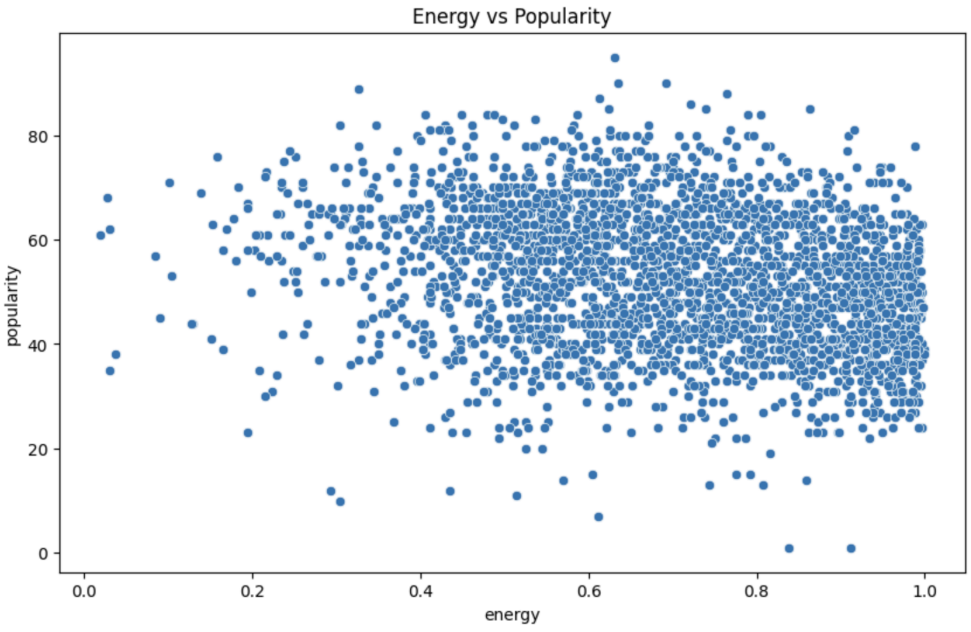




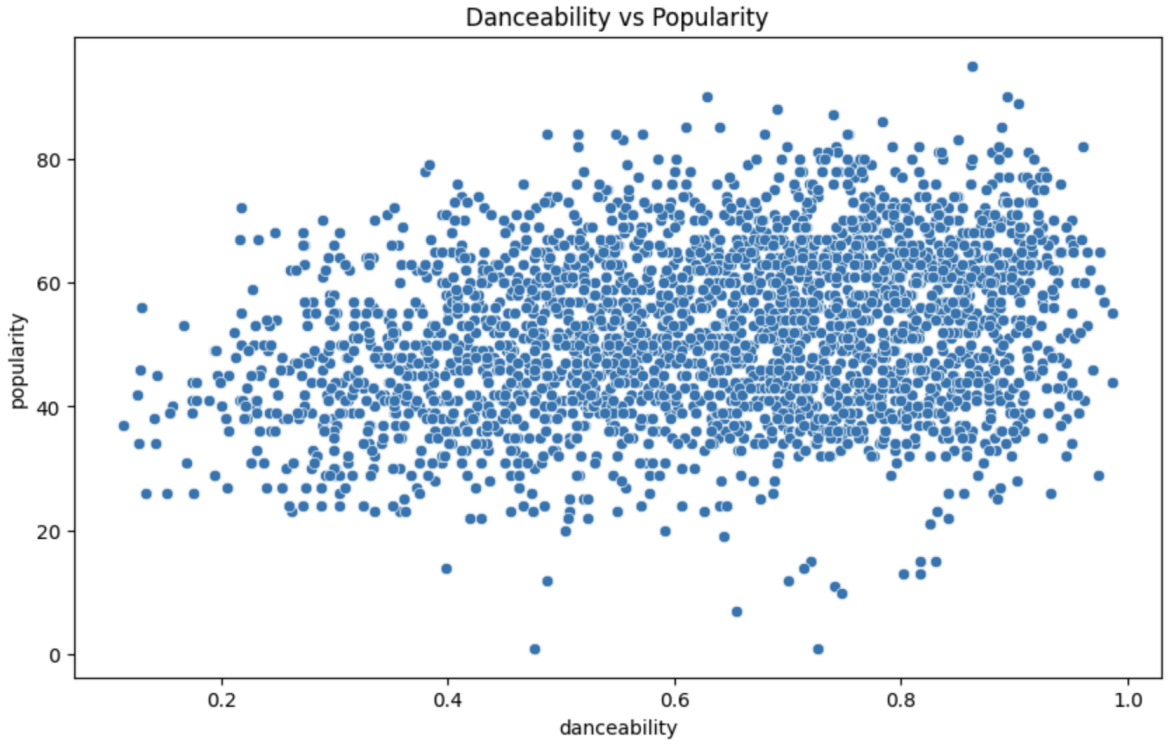
The boxplot of popularity scores demonstrates that the majority of songs fall between approximately 42 and 64 on the popularity scale, with several outliers on both ends. The distribution is further illustrated by the histogram, which shows a roughly normal distribution of popularity scores with a slight right skew. The peak of the distribution occurs around the 40-65 range, with fewer songs at the extreme ends of the popularity spectrum.



A particularly interesting trend emerges in the scatter plot of Release Year vs Popularity, which shows a clear positive correlation between more recent releases and higher popularity scores. This suggests that newer songs tend to have higher popularity ratings on Spotify, possibly due to recency bias or the platform's younger user demographic. The density of points increases significantly from the 1980s onward, indicating more comprehensive data collection in recent decades.

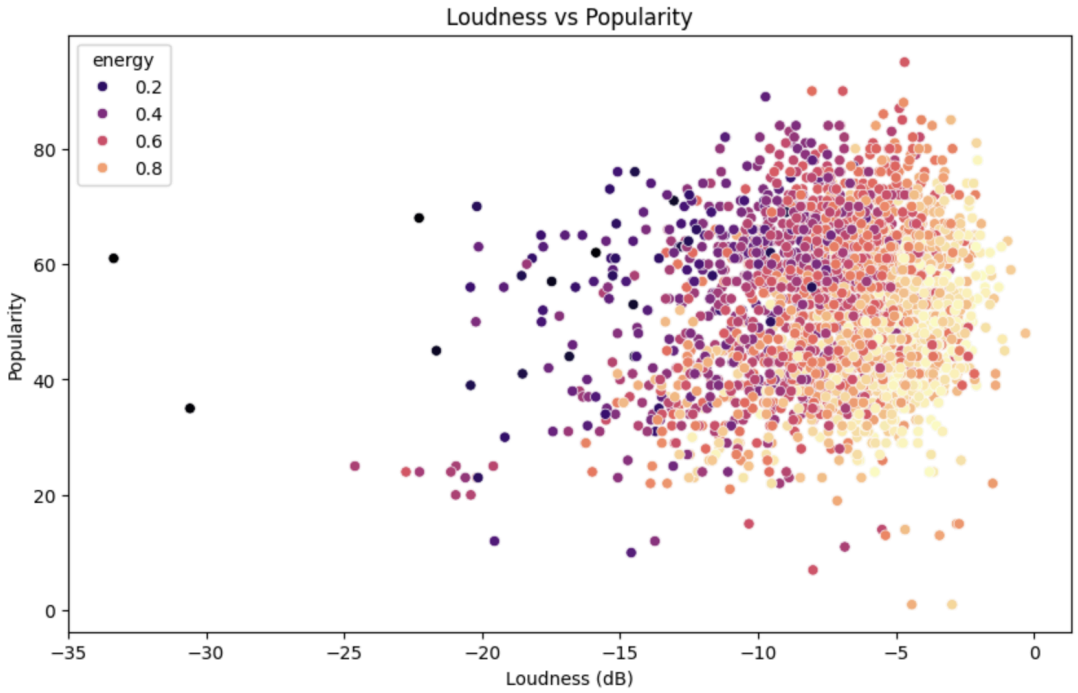


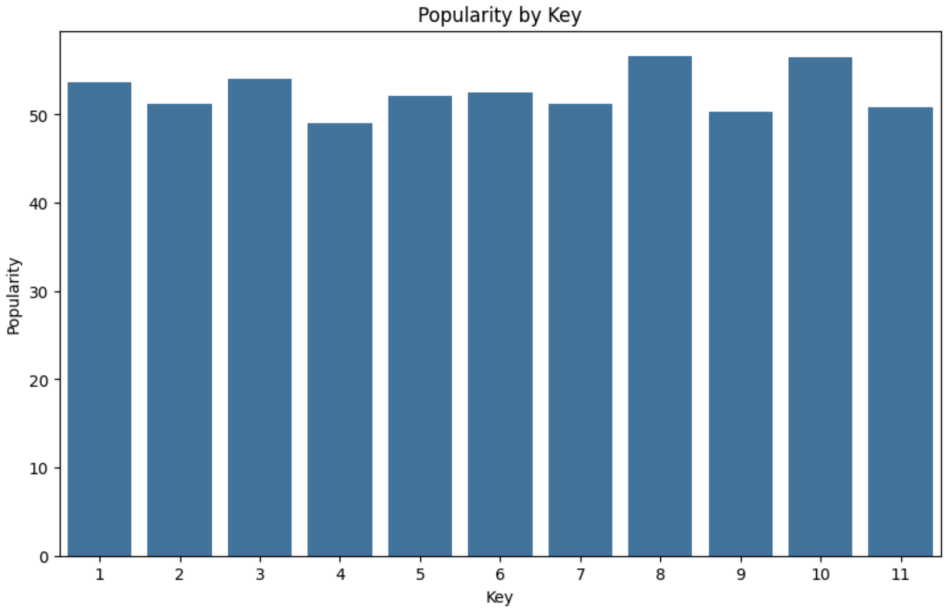
The Energy vs Popularity scatter plot reveals a more complex relationship between these variables. While there isn't a strong linear correlation, there appears to be a slight positive relationship between energy and popularity up to a certain point. The plot shows a dense cluster of points in the middle energy range (0.4 to 1.0), suggesting that songs with moderate to high energy levels tend to be more common in the dataset. However, high popularity scores can be found across all energy levels, indicating that energy alone is not a determining factor for a song's popularity.



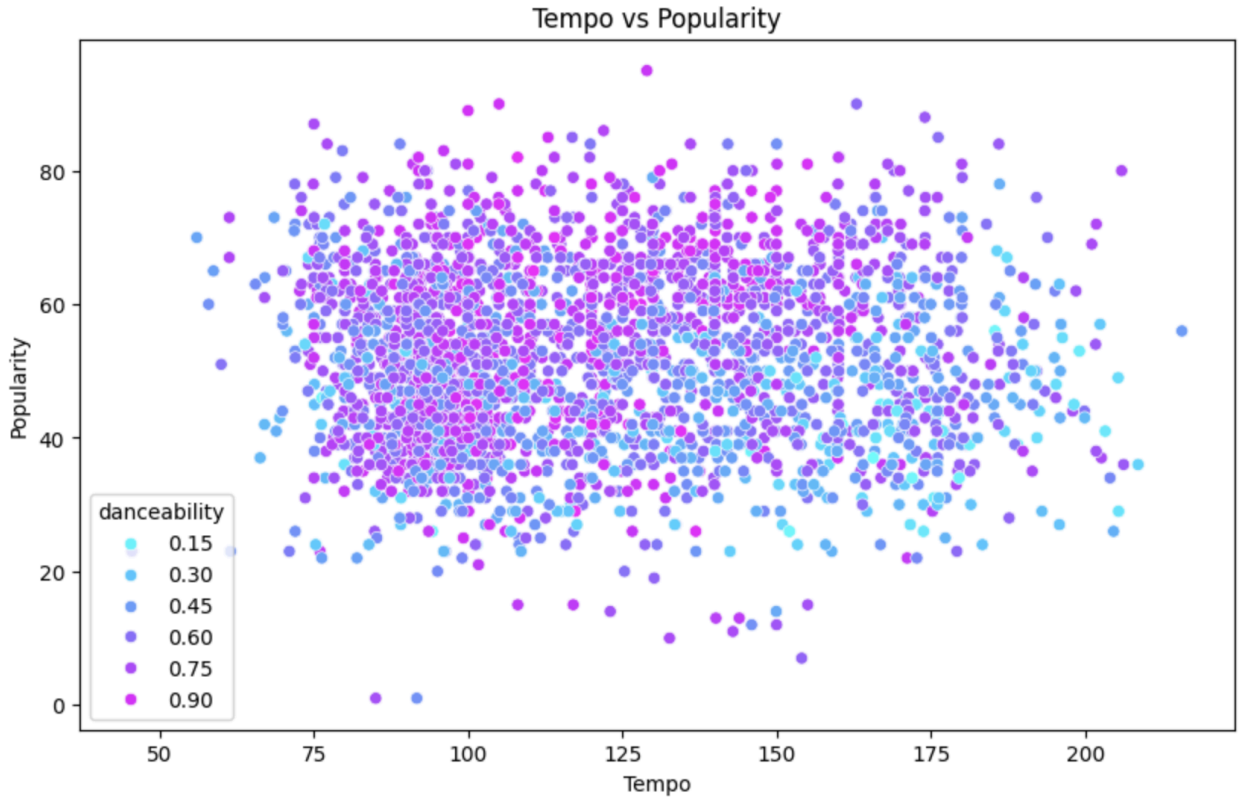
The scatter plot of Danceability vs Popularity shows a slight positive correlation. Songs with higher danceability scores tend to have somewhat higher popularity ratings, although the relationship is not very strong. The majority of songs cluster in the danceability range of 0.4 to 0.9, suggesting that moderately to highly danceable tracks are more common in the dataset.

Examining the Loudness vs Popularity plot, we observe a more pronounced positive correlation. As loudness increases, there's a general trend towards higher popularity scores. This could indicate that louder songs tend to be more popular on the platform, possibly due to their perceived energy or impact.



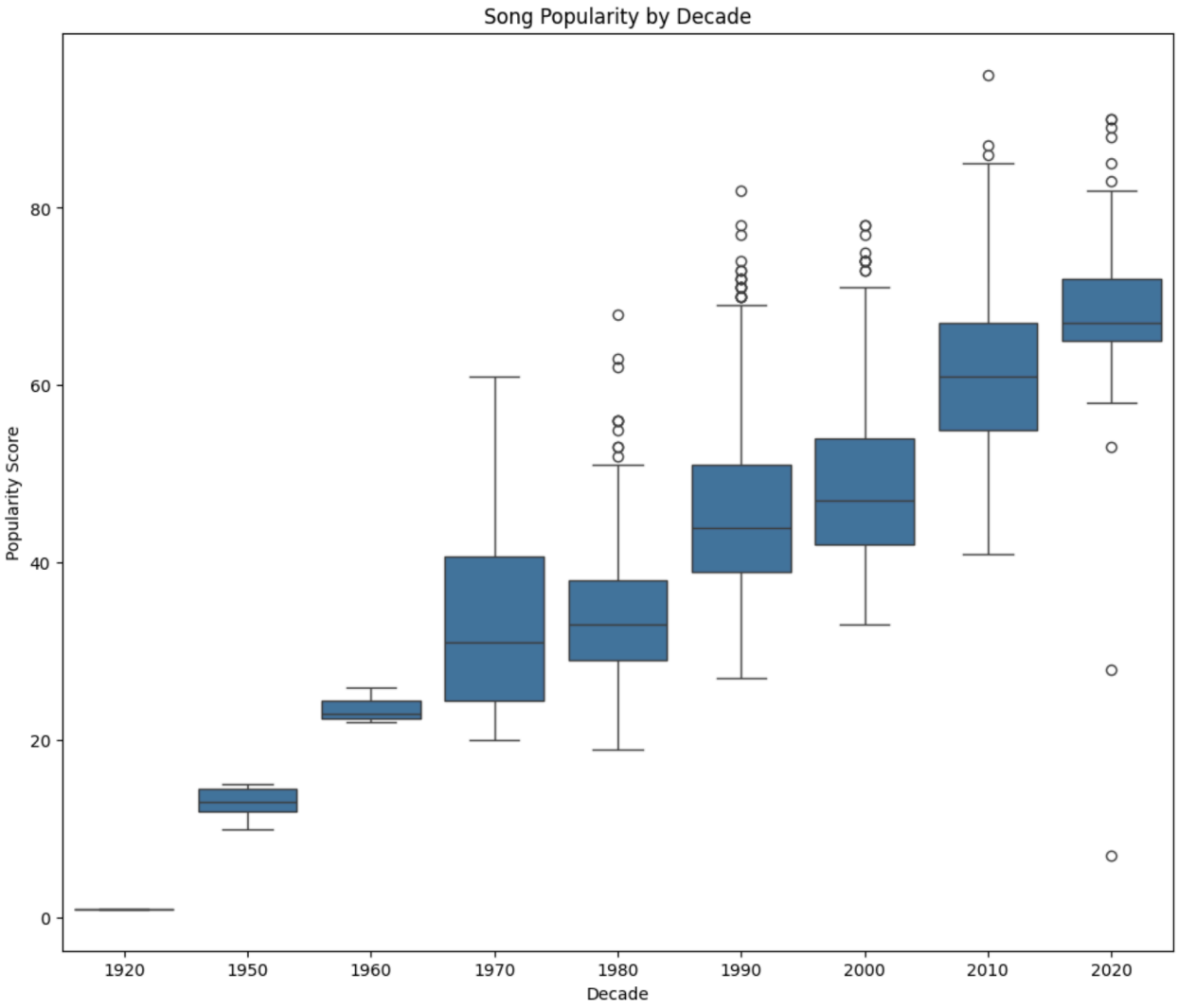


The bar plot of Popularity by Key reveals that certain musical keys may be associated with slightly higher popularity scores. Keys 8 and 10 appear to have the highest median popularity, while key 4 has the lowest. However, the differences are not at all dramatic, suggesting that the key of a song is not a major determinant of its popularity.



The Tempo vs Popularity scatter plot, colored by danceability, provides a multi-dimensional view of these attributes. There doesn't appear to be any strong correlation between tempo and popularity. However, the color gradient suggests that songs with higher danceability (represented by warmer colors) tend to cluster in the lower mid-tempo range and often have higher popularity scores.

Finally, the box plot of Song Popularity by Decade demonstrates a clear trend of increasing popularity scores over time. There's a consistent upward trend from the 1920s to the 2020s, with each subsequent decade showing higher median popularity scores. This could be due to various factors, including recency bias in streaming platforms, improved recording quality, or changes in musical tastes over time. The spread of popularity scores also increases in more recent decades, suggesting greater variability in song reception in modern times.



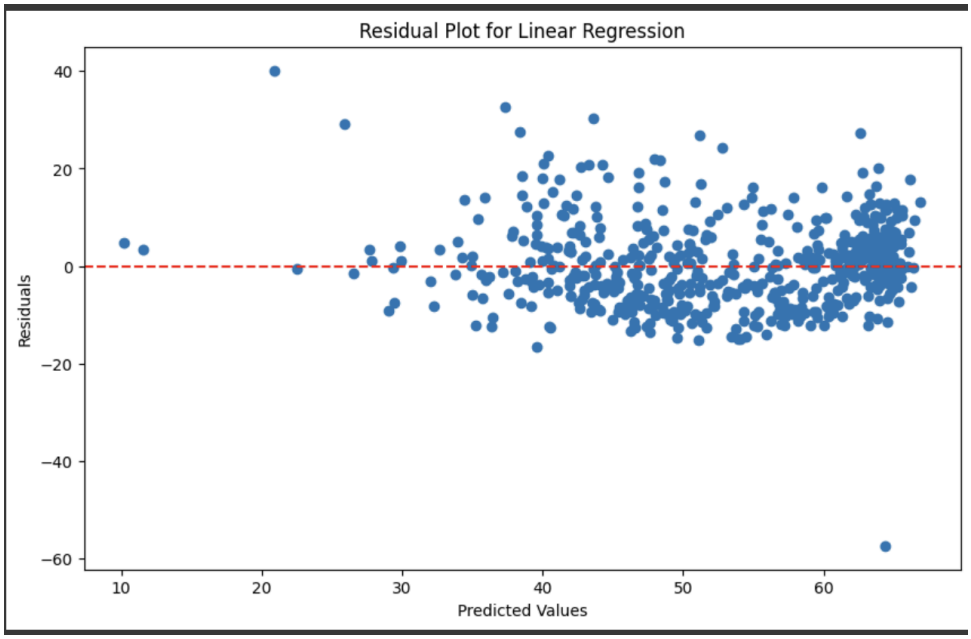
These visualizations provide valuable insights into the complex interplay between various musical attributes and song popularity on Spotify, highlighting both clear trends and more nuanced relationships that warrant further investigation.

Models and Methods:

To predict popularity scores, I decided to use multiple different regression models and see which one performs the best in predicting these scores and accounting for the variation in my data and the fluctuations in score. For each of these models, I decided to utilize an 80-20 train-test split, training my model on 80% of the data and then testing it on the remaining 20%.

I prepared the data with splitting the dataset into training and test sets using an 80-20 split ratio, resulting in 2,230 training samples and 558 test samples, each with 15 features. Before training the model, the features were standardized using StandardScaler to ensure all variables were on the same scale, which is crucial for linear regression performance.

1. Linear Regression Model:



The model was trained on the scaled features using scikit-learn's LinearRegression class, which fits the model by minimizing the residual sum of squares between the observed targets and the predicted values. The model's performance metrics reveal moderate predictive capability, with a training MSE of 72.17 and a test MSE of 79.45. The R-squared scores of 0.59 for training and 0.55 for test data indicate that the model explains approximately 55-59% of the variance in song popularity. Feature importance analysis shows that year (11.18) and decade (1.91) are the

strongest predictors of popularity, followed by energy (1.08), loudness (1.03), and danceability (0.67).

The residual plot shows a relatively random scatter around zero, though with some heteroscedasticity at higher predicted values, suggesting that the model's assumptions are reasonably met but I think there might be room for improvement through non-linear modeling approaches. So I decide to do KNN regression as my next regression model.

2. K-Nearest Neighbors Regression Model:

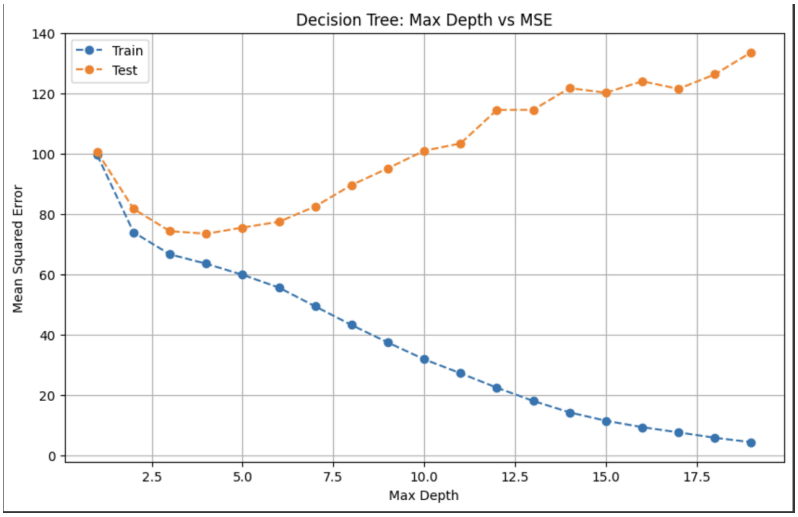
The K-Nearest Neighbors (KNN) regression model was implemented using scikit-learn's KNeighborsRegressor within a pipeline that included StandardScaler for feature normalization. A grid search was performed to optimize the model's hyperparameters, exploring different values for n_neighbors (5, 10, 15, 20, 25, 30) and weights ('uniform', 'distance'). The grid search utilized 5-fold cross-validation with negative mean squared error as the scoring metric. The optimal parameters were found to be 15 neighbors with uniform weights, suggesting that these settings provided the best balance between bias and variance.

The KNN model's performance metrics show promising results, with a training MSE of 71.37 and test MSE of 85.09, along with R-squared scores of 0.59 for training and 0.52 for test data. Feature importance analysis using permutation importance reveals that temporal features (year and decade) are the most significant predictors, with importance scores of 0.22 and 0.14 respectively, followed by audio features like liveness (0.014), instrumentalness (0.011), and danceability (0.011). This suggests that the release timing of a song has a stronger influence on its popularity than its acoustic characteristics, though both contribute to the model's predictions.

3. Decision Tree Regression Model:

The Decision Tree Regression model was implemented using scikit-learn's DecisionTreeRegressor within a pipeline that included StandardScaler for feature normalization. To optimize the model's performance, a systematic approach was taken to find the ideal maximum depth of the tree. The analysis explored depths ranging from 1 to 20, evaluating the

model's performance on both training and test sets using Mean Squared Error (MSE) as the metric.



The results, visualized in the "Decision Tree: Max Depth vs MSE" plot, reveal a clear trade-off between model complexity and generalization. As the tree depth increases, the training MSE consistently decreases, indicating an improved fit on the training data. However, the test MSE shows a different pattern, initially decreasing until it reaches an optimal point at a depth of 4, after which it begins to increase. This behavior is characteristic of the bias-variance trade-off, where deeper trees overfit the training data, leading to poor generalization on unseen data. The optimal depth of 4 was selected as it minimizes the test MSE, resulting in a model with a training MSE of 63.5838, a test MSE of 73.5388, and R-squared scores of 0.6390 and 0.5839 for training and test sets respectively. These metrics suggest that the model explains approximately 58-64% of the variance in song popularity, demonstrating moderate predictive power while avoiding overfitting.

4. Random Forest Regression Model:

The Random Forest Regression model was implemented using a pipeline that combined StandardScaler for feature normalization and RandomForestRegressor. A grid search was performed to optimize the model's hyperparameters, exploring different values for n_estimators (50, 100, 200), max_depth (5, 10, 15, None), and min_samples_split (2, 5, 10). The best parameters found were 50 estimators, a max depth of 5, and a min_samples_split of 5. The model achieved strong performance with a training MSE of 56.5095 and a test MSE of 70.1871, along with R-squared scores of 0.6792 for training and 0.6028 for test data. Feature importance analysis revealed that the year of release was by far the most significant predictor (0.844903), followed by decade (0.076022) and loudness (0.017750). This suggests that temporal features

play a crucial role in predicting song popularity, with audio characteristics like loudness, duration, and speechiness also contributing to the model's predictions, albeit somewhat.

5. Gradient Boosting Regression:

The Gradient Boosting Regressor was implemented to further explore the non-linear relationships between features and song popularity, leveraging its iterative nature to optimize predictions. A pipeline with scaling and hyperparameter tuning was used, resulting in the best parameters: 300 estimators, a learning rate of 0.01, and a maximum depth of 3. This model demonstrated strong performance, achieving a training MSE of 60.50 and a test MSE of 69.31, with R-squared scores of 0.66 for training and 0.61 for testing. Feature importance analysis revealed that temporal factors like year (0.835) and decade (0.112) were the most significant predictors, followed by loudness (0.017), indicating that while audio features contribute to popularity, temporal features dominate. Gradient Boosting's ability to handle complex interactions and reduce overfitting makes it a robust choice for this task, providing slightly better predictive power compared to other models like Random Forest.

In all these models, Linear Regression serves as a fundamental baseline, providing straightforward interpretation of feature importance and linear relationships. KNN builds upon this by introducing non-linearity and capturing more complex patterns in the data. The Decision Tree model further explores non-linear relationships and feature interactions, while the Random Forest, as an ensemble method, combines multiple decision trees to create a robust, high-performance model capable of handling complex interactions.

After exploring the regression models, incorporating a neural network as the final step in my analysis was a logical progression I think. Neural networks excel at capturing complex, non-linear relationships in high-dimensional data, which is particularly relevant for the multifaceted nature of song popularity prediction. This approach would allow me to potentially uncover subtle patterns and interactions that might have been missed by the previous models.

6. Neural Networks Model(PyTorch):

The Neural Network implementation utilized a simple feedforward architecture trained with the Adam optimizer and Mean Squared Error (MSE) loss function. The model was trained for 100 epochs with a batch size of 32, showing consistent improvement in loss values from 99.6292 in epoch 10 to 57.9398 in the final epoch. The model achieved a training MSE of 84.2474 and a test MSE of 76.9038, with R-squared scores of 0.6835 for training and 0.5648 for testing data. This performance indicates that while the neural network was able to capture meaningful patterns in the data, there is some overfitting as evidenced by the gap between training and testing metrics. The gradual decrease in loss values across epochs suggests that the learning rate of 0.001 was appropriate for this task, allowing the model to converge steadily without getting stuck in local minima or overshooting the optimal parameters.

Results and Interpretation:

1. Model Performance Comparison:

The Random Forest Regressor demonstrated strong performance, achieving a training MSE of 56.5095 and a test MSE of 70.1871. Its R-squared scores of 0.6792 for training and 0.6028 for testing indicate that the model explains approximately 60-68% of the variance in song popularity. The best parameters for this model were 50 estimators, a max depth of 5, and a min_samples_split of 5.

The Gradient Boosting Regressor showed slightly better performance, with a training MSE of 60.5006 and a test MSE of 69.3114. Its R-squared scores of 0.6565 for training and 0.6078 for testing suggest that it explains about 61-66% of the variance in popularity. The optimal parameters for this model were 300 estimators, a learning rate of 0.01, and a max depth of 3.

The Neural Network model achieved comparable results, with a training MSE of 55.7567 and a test MSE of 76.9038. Its R-squared scores of 0.6835 for training and 0.5648 for testing indicate good performance on the training set but some overfitting, as evidenced by the lower test score.

2. Feature Importance:

Both the Random Forest and Gradient Boosting models consistently identified the year of release as the most important feature in predicting song popularity. The Random Forest model assigned an importance of 0.844903 to the year, while the Gradient Boosting model gave it 0.834997.

The decade feature was the second most important, with importance scores of 0.076022 and 0.112407 for Random Forest and Gradient Boosting, respectively. This underscores the significant role that temporal factors play in a song's popularity.

Among the audio features, loudness emerged as the most influential, followed by duration, tempo, and danceability. However, their importance scores were substantially lower than those of the temporal features, suggesting that while audio characteristics do contribute to popularity, they are less predictive than the release timing.

3. **Interpretation:**

Temporal Relevance: The dominance of year and decade in feature importance suggests that recency plays a crucial role in a song's popularity on Spotify. This could be due to various factors, such as the platform's user demographics, algorithmic preferences, or cultural trends favoring newer music.

Audio Characteristics: While less important than temporal factors, certain audio features do contribute to popularity prediction. Loudness, in particular, seems to have a positive correlation with popularity, which aligns with the "loudness war" phenomenon in music production.

Model Consistency: The similarity in feature importance rankings across different models (Random Forest and Gradient Boosting) adds confidence to these findings, suggesting that these patterns are robust across different modeling approaches.

Predictive Power: With R-squared values around 0.60-0.65 for test data across models, we can conclude that while these features provide significant predictive power, there's still a

substantial portion of popularity variance unexplained by our models. This suggests that other factors not captured in our dataset (e.g., marketing efforts, artist fame, social media trends) likely play important roles in determining a song's popularity.

These results provide valuable insights for music industry professionals, potentially guiding decisions in music production, release timing, and marketing strategies. However, the models' limitations also highlight the complex nature of music popularity, which extends beyond purely quantifiable audio features and release timing.

Conclusion and Next Steps:

This comprehensive analysis of the Spotify dataset has revealed significant insights into the factors influencing song popularity. The implementation of multiple regression models, including Linear Regression, KNN Regression, Decision Tree Regression, Random Forest, Gradient Boosting, and a machine learning model - Neural Networks, consistently demonstrated that temporal features (year and decade) are the strongest predictors of song popularity. After year and decade, the two factors that can most accurately predict song popularity are loudness and energy. The feature importance analysis from the Random Forest model showed that loudness (0.017750) was the third most important predictor after year and decade. Additionally, in the Linear Regression model results, energy (1.08) and loudness (1.03) were identified as the next most significant predictors of popularity after the temporal features. The models achieved R-squared scores ranging from 0.55 to 0.60 on test data, indicating moderate predictive power. The Gradient Boosting model performed particularly well, with a test MSE of 69.3114 and an R-squared score of 0.6078, suggesting that it captured meaningful patterns in the relationship between song features and popularity.

Several promising directions for future research could enhance this analysis. First, incorporating additional features such as genre classifications, artist popularity metrics, and social media engagement data could provide a more complete picture of what drives song popularity. Second, developing more sophisticated deep learning architectures or ensemble methods might capture

complex interactions between features that our current models may have missed. Finally, creating a time-series analysis component could help better understand how musical preferences evolve over time and potentially forecast future trends in music popularity. These enhancements, combined with regular model retraining as new data becomes available, could make the system more robust and valuable for music industry applications.