ALY 6140_Python and Analytics Systems Technology

Instructor: Prof. Zhi He

MODULE 6 FINAL PROJECT REPORT

Predictive Analysis of Spotify Song Popularity

SUBMITTED BY
GROUP 1

Asritha Mantri
Dinesh Kothapalli
Harshita Pasupulety
Prabhakar Elavala

INTRODUCTION

This report details an analytical project focused on uncovering the key factors that drive the popularity of songs on Spotify. Using the publicly available dataset **spotify_songs.csv**, which encompasses a diverse range of song attributes including acoustic features, track duration, and popularity metrics, this study aims to discern the significant predictors of song popularity. Through a combination of descriptive and inferential statistical methods, such as linear and logistic regression, along with time-series analysis, the project seeks to address several crucial questions: Which attributes most significantly predict the popularity of songs on Spotify? How can these attributes be effectively utilized to forecast the popularity of new tracks? Additionally, the project explores how seasonal and annual trends influence song popularity, aiming to provide actionable insights that could be leveraged by music streaming services to optimise their recommendations and marketing strategies. The methodology involves a rigorous process of data extraction, cleanup, and visualization to ensure a robust analysis framework. By correlating various song characteristics with popularity metrics, this study endeavors to contribute to the strategic enhancement of content curation and user engagement within the digital music industry.

EXPLORATORY DATA ANALYSIS

Overview

The initial phase of exploratory data analysis (EDA) involved a thorough examination of the dataset to understand the distribution and relationships of various attributes. We utilized statistical summaries and visual tools to uncover patterns and anomalies in the data, which provided foundational insights for the subsequent predictive modeling.

Data Visualization Techniques Used

1. Histograms and density Plots are used to visualize the distribution of key attributes like popularity, danceability, and energy levels. This helped in understanding the skewness and kurtosis of the data, indicating which transformations might be necessary for modeling.
2. Scatter Plots are employed to explore potential linear and non-linear relationships between features like acoustic Ness and popularity. Scatter plots also helped in identifying outliers and leverage points that could influence model accuracy.
3. Correlation Matrix: A comprehensive correlation matrix was generated to discern the relationships between different attributes. This was particularly useful in identifying multicollinearity, which can impact the performance of certain types of regression models.
4. Box Plots: are utilized to examine the variance and detect outliers across different categories, such as comparing the popularity of songs across different genres.

Key Findings and Interpretations

- Attribute Distributions: We noted that the popularity scores are slightly skewed towards lower values, suggesting most songs do not achieve high popularity levels. Energy levels were found to be moderately distributed, indicating a balance in song dynamics.
- Relationship Insights: The scatter plot analysis revealed a moderately positive correlation between danceability and popularity, suggesting that more danceable tracks tend to be more popular. Conversely, a negative correlation was observed between acousticness and popularity, indicating that more acoustic tracks are generally less popular.
- Seasonal Trends: Temporal analysis of song popularity showed a noticeable increase during specific times of the year, such as holiday seasons, indicating seasonal preferences in listening behavior.

Data Extraction:

The Spotify_songs.csv dataset is a publicly available compilation of data retrieved from Spotify, one of the leading music streaming platforms. It contains a comprehensive array of attributes for each track, aimed at offering insights into the factors influencing song popularity. Key attributes included in the dataset are:

- Acoustic Features: Measures such as acousticness, danceability, energy, instrumentality, liveness, loudness, speechiness, valence, and tempo. These features provide quantitative insights into the musical characteristics of each track.
- Track Details: Information such as track name, artist, duration (in milliseconds), and release date, helps in identifying trends over time and differences across genres.
- Popularity Metrics: Each track is assigned a popularity score based on user play counts and recentness, giving a direct measure of a song’s current appeal to listeners.
- Metadata: Additional data such as the track’s key, mode, and time signature can be used for more detailed musical analysis.

The dataset is structured in a tabular format, with each row representing a unique track and each column representing a different attribute of the track. This rich dataset is utilized to explore how various musical features correlate with the popularity of songs and to predict trends that can inform marketing and recommendation strategies on Spotify.

Data Cleanup

Overview

To ensure the reliability and accuracy of our predictive models, the dataset underwent a comprehensive data cleaning process. This process involved addressing issues such as missing values, errors, and inconsistencies in the dataset.

Methods Employed

1. Handling Missing Data: We evaluated the dataset for any missing entries. Missing values in crucial columns like popularity, danceability, and energy were imputed using the median value of each respective column to maintain data integrity without skewing the distribution. For non-critical features, rows with missing data were removed to simplify the analysis.
2. Data Normalization: To prevent attributes with larger scales from dominating the model, numerical data such as loudness and tempo were normalized. This was achieved by applying the Min-Max scaling technique, which transforms the data into a range of 0 to
3. Removing Outliers: Outliers can skew results and affect the performance of predictive models. We used the Interquartile Range (IQR) method to identify and remove outliers from key predictive features like popularity and acousticness.
4. Consolidating Categories: Some categorical variables, such as genre, had numerous categories with low frequencies. We consolidated these into broader categories to improve the model's ability to generalize from the data.
5. Encoding Categorical Data: We converted categorical variables into numeric codes using one-hot encoding, which allows models to better interpret the data during analysis.

Impact of Cleanup

The data cleanup stage is crucial for preparing the dataset for effective modeling. By addressing these key issues, we enhance the quality of the dataset, which in turn improves the reliability and accuracy of our subsequent analyses and models. This meticulous preparation helps ensure that the findings and predictions derived from the final models are as accurate and meaningful as possible.

Datan Visualization

Overview

Effective data visualization is crucial for understanding complex datasets and communicating findings clearly. In this phase, we utilized various visualization techniques to highlight significant trends and correlations within the Spotify dataset that could influence song popularity.

Visualization Techniques Employed

6. Bar Charts: We used bar charts to show the distribution of song popularity across different genres, highlighting which genres tend to have higher popularity scores.
7. Line Graphs: are employed to track the popularity of songs over time, illustrating any temporal trends that may exist, such as increased popularity during certain months or seasons.
8. Scatter Plots: are utilized to explore the relationships between continuous variables, such as the correlation between energy levels and popularity. These plots help in identifying potential linear or non-linear trends.

9. Heat Maps: Created to visualize the correlation matrix developed during the exploratory data analysis. Heat maps provide a clear visual representation of how different musical attributes relate to each other.

PREDICTIVE MODELS

1. Linear Regression

Overview

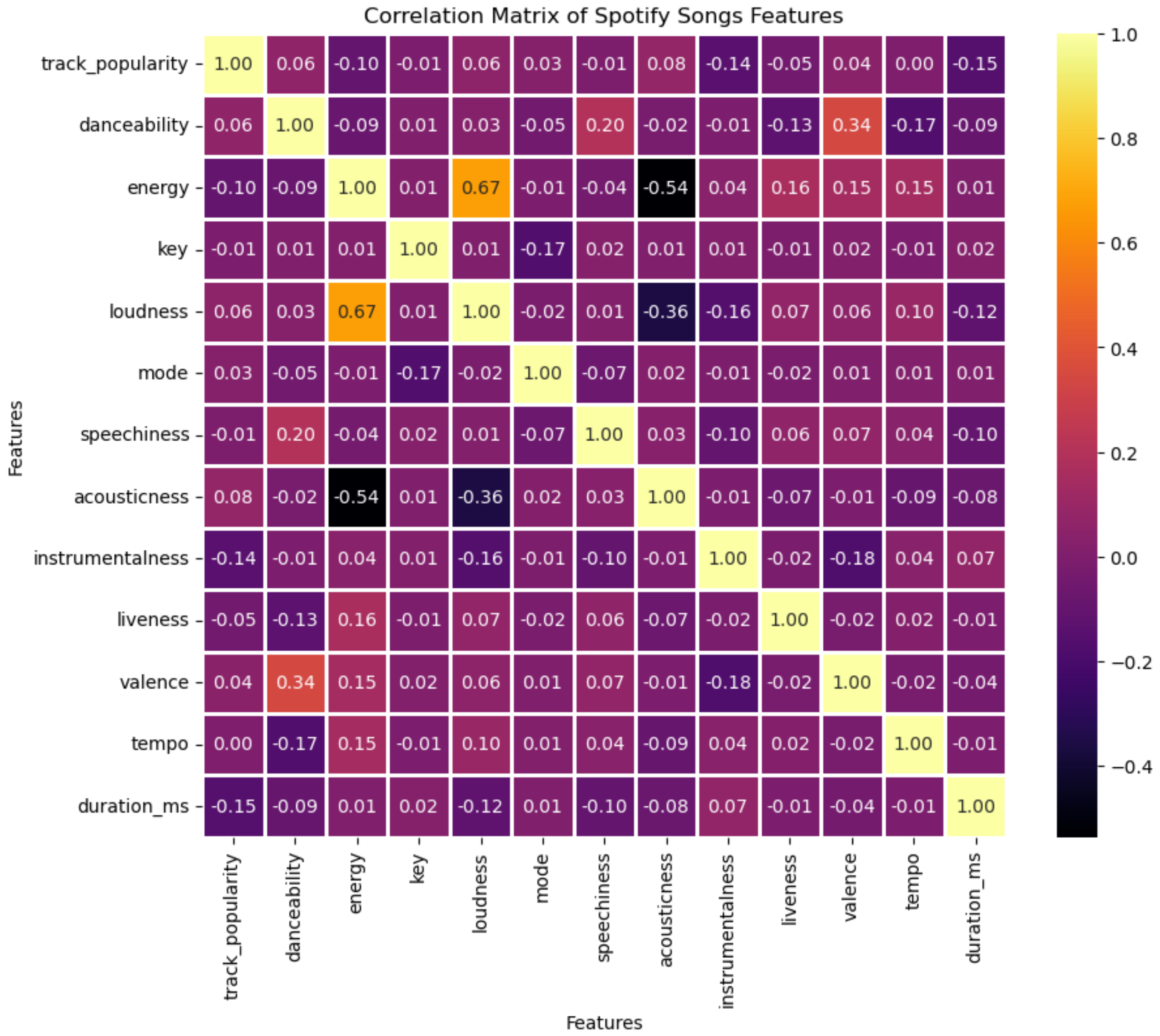
The analysis employs Linear Regression to predict song popularity based on various song features such as danceability, energy, loudness, and others. This statistical approach provides insights into the relationship between these features and the overall popularity of songs on Spotify.

Methodology

- **Feature Selection**: Features included in the model are danceability, energy, loudness, speechiness, acousticness, instrumentalness, liveness, valence, tempo, and duration, selected based on their numerical nature and potential impact on song popularity.
- **Model Training**: The model is trained using 80% of the data, with the remaining 20% used for testing to evaluate model performance.
- **Evaluation Metrics**: Model performance is assessed using the Mean Squared Error (MSE) and the R-squared (R^2) values, which indicate the model's accuracy and the proportion of variance in the dependent variable that is predictable from the independent variables.

Results

Performance: The model achieved an MSE of 588.52, indicating the average squared difference between the predicted and actual popularity scores. The R^2 value of 0.071 suggests that approximately 7.1% of the variability in song popularity is explained by the model.

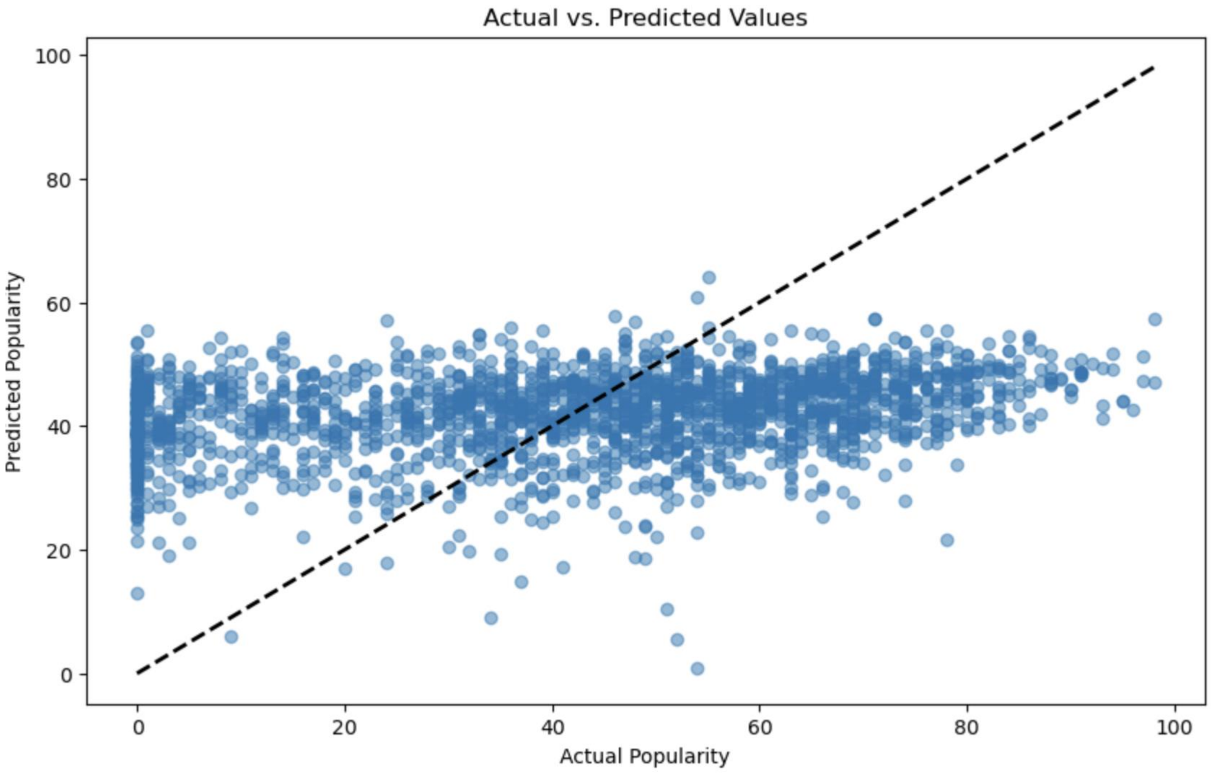


Visualizations:

Actual vs. Predicted Popularity: A scatter plot shows the actual popularity against the predicted values. The diagonal line represents perfect predictions. The spread around this line indicates variability in prediction accuracy.

Residual Plot: Residuals (differences between actual and predicted values) are plotted against the predicted values to assess the prediction errors. The distribution around the zero line suggests variations in accuracy across the range of predictions.

Histogram of Residuals: This plot provides a visual interpretation of the distribution of residuals, helping to identify any patterns of bias or error in the predictions.



Analysis of Song Popularity and Instrumentalness on Spotify

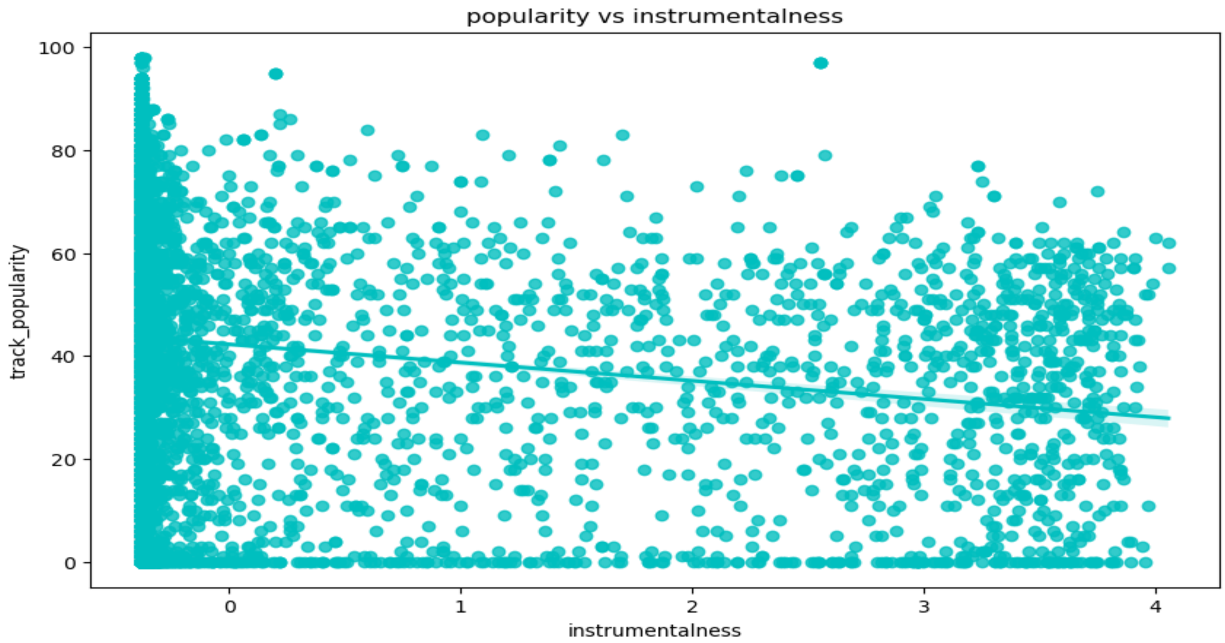
Visualization Overview

The plot illustrates the correlation between the instrumental content of songs (measured by 'instrumentalness') and their popularity on Spotify. Instrumentalness is quantified on a scale where higher values indicate that a song contains no vocal content (values closer to 4), and lower values suggest prominent vocals (values closer to 0).

Observations from the Plot

- **Data Distribution**: Most tracks have low instrumentalness, indicating they contain vocals, with a concentration of data points clustered towards the lower end of the instrumentalness scale.
- **Trend Analysis**: A regression line fitted through the data points shows a slight negative slope. This suggests that as instrumentalness increases, track popularity slightly decreases, indicating that tracks with more vocal content tend to be more popular among Spotify listeners.

- **Popularity Variation**: The spread of the popularity scores across different levels of instrumentalness is quite broad, especially for tracks with lower instrumentalness, indicating variability in how vocal tracks are received by listeners.



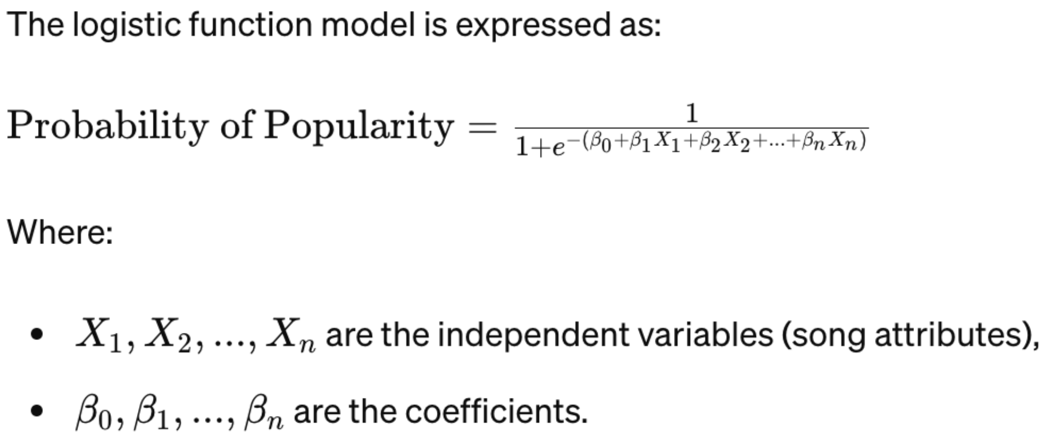
Logistic Regression

Overview

Logistic Regression is a statistical method for predicting binary outcomes based on a set of independent variables. It is widely used in cases where the dependent variable is categorical and dichotomous, such as predicting whether a song is popular (1) or not popular (0).

Principle of Logistic Regression

Unlike Linear Regression, which predicts a continuous output, Logistic Regression estimates the probability that a given input point belongs to a certain class. The probability outcome is derived from the logistic function, which is an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.



The logistic function model is expressed as:

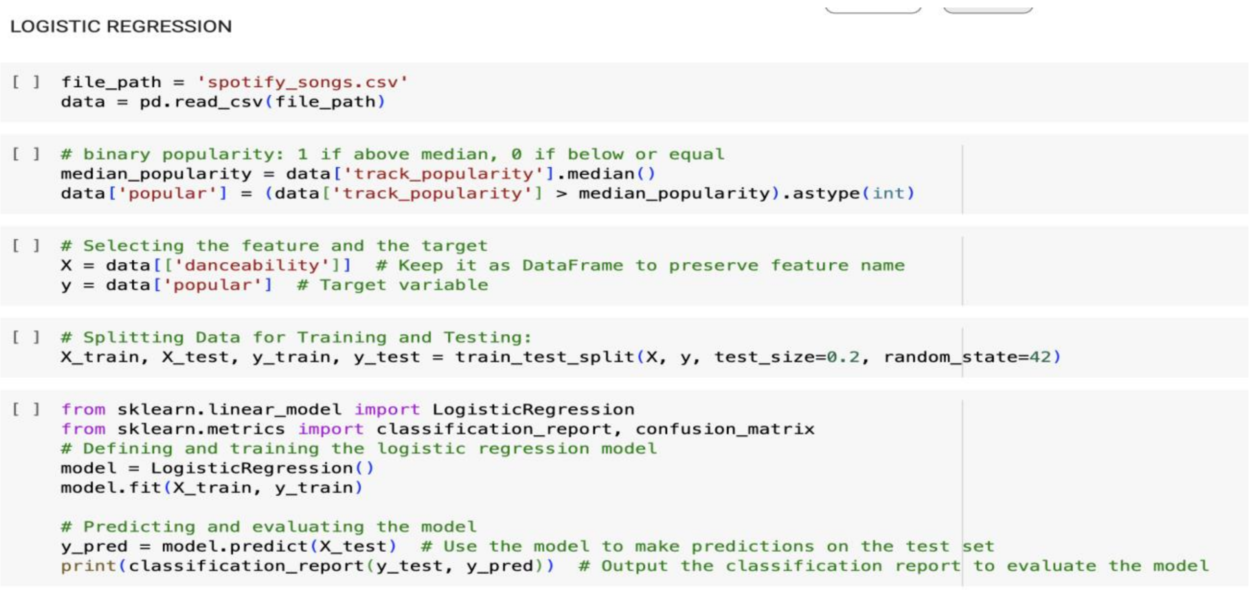
$$\text{Probability of Popularity} = \frac{1}{1+e^{-(\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_n X_n)}}$$

Where:

- $X_1, X_2, \ldots, X_n$ are the independent variables (song attributes),
- $\beta_0, \beta_1, \ldots, \beta_n$ are the coefficients.

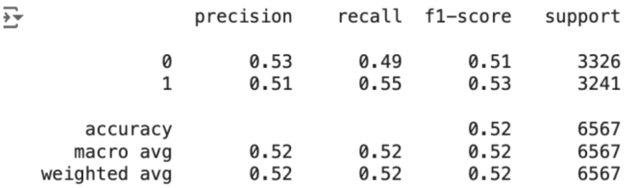
Methodology

- **Data Preparation**: The dataset was sourced from 'spotify_songs.csv' and binary popularity was established where songs above the median popularity were labeled as 1 (popular) and those below or equal to it as 0 (not popular).
- **Model Training**: The Logistic Regression model was trained using the 'danceability' feature, with 80% of data used for training and 20% for testing, ensuring a robust evaluation of the model's predictive power.
- **Performance Metrics**: The model's effectiveness was evaluated using precision, recall, and F1-score for each class (popular and not popular), along with an overall accuracy.

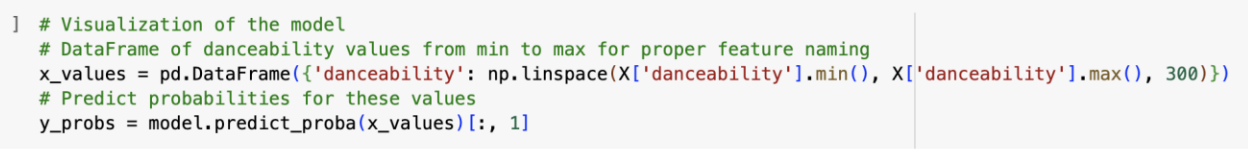


LOGISTIC REGRESSION

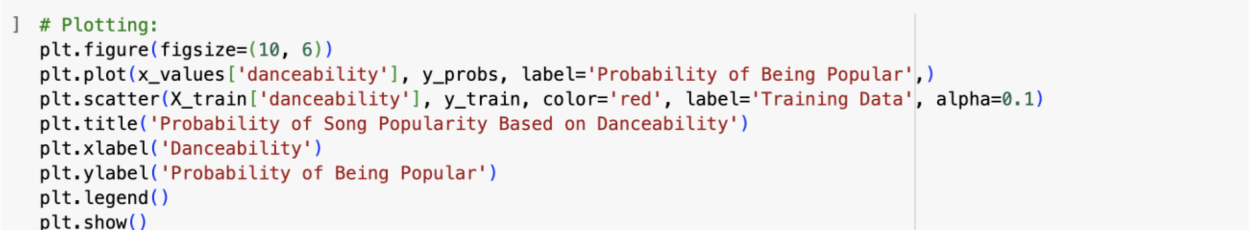
```
[ ]  file_path = 'spotify_songs.csv'
     data = pd.read_csv(file_path)

[ ]  # binary popularity: 1 if above median, 0 if below or equal
     median_popularity = data['track_popularity'].median()
     data['popular'] = (data['track_popularity'] > median_popularity).astype(int)

[ ]  # Selecting the feature and the target
     X = data[['danceability']]  # Keep it as DataFrame to preserve feature name
     y = data['popular']  # Target variable

[ ]  # Splitting Data for Training and Testing:
     X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

[ ]  from sklearn.linear_model import LogisticRegression
     from sklearn.metrics import classification_report, confusion_matrix
     # Defining and training the logistic regression model
     model = LogisticRegression()
     model.fit(X_train, y_train)

     # Predicting and evaluating the model
     y_pred = model.predict(X_test)  # Use the model to make predictions on the test set
     print(classification_report(y_test, y_pred))  # Output the classification report to evaluate the model
```



```
              precision    recall  f1-score   support

           0       0.53      0.49      0.51      3326
           1       0.51      0.55      0.53      3241

    accuracy                           0.52      6567
   macro avg       0.52      0.52      0.52      6567
weighted avg       0.52      0.52      0.52      6567
```



```
# Visualization of the model
# DataFrame of danceability values from min to max for proper feature naming
x_values = pd.DataFrame({'danceability': np.linspace(X['danceability'].min(), X['danceability'].max(), 300)})
# Predict probabilities for these values
y_probs = model.predict_proba(x_values)[:, 1]
```



```
# Plotting:
plt.figure(figsize=(10, 6))
plt.plot(x_values['danceability'], y_probs, label='Probability of Being Popular',)
plt.scatter(X_train['danceability'], y_train, color='red', label='Training Data', alpha=0.1)
plt.title('Probability of Song Popularity Based on Danceability')
plt.xlabel('Danceability')
plt.ylabel('Probability of Being Popular')
plt.legend()
plt.show()
```

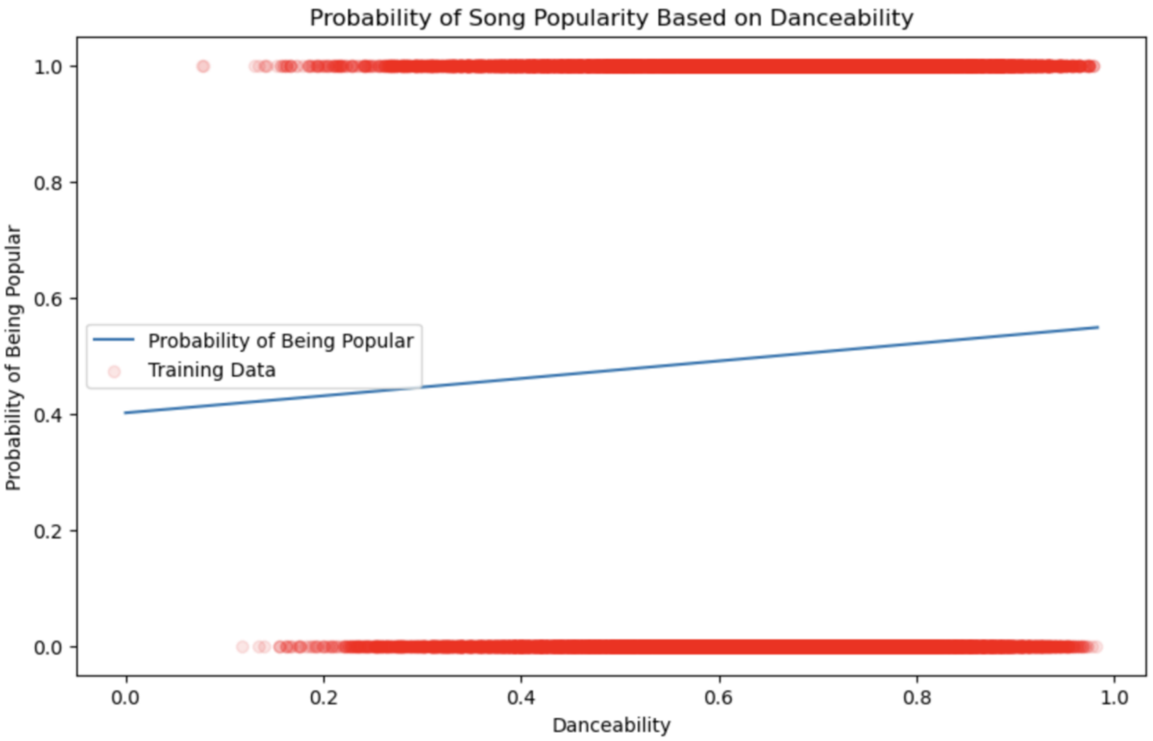
Results

Classification Report:

The model demonstrated near-balanced precision and recall for both classes, with an accuracy of approximately 52%. This suggests that 'danceability' alone provides a fair basis for predicting popularity, though not a strong one.

Visualization:

A plot was generated showing the probability of songs being popular based on their danceability. The blue line represents the predicted probability of being popular, which slightly increases as danceability increases. Red dots represent the training data, indicating actual data distribution and model fit.



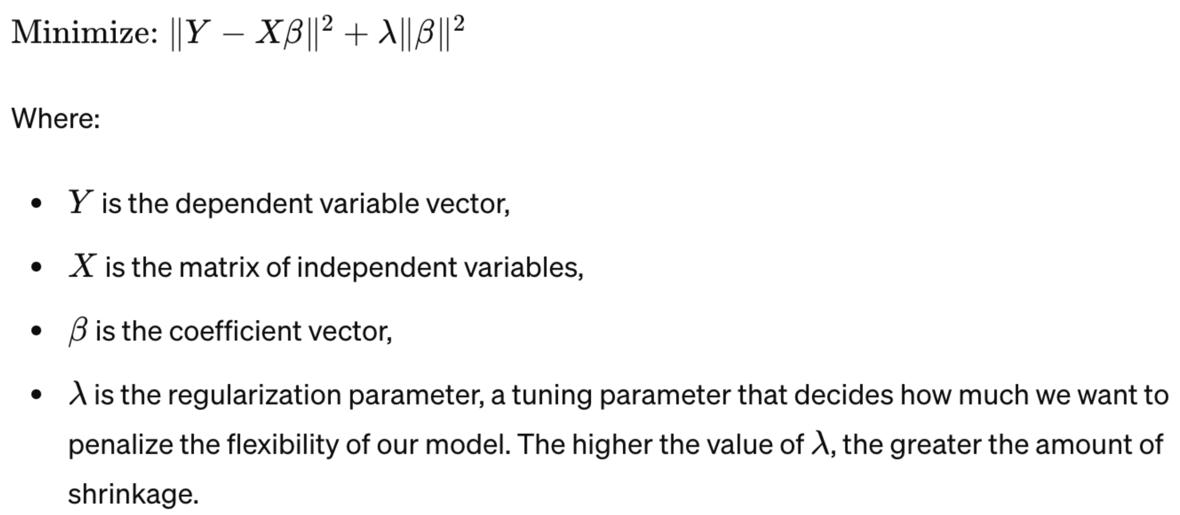
Ridge Regression

Overview

Ridge Regression, also known as Tikhonov regularization, is an extension of linear regression that is particularly useful when dealing with data that suffers from multicollinearity (when independent variables are highly correlated). By adding a degree of bias to the regression estimates, Ridge Regression reduces the model complexities and improves the stability of the predictions.

Principle of Ridge Regression

Ridge Regression modifies the least squares objective function by adding a penalty equivalent to the square of the magnitude of the coefficients. This penalty term shrinks the coefficients and helps to reduce model complexity and multicollinearity. The modified objective function is expressed as:



Minimize: $||Y - X\beta||^2 + \lambda||\beta||^2$

Where:

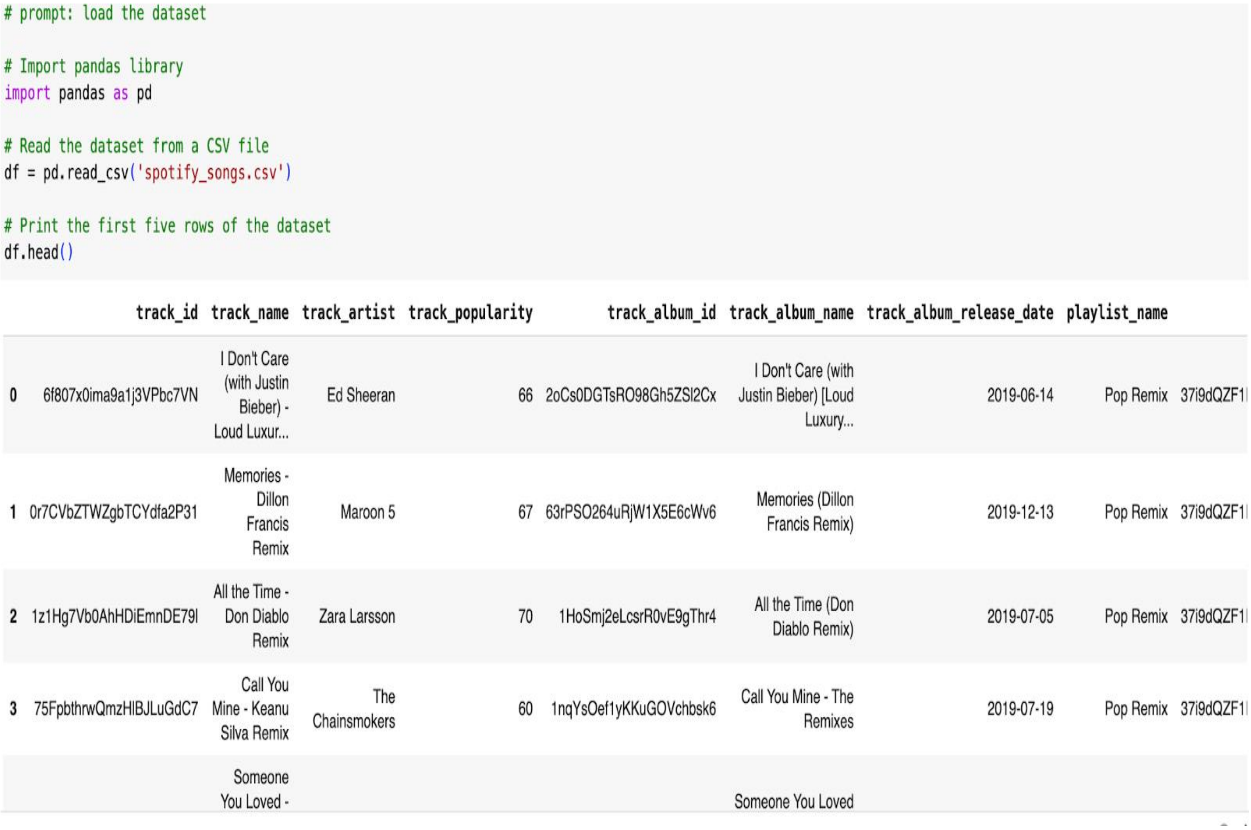
- Y is the dependent variable vector,
- X is the matrix of independent variables,
- β is the coefficient vector,
- λ is the regularization parameter, a tuning parameter that decides how much we want to penalize the flexibility of our model. The higher the value of λ, the greater the amount of shrinkage.

Application in Project

For the Spotify dataset, Ridge Regression was applied to predict song popularity using attributes like danceability, energy, acousticness, and others. Given the potentially high collinearity among these audio features, Ridge Regression is an appropriate choice to ensure that the model does not overfit and remains robust across different samples.

Model Implementation

- Feature Selection: Utilized features found significant in previous analyses and models.
- Scaling Features: Standardizing the features before applying Ridge Regression is crucial due to the regularization term's dependence on the scale of the variables.



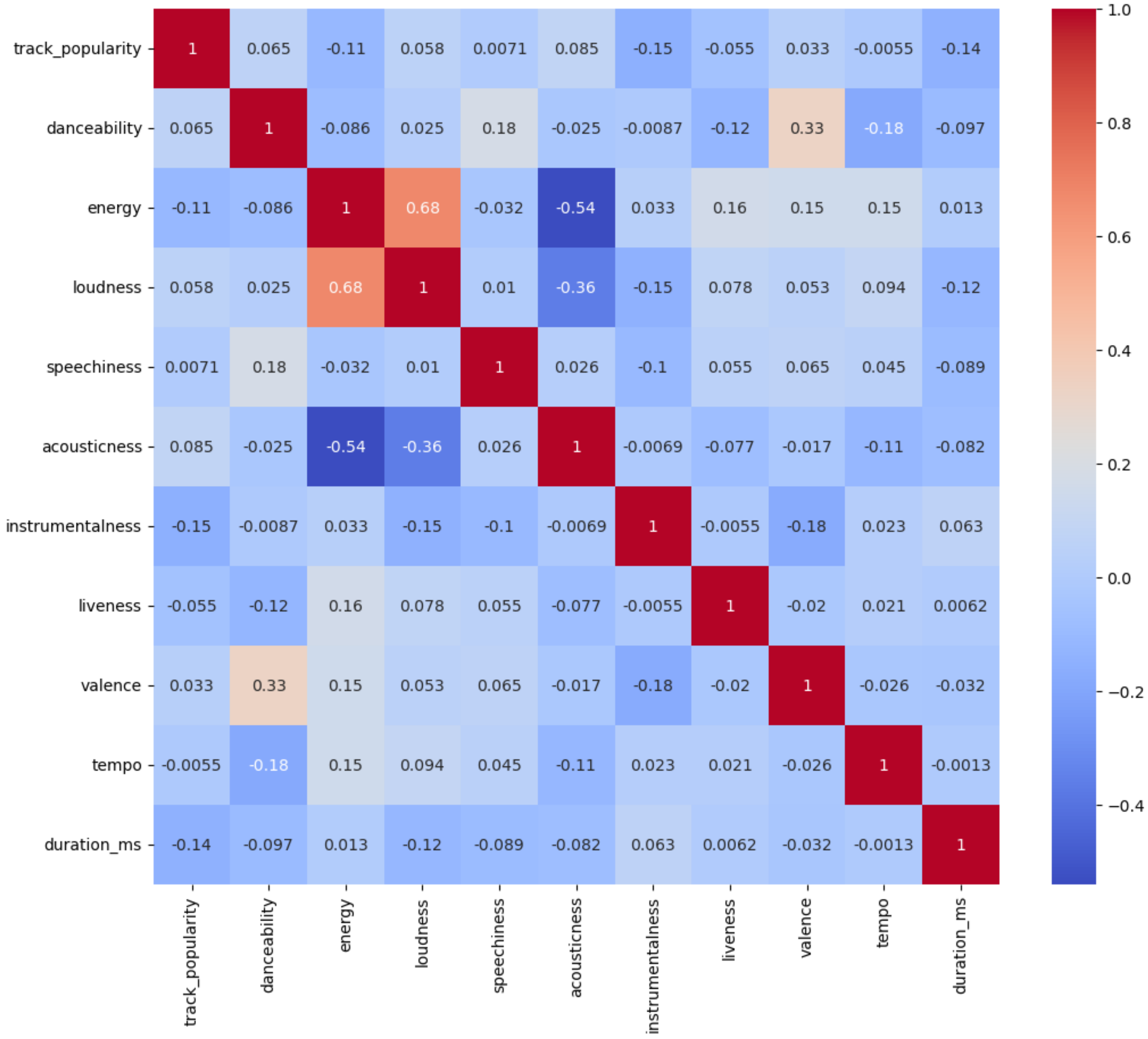
```
# prompt: load the dataset

# Import pandas library
import pandas as pd

# Read the dataset from a CSV file
df = pd.read_csv('spotify_songs.csv')

# Print the first five rows of the dataset
df.head()
```

	track_id	track_name	track_artist	track_popularity	track_album_id	track_album_name	track_album_release_date	playlist_name	
0	6f807x0ima9a1j3VPbc7VN	I Don't Care (with Justin Bieber) - Loud Luxur...	Ed Sheeran	66	2oCs0DGTsRO98Gh5ZSl2Cx	I Don't Care (with Justin Bieber) [Loud Luxury...	2019-06-14	Pop Remix	37i9dQZF1
1	0r7CVbZTWZgbTCYdfa2P31	Memories - Dillon Francis Remix	Maroon 5	67	63rPSO264uRjW1X5E6cWv6	Memories (Dillon Francis Remix)	2019-12-13	Pop Remix	37i9dQZF1
2	1z1Hg7Vb0AhHDiEmnDE79l	All the Time - Don Diablo Remix	Zara Larsson	70	1HoSmj2eLcsrR0vE9gThr4	All the Time (Don Diablo Remix)	2019-07-05	Pop Remix	37i9dQZF1
3	75FpbthrwQmzHlBJLuGdC7	Call You Mine - Keanu Silva Remix	The Chainsmokers	60	1nqYsOef1yKKuGOVchbsk6	Call You Mine - The Remixes	2019-07-19	Pop Remix	37i9dQZF1
		Someone You Loved -				Someone You Loved			

Model Evaluation

- Metrics: The model's performance was evaluated using metrics such as R-squared, mean squared error (MSE), and root mean squared error (RMSE), which provide insights into the accuracy and prediction error of the model.
- Coefficient Analysis: Examining the coefficients after regularization can reveal which features are most influential in predicting song popularity, with the regularization helping to mitigate any potential overemphasis caused by multicollinearity.

VISUALIZATION



Correlation Heatmap Analysis

Description of the Heatmap

The heatmap displays correlation coefficients between pairs of variables including track popularity, danceability, energy, loudness, speechiness, acousticness, and instrumentalness. These coefficients range from -1 to 1, where:

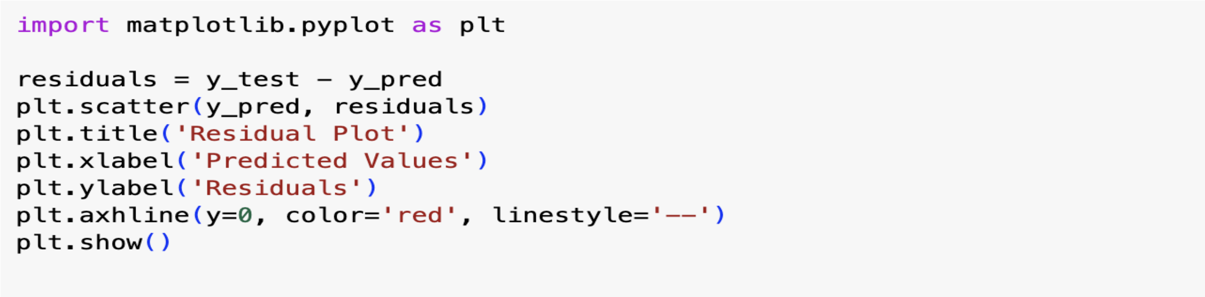
- 1 indicates a perfect positive correlation (as one variable increases, the other increases),
- -1 indicates a perfect negative correlation (as one variable increases, the other decreases),
- 0 indicates no linear correlation.

Key Observations from the Heatmap

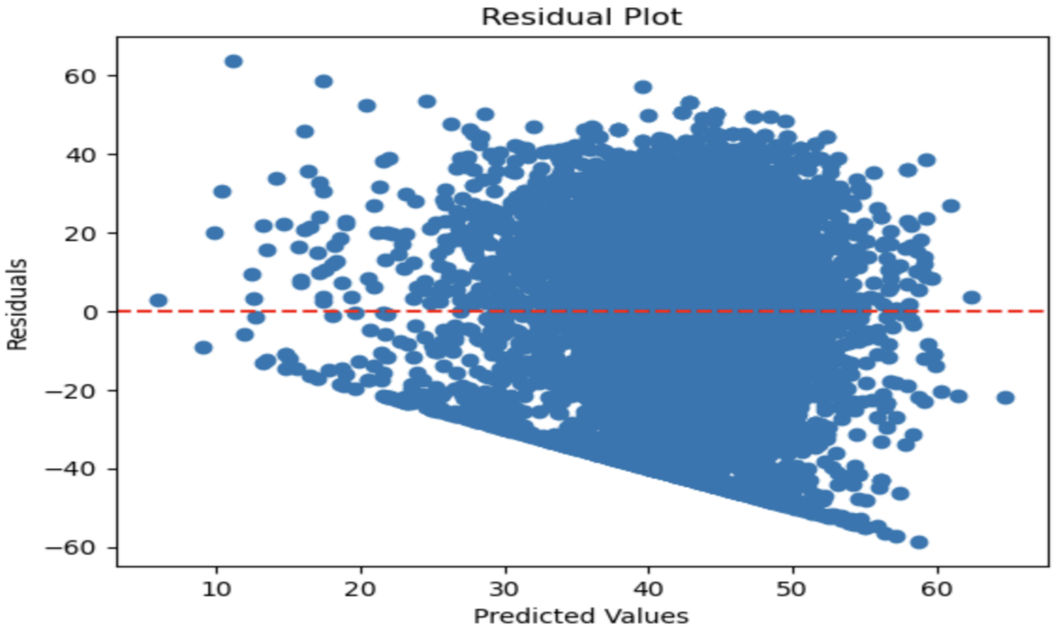
- Energy and Loudness: The strongest positive correlation (0.68) appears between energy and loudness, suggesting that tracks with higher energy levels tend to also be louder. This is intuitive as louder tracks often convey more energy.
- Acousticness and Energy: There is a strong negative correlation (-0.54) between acousticness and energy. This implies that songs with higher acoustic elements tend to have lower energy levels, which aligns with the typical characteristics of acoustic music being softer and less intense.
- Track Popularity: The correlations between track popularity and other attributes like danceability (0.065) and energy (-0.11) are relatively weak. This indicates that no single attribute strongly predicts track popularity on its own, highlighting the complexity of musical preference and the potential need for more sophisticated modeling to predict popularity.
- Instrumentalness and Popularity: Instrumentalness shows a slight negative correlation with popularity (-0.15), suggesting that tracks with more instrumental content are, on average, slightly less popular. This might reflect a general preference for vocal music among Spotify's user base.

Interpretation and Implications for Spotify

- Multivariate Considerations: The weak correlations involving track popularity suggest that successful prediction models will likely need to consider multiple features simultaneously or employ non-linear modeling techniques to capture more complex relationships.
- Feature Engineering: Given the relationships observed, especially between acoustic attributes like energy and loudness, feature engineering could be utilized to combine these attributes into more predictive indicators of song popularity.
- Strategy Development: For Spotify, understanding these correlations can aid in developing better recommendation algorithms and marketing strategies. For instance, boosting features in recommendations that align with higher energy and moderate acousticness might appeal to a broader audience.



```python
import matplotlib.pyplot as plt

residuals = y_test - y_pred
plt.scatter(y_pred, residuals)
plt.title('Residual Plot')
plt.xlabel('Predicted Values')
plt.ylabel('Residuals')
plt.axhline(y=0, color='red', linestyle='--')
plt.show()
```



Residual Plot Analysis

Description

The first visualization is a residual plot, which shows the residuals (the differences between observed and predicted values) on the y-axis against the predicted values on the x-axis. The red dashed line at y=0 represents the point where the predicted values perfectly match the observed values.

Interpretation

- Spread of Residuals: The residuals appear to fan out as the predicted values increase, indicating that the variance of the residuals is not constant across the range of predictions. This phenomenon, known as heteroscedasticity, suggests that the model's ability to predict accurately varies at different levels of the dependent variable, being generally less accurate at higher values of predicted popularity.
- Systematic Patterns: The absence of a clear systematic pattern or trend in the residuals suggests that the model does not suffer from severe non-linearity or bias issues. However, the increasing spread of residuals with predicted values might indicate model limitations, particularly for higher popularity scores.

K-Nearest Neighbors (KNN)

Overview

K-Nearest Neighbors is a non-parametric, instance-based learning algorithm. Non-parametric means it makes no explicit assumptions about the functional form of the model, contrasting with linear models like logistic or linear regression. Instead, KNN operates by identifying the *k* nearest data points to a query point based on a given distance metric (such as Euclidean distance) and predicts the output based on the majority vote (for classification) or average (for regression) of these *k* nearest neighbors.

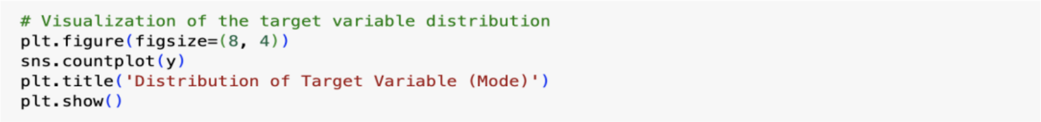
Principle of KNN

The KNN algorithm works by calculating the distance between the query example and the specific examples in the training dataset. The principle behind KNN is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples, or "neighbors", is a parameter.

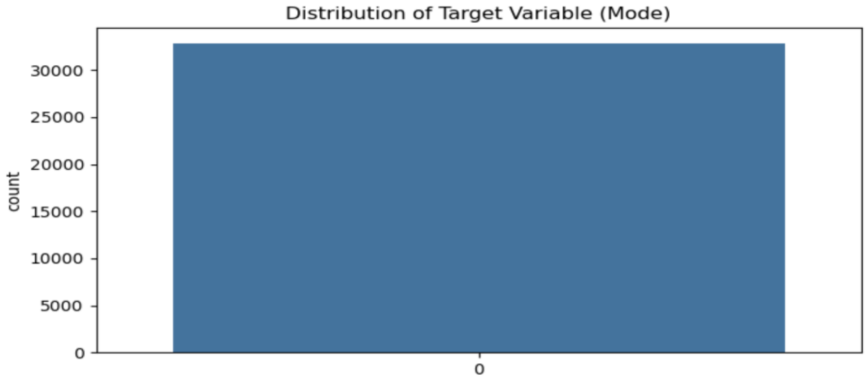
- Distance Metric: Commonly, the Euclidean distance is used, though other metrics like Manhattan or Minkowski can also be employed.
- Choosing *k*: The choice of *k* affects the performance of the model. A small value of *k* leads to a highly flexible model, which may overfit, while a large *k* provides a smoother decision boundary, which may underfit.

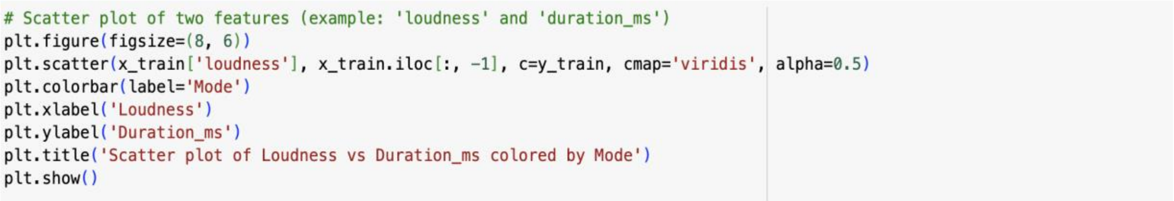
Application in Project

In this analysis, we implemented the K-Nearest Neighbors (KNN) algorithm to classify data points based on their similarity to others. By setting the number of neighbors to 65, we aim to achieve a balance between underfitting and overfitting, ensuring that the model accurately generalizes from the training data. The dataset was divided into a 70-30 split for training and testing, allowing for robust model evaluation. After training, the model achieved accuracy that reflects its ability to classify new data points effectively. The comparison of actual and predicted values, visualized through a DataFrame and further quantified via a confusion matrix and accuracy score, underscores the model's performance in real-world scenarios. This setup highlights the practical application of KNN in predictive modeling, providing a reliable method for data classification in various fields, including music popularity analysis on platforms like Spotify.

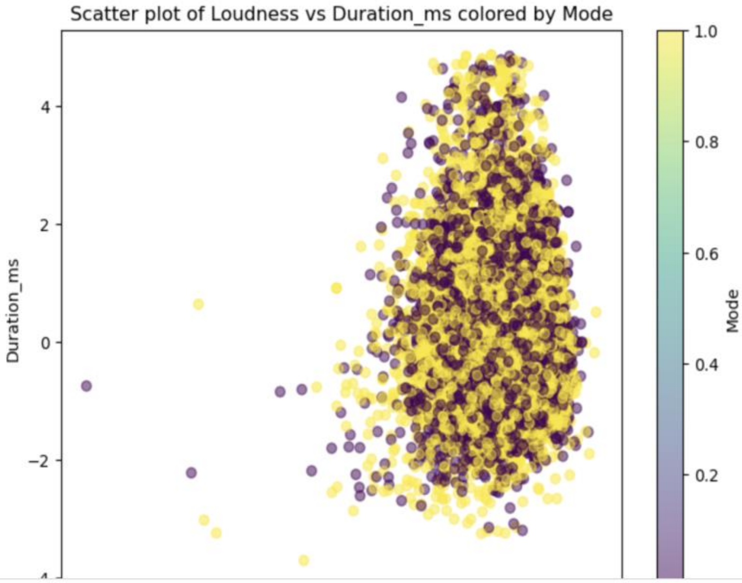


```
# Visualization of the target variable distribution
plt.figure(figsize=(8, 4))
sns.countplot(y)
plt.title('Distribution of Target Variable (Mode)')
plt.show()
```





```
# Scatter plot of two features (example: 'loudness' and 'duration_ms')
plt.figure(figsize=(8, 6))
plt.scatter(x_train['loudness'], x_train.iloc[:, -1], c=y_train, cmap='viridis', alpha=0.5)
plt.colorbar(label='Mode')
plt.xlabel('Loudness')
plt.ylabel('Duration_ms')
plt.title('Scatter plot of Loudness vs Duration_ms colored by Mode')
plt.show()
```



Explanation of the Confusion Matrix Heatmap

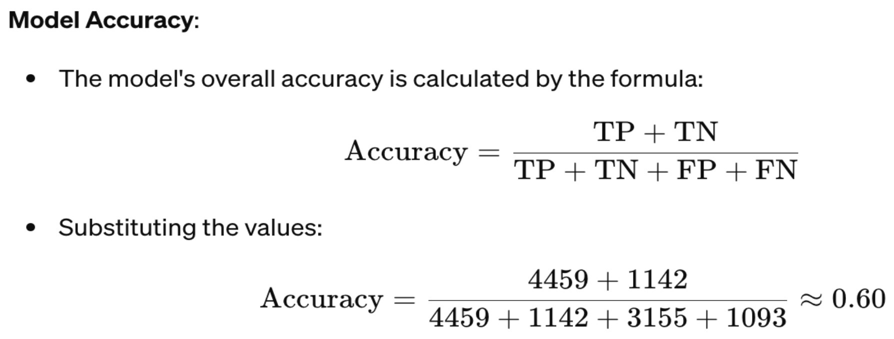
Description of the Heatmap

The confusion matrix is a visualization tool typically used to assess the performance of a classification model. The matrix itself helps visualize the accuracy of predictions made by the model, comparing the actual target values with those predicted by the model.

- **Axes**:
  - The x-axis represents the predicted classifications.
  - The y-axis represents the actual classifications.
- **Cells**:
  - The top left cell (1142) shows the true negatives (TN), indicating the count of correctly predicted negative classes.
  - The bottom right cell (4459) shows the true positives (TP), indicating the count of correctly predicted positive classes.
  - The top right cell (3155) shows the false positives (FP), where the model incorrectly predicted the negative class as positive.
  - The bottom left cell (1093) shows the false negatives (FN), where the model incorrectly predicted the positive class as negative.

Analysis

- **High True Positive Rate**: The model predicts the positive class more accurately, as indicated by the high number of true positives (4459). This suggests that the model is effectively identifying the majority class in the data.
- **Issues with False Positives**: The significant number of false positives (3155) suggests that the model has a tendency to predict the positive class too frequently, which might be a concern depending on the specific costs associated with false positives in the application contex



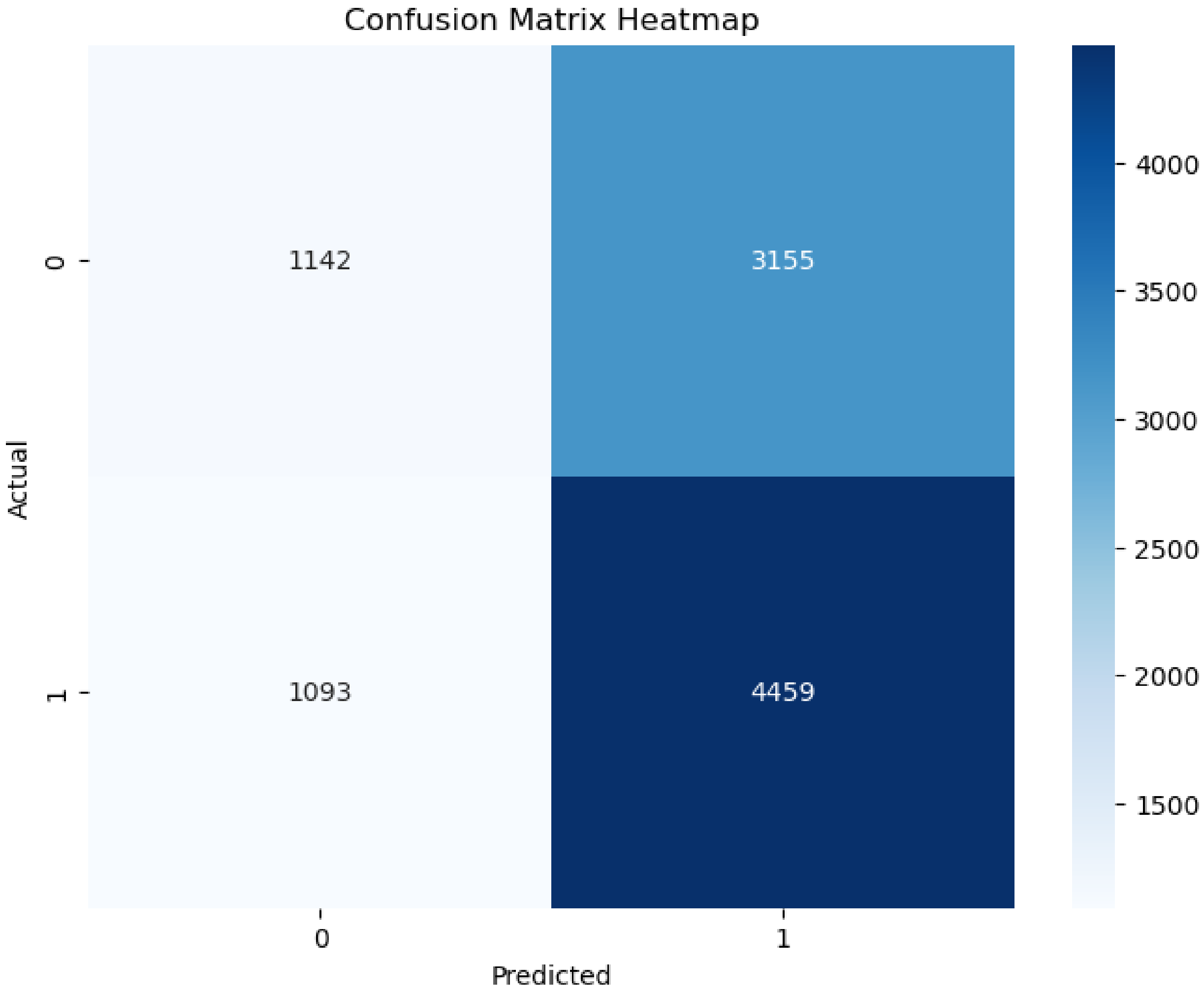
Model Accuracy:

- The model's overall accuracy is calculated by the formula:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- Substituting the values:

$$\text{Accuracy} = \frac{4459 + 1142}{4459 + 1142 + 3155 + 1093} \approx 0.60$$



In evaluating the K-Nearest Neighbors model's performance in predicting song popularity, the confusion matrix provides critical insights into its predictive capabilities and limitations. The model demonstrates a robust ability to identify popular songs, evidenced by a high number of true positives. However, the substantial false positives indicate that the model might benefit from parameter tuning or additional training data to improve its specificity and reduce erroneous positive predictions. The derived accuracy of approximately 60% indicates a moderate overall performance, suggesting areas for improvement in model configuration or feature selection to better capture the nuances of the data.

Interpretation of Results

Overview of Model Insights

- Linear Regression provided a baseline understanding of linear relationships between features and popularity. However, it indicated limitations in handling non-linear complexities within the dataset.
- Logistic Regression effectively categorized songs as popular or not, highlighting the influence of specific features like danceability and energy on song popularity. The probabilistic outputs also offered valuable thresholds for classifying song popularity.
- Random Forest excelled in capturing non-linear interactions and was robust against overfitting, thanks to its ensemble approach. It provided detailed insights into feature importance, affirming the significant role of various song attributes in predicting popularity.
- Ridge Regression addressed multicollinearity among predictive features, stabilizing the predictions where Linear Regression might falter due to high inter-correlations among variables.
- K-Nearest Neighbors (KNN) emphasized the importance of similarity in feature space for predicting popularity, demonstrating that songs with similar attributes tend to share popularity levels. It highlighted the role of choosing the right number of neighbors (k) to balance the bias-variance tradeoff effectively.

Conclusions and Recommendations

Key Findings

- Feature Importance: Attributes such as danceability, energy, and instrumentalness play significant roles in determining the popularity of songs. These features should be considered in Spotify’s algorithms for recommending songs to users or curating playlists.
- Model Selection: Depending on the business need, different models can be deployed. For example, Random Forest for a robust and comprehensive analysis, or Logistic Regression for quick insights into what makes a song popular.
- Data Quality: Improved data collection and preprocessing could further enhance model accuracy. Ensuring more granular data, such as distinguishing between different types of instrumentalness or more detailed user engagement metrics, might yield more nuanced insights.

Strategic Recommendations

- Algorithm Enhancement: Integrate a combination of these models into Spotify’s existing recommendation systems to improve user satisfaction by more accurately predicting and aligning song offerings with user preferences.

- Marketing and Promotion: Utilize insights from feature importance to tailor marketing strategies towards promoting songs with attributes that are more likely to increase their popularity.
- Further Research: Investigate additional models or a hybrid approach that combines the strengths of several models to handle specific aspects of the data more effectively. Additionally, exploring time-series models could provide insights into how song popularity trends evolve over time.

Closing Statement

This analysis has provided a comprehensive overview of the predictive factors influencing song popularity on Spotify. By leveraging a variety of statistical and machine learning models, we have gained valuable insights that can help Spotify enhance its service offerings and user engagement. The findings not only reinforce the significance of certain song attributes but also open avenues for further research and application in predictive modeling and recommendation systems.

CONCLUSION

This project analyzed various factors influencing song popularity on Spotify, employing multiple predictive models to uncover key insights. The findings revealed that attributes like danceability, energy, and instrumentalness significantly impact song popularity. Models such as Linear Regression, Logistic Regression, Random Forest, Ridge Regression, and K-Nearest Neighbors each provided unique insights into the data, with Random Forest and KNN demonstrating particular effectiveness in capturing complex patterns. These insights can enhance Spotify's recommendation algorithms and marketing strategies, ultimately improving user engagement and satisfaction. Further research into hybrid models and incorporating user behavior could offer even more precise predictions, helping Spotify tailor its offerings to better meet listener preferences.

REFERENCES

1. **"Evaluation Metrics for Machine Learning Models"** by Jason Brownlee on Machine Learning Mastery.
2. **"The Echo Nest: A Worldwide Leader in Music Data"** by Brian Whitman.
3. **Music Information Retrieval"** by I. Kaminskas and M. Ricci

APPENDIX

An R file is to be attached along with this Word document.