"Popularity Prediction & Personalized Song Recommendations for Spotify"

Team Members:

1.

2.

3.

4.

**Goals and Objectives**

**Primary Goals:**

**Develop a Robust Song Popularity Prediction Model**: Utilize advanced machine learning algorithms to analyze the Spotify dataset and predict song popularity. This model will leverage a range of song features including acousticness, danceability, energy, instrumentalness, key, liveness, loudness, audio mode, speechiness, tempo, time signature, and audio valence.

**Secondary Goals:**

1. **Enhance User Experience on Spotify**: Through the recommendation system, improve user engagement by providing more relevant and engaging content, tailored to individual preferences and listening habits.
2. **Create a Personalized Song Recommendation System**: Design a recommendation engine that curates personalized playlists for Spotify users. This system will use clustering, PCA for dimensionality reduction, and NLP techniques to understand song characteristics, lyrical content, and user preferences, delivering more targeted and resonant song suggestions.

**Objectives:**

1. Data Collection and Preprocessing: Efficiently gather and process data using Spotify's API, ensuring high-quality datasets for model training and analysis.
2. Model Development and Optimization: Experiment with various machine learning models, evaluate their performance, and optimize them for the best results in terms of accuracy and efficiency.
3. Integration and Testing: Seamlessly integrate the models into Spotify's platform and conduct rigorous testing to ensure reliability and effectiveness in real-world scenarios.
4. Feedback Incorporation and Finalization: Gather user feedback, refine the system accordingly, and prepare for final deployment, ensuring the product meets the intended goals and user needs.

**Desired Outcomes:**

- A significant increase in user engagement and satisfaction on the Spotify platform.

- Improved visibility and accessibility of a broader range of music, benefiting users and artists alike.

- Establishment of a new standard in music recommendation systems, showcasing the successful application of machine learning and NLP techniques in enhancing user experience.

**Related Work (Background)**

The field of Hit Song Prediction (HSP) under Music Information Retrieval (MIR) extensively explores machine learning methods to predict the likelihood of a song becoming a hit. This research is valuable for artists, music labels, and vendors, as it helps identify potential hits that could resonate with a large audience. Notably, a study titled "SpotHitPy" used nearly 18,500 hit and non-hit songs, analyzing audio features through the Spotify Web API. This study achieved approximately 86% accuracy in predicting Billboard success, with Random Forest and Support Vector Machine (SVM) as the most successful algorithms.[[Source](https://ar5iv.labs.arxiv.org/html/2301.07978)].

In HSP, the primary assumption is that hit songs share similar audio features. Various approaches, including deep learning techniques and combinations of low- and high-level audio features, have been explored in the field. Some studies have achieved accuracy rates of around 75% to 88% in predicting Billboard success.[[Source](https://ar5iv.labs.arxiv.org/html/2301.07978)]

**Dataset:**

For the project, two datasets were central:

1. The Spotify Playlists Dataset from Kaggle, providing a diverse array of songs, artists, and playlist data.
2. The Song Popularity Dataset, also from Kaggle, containing details like track popularity, features, and metadata.

The datasets included features such as Spotify track ID, artist name, song popularity (ranging from 0 to 100), explicit content, album type, danceability, energy, key, loudness, mode, speechiness, acousticness, and more[[Source](https://www.kaggle.com/datasets/yasserh/song-popularity-dataset/data)].

**Detail Design of Methods:**

**Data Collection and Preprocessing**

Data was gathered from the Billboard top 100 hits using the Billboard API and the Spotify API, employing libraries like `billboard.py` and `spotipy`. The dataset was enriched with audio features from Spotify, following a rigorous data cleanup to ensure uniqueness and completeness of tracks. Techniques like RandomOverSampler from the `sklearn` library were used to balance the dataset, addressing the issue of class imbalance.

**Feature Selection and Extraction**

Principal Component Analysis (PCA) was utilized post-standardization to identify significant features and reduce dataset dimensions. This process involved creating new variables (principal components) as linear combinations of initial variables, focusing on the first few components that accounted for the majority of variance.

**Analysis:**

Exploratory Data Analysis (EDA) plays a crucial role in understanding the underlying patterns of the Spotify dataset. The histogram of the `song\_popularity` reveals a normal distribution, indicating that most songs have a moderate popularity score, with fewer songs at the extremes of low and high popularity. This suggests that while hits are rare, the majority of songs maintain a certain level of listener interest.

The categorical feature visualization showcases the distribution of variables like `audio\_mode` and `time\_signature` across songs. The relatively even distribution of `audio\_mode` between major and minor keys could imply a diverse musical mood within the dataset, whereas the `time\_signature` predominantly favors common time (4/4), highlighting a strong preference for this time signature in song production.

The distribution of numeric features such as `danceability`, `energy`, `speechiness`, `acousticness`, `instrumentalness`, `loudness`, `liveness`, `tempo`, and `audio\_valence` is illustrated through histograms. Each feature presents a distinct distribution, with `danceability` and `energy` showing a skew towards higher values, suggesting a trend towards more energetic and danceable tracks in the dataset. In contrast, `acousticness` and `instrumentalness` exhibit a steep drop-off, indicating fewer acoustic and instrumental tracks relative to electronic or amplified music.

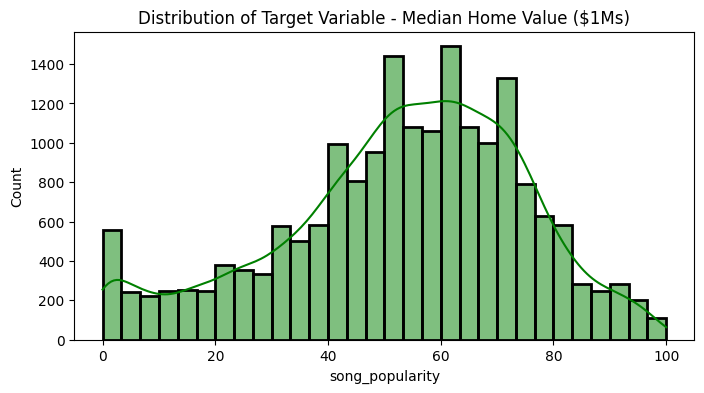
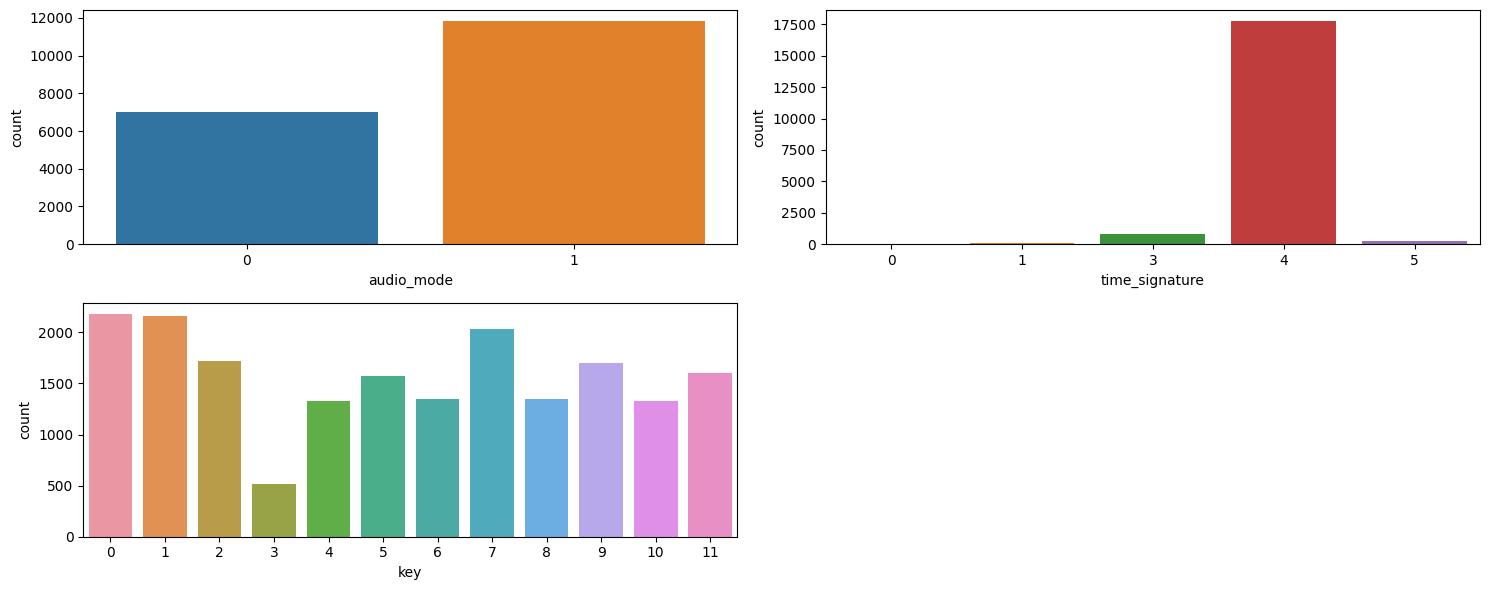
 

Figure 1 Distribution of song\_popularity with count. Figure 2 : Categorical Features Visualization

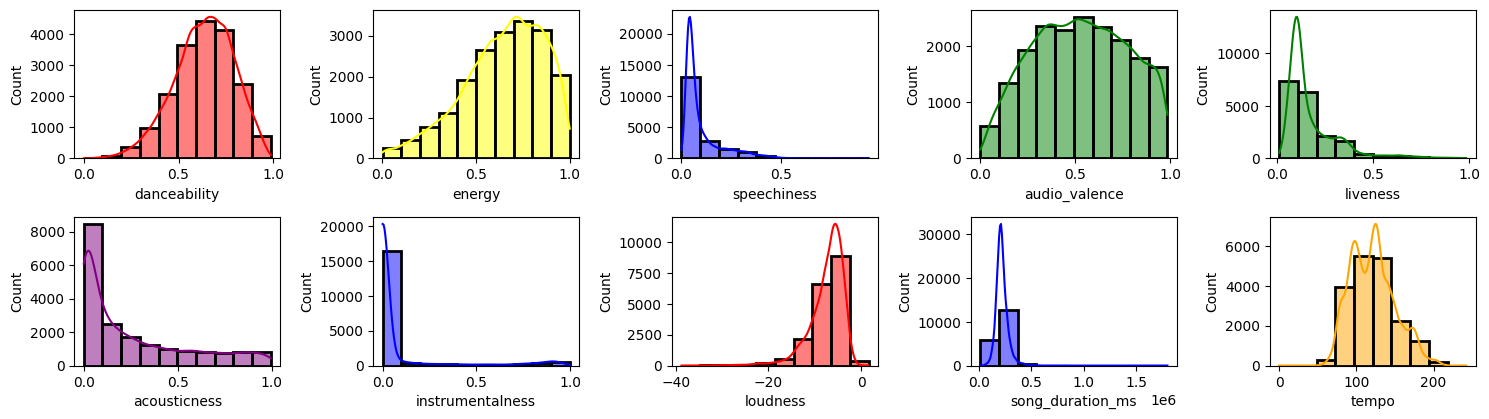


Figure 3 Numeric Features Distribution

**Data Pre-processing**

The data pre-processing phase began with ensuring data quality by removing duplicate records, resulting in 3,911 duplicates being identified and removed from the dataset. Subsequently, a meticulous inspection for null values was conducted, and the data was cleansed accordingly to ensure a robust dataset for model training.

Encoding categorical variables is a pivotal step to convert them into a format that can be provided to machine learning algorithms to do a better job in prediction. The dataset utilized One-Hot Encoding for binary categorical features and Dummy Encoding for categorical features with more than two and less than seventeen unique values.

Outlier detection and removal were performed on numerical features using the Interquartile Range (IQR) method to eliminate extreme values that could skew the model results. This step significantly refined the dataset from 14,924 to 8,950 samples, indicating a more concise and focused dataset for training predictive models.

Finally, the dataset underwent a renaming process to ensure consistency in feature names, and a pie chart was created to illustrate the composition of the final dataset. The visualization and the subsequent printouts provided a clear insight into the data retention post-cleanup, confirming that 60% of the data was retained for the final analysis. This pie chart representation encapsulates the extensive data cleaning process, marking the transition into the modelling phase with a clean and prepared dataset.

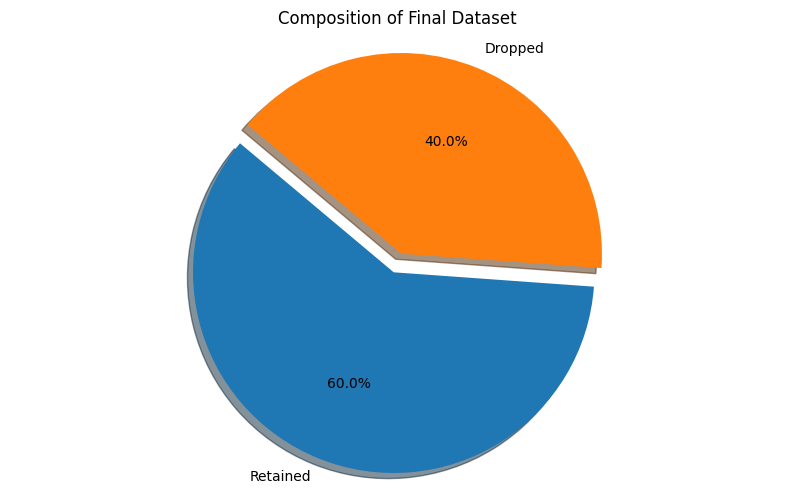


Figure 4 Statistics of removed and retained data

Outliers Analysis:

Before outlier removal: 14924 samples.

After outlier removal: 8950 samples.

**Project Management**

**Implementation Status Report**

**Work Completed:**

- **Model Implementation and Evaluation**: The project successfully implemented multiple regression models to predict song popularity.

- **Description**: Development of **Multi LinearRegression, Ridge LinearRegression, Lasso LInearRegression, Elastic NetRegression,** and **Polynomial Feature Regression** models.

- Responsibility: Machine Learning Engineer (Person responsible for model development and evaluation).

- **Contributions**: All team members contributed, with the Machine Learning Engineer taking a lead role, accounting for approximately 60% of the work, supported by the Data Engineer and NLP Specialist for data preparation and feature engineering.

**Results achieved**

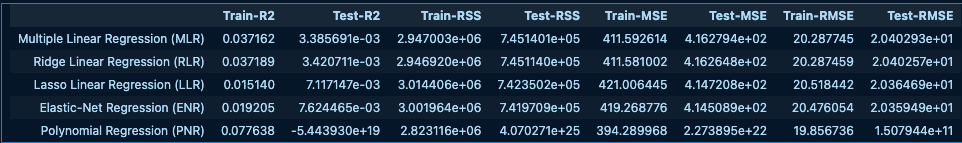
****

Figure 5 Implemented Model's Comparison

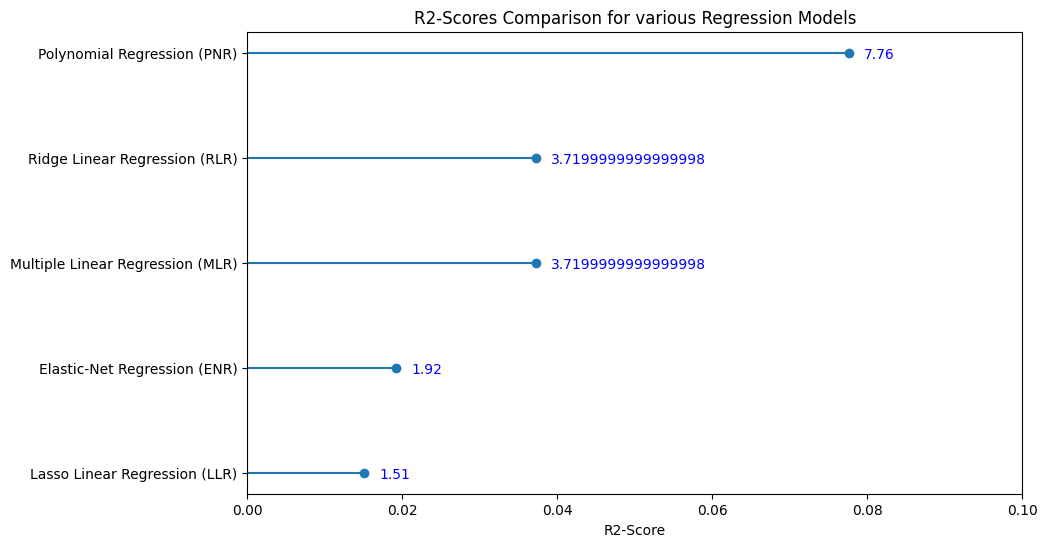


Figure 6 : Comparison of so-far Trained Model

**Work to be Completed:**

- Song Recommendation System Development:

- Description: Design and implementation of the personalized song recommendation system.

- Responsibility: Team member 1 (Data collection and preprocessing), team Member 2 (EDA and data modification), Team Member3 (Model creation and fitting), Team member 4 (evaluation and integration)

**Issues/Concerns:**

- A need for further optimization of models based on evaluation metrics.

- Balancing the contribution among team members to ensure an even workload distribution.

**References/Bibliography**

1. Spotify Playlists Dataset, Kaggle [[Source](https://www.kaggle.com/datasets/andrewmvd/spotify-playlists)].

2. Song Popularity Dataset, Kaggle .[[Source](https://www.kaggle.com/datasets/yasserh/song-popularity-dataset/data)]

3. "SpotHitPy: A Study For ML-Based Song Hit Prediction Using Spotify" for insights into similar works and methodologies [[Source](https://ar5iv.labs.arxiv.org/html/2301.07978)].

4. Scikit-Learn documentation for machine learning algorithms [[Ridge](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html)] [[Lasso](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html)].