Contents

1. Introduction

This project focuses on leveraging machine learning techniques to analyze, predict, and cluster music data, uncovering insights into genre classification, song popularity, and playlist generation. By addressing key challenges in music analytics, the project aims to provide data-driven solutions for enhancing the music experience.

Objective: To understand the features and characteristics of trendy music and album covers using machine learning techniques.

Research Questions:

- What characteristics are most influential in determining a song's popularity?
- Can the popularity of a song be predicted, given its attributes and metadata?
- Can songs be classified into genres based on their visual album cover?
- Can we cluster songs by audio features and identify patterns (example: mood) to generate playlists?

Further, the below hypotheses are to be validated through the research results:

- **Hypothesis 1:** Music genre can be classified based on visual album cover.
  - **Problem 1:** Determine relationships and trends between album cover and genres.
  - **Problem 2:** Predict song genres based on album cover using classification.
- **Hypothesis 2:** Song popularity can be predicted based on danceability, loudness, and other features.
  - **Problem 3:** Predict the potential popularity of a song (regression problem) based on its features.
- **Hypothesis 3:** Songs can be clustered by audio features to reveal patterns.
  - **Problem 4**: identify audio features to cluster songs and explore how these clusters vary to create diverse playlists.

2. Methodology

The tasks for this work were divided for each person to handle a model from start to finish so we have several notebooks containing parts for the data analysis & explain the structure of the experimental setup part . GitHub readme file contains a pipeline to explain the flow of the portfolio.

The below datasets will be used for this research:

Spotify Tracks:

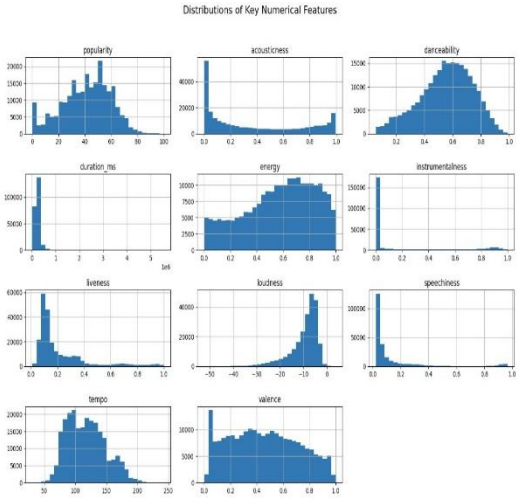
https://www.kaggle.com/datasets/zaheenhamidani/ultimate-spotify-tracks-db

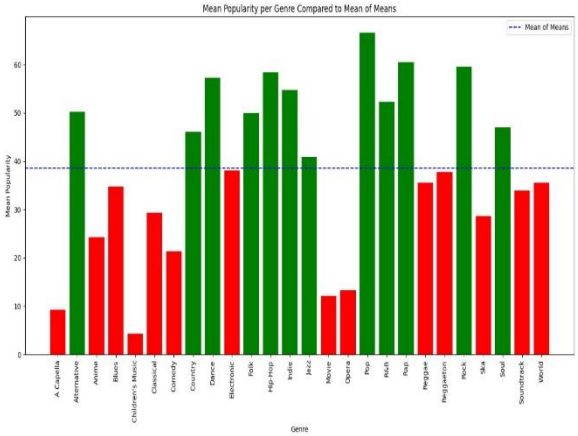
Features:

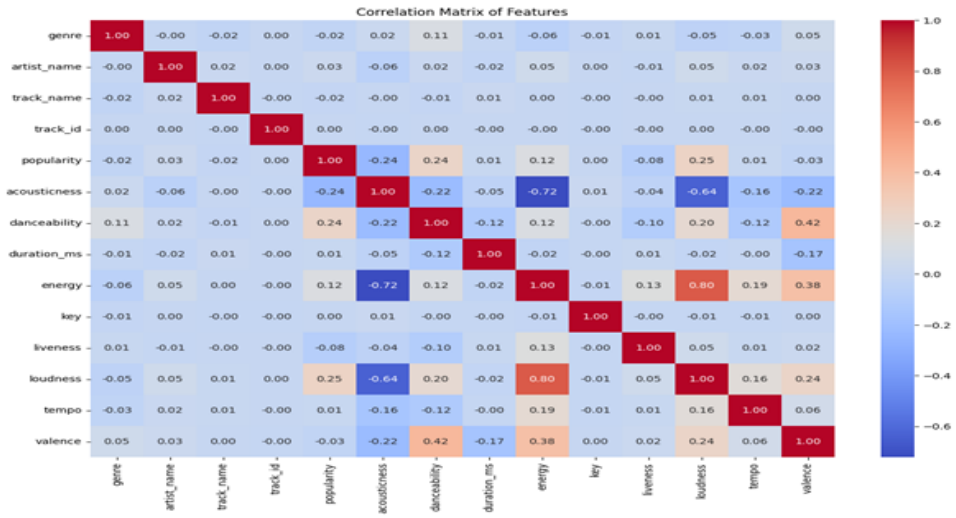
Genre, Popularity, Acousticness, Danceability, Duration (ms), Energy, Instrumentalness, Key, Liveness, Loudness, Mode, Speechiness, Tempo, Time Signature, Valence

General Data analysis & insights:

- Popularity is positively correlated with features such as loudness, energy, and danceability.
- Acousticness shows a negative correlation with popularity.
- The most common genres in the dataset are Comedy, Soundtrack, and Indie.
- Songs in major mode dominate the dataset, often associated with a brighter or happier tone.
- Popularity tends to fall within mid-range values, whereas danceability and energy exhibit a more even distribution.







Album covers:

https://www.kaggle.com/datasets/michaeljkerr/20k-album-covers-within-20-genres

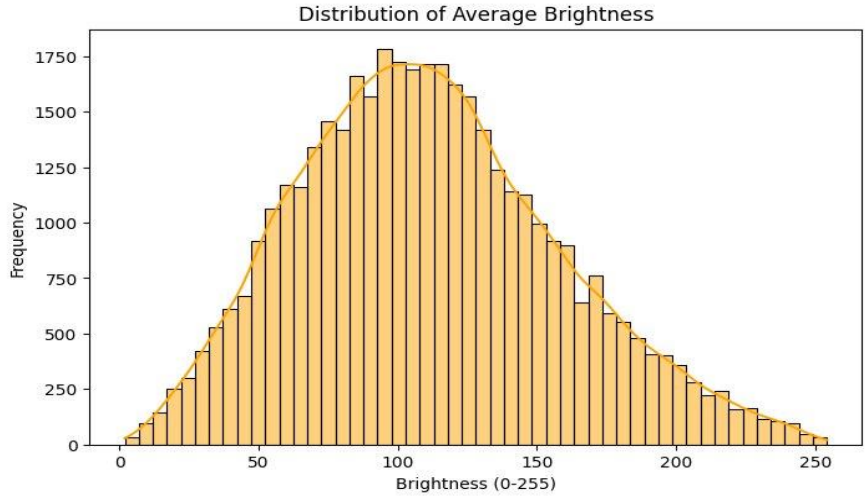
Features: Genre folders , inside each folder set of album cover images

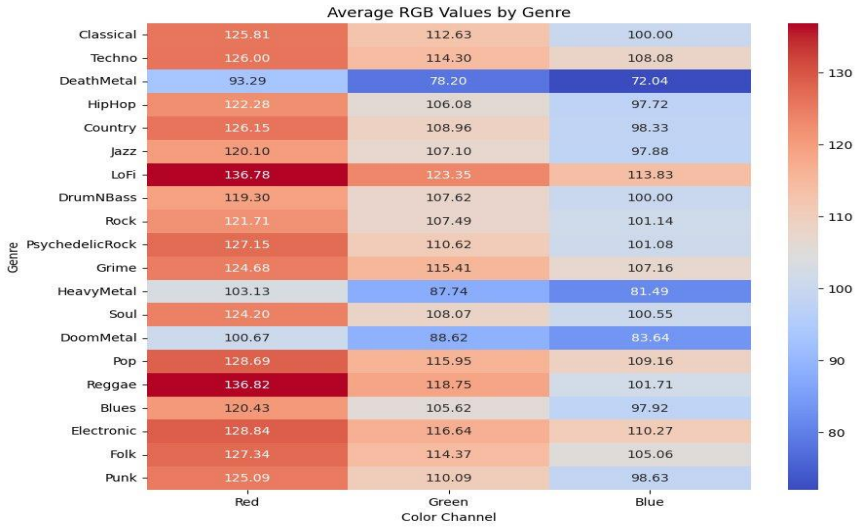
Data Cleansing Outlines:

Remove Low-Quality Images, Format Consistency (JPEG , PNG ..) , Resizing Images (One Aspect Ratio), Remove Duplicates, Balancing genre's datasets

Insights from data Analysis:

- Warm Colors: Genres like LoFi and Reggae often feature red tones.
- Darker Tones: Genres such as DeathMetal, DoomMetal, and HeavyMetal commonly use darker, more muted colors with low RGB values.
- Natural Color Palette: Genres like Classical, Jazz, HipHop, Rock, and Country tend to exhibit balanced and natural color tones.
- Brightness distribution follows a near-normal (bell curve) pattern.
- Most album covers exhibit moderate brightness levels.
- Extremes in brightness are rare.
- The dataset was balanced prior to cleansing , All album covers share a single aspect ratio.





3. Related work

Similar datasets have been previously used on the below studies:

- Spotify Data Analysis and Song Popularity Prediction

  https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4793176

- Evolution of global music trends: An exploratory and predictive approach based on Spotify data

  https://www.sciencedirect.com/science/article/pii/S1875952122000593

- Genre Classification via Album Cover

  https://cs230.stanford.edu/projects_spring_2020/reports/38953572.pdf

4. Experimental Setup

4.1 Prediction (Regression):

4.1.1 Linear Regression

Data Processing:

- Remove the null values
- Remove duplicates
- Remove the genre “Children’s Music Aaa” records as it has many irrelevant songs that are not part of that genre
- Standardize/normalize data

Data Analysis:

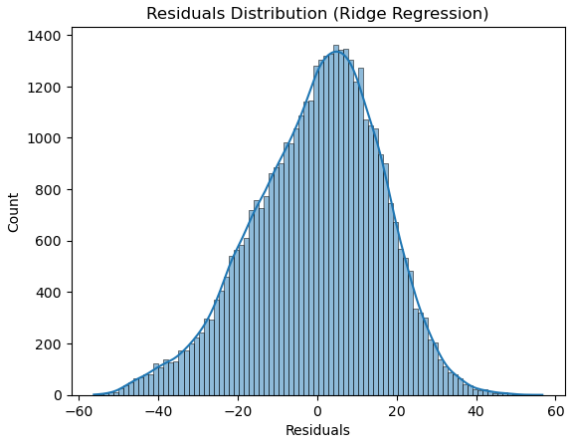
Based on correlation matrix, loudness, danceability, energy, acousticness, tempo and duration were highly correlated with popularity. As a result, they were the selected features for linear regression.

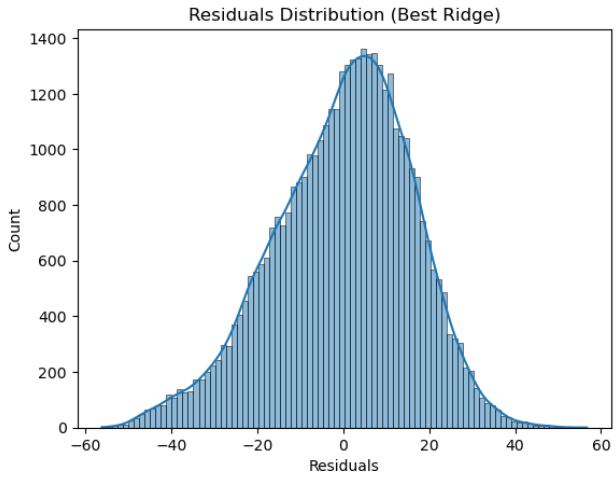
Experimental Setup:

- Split data between train and test data
- Standardize the features using scaler
- Hyper parameter tuning for ridge regression
- Regularization of linear regression /Best Ridge model and mean squared error for best ridge model calculate
- Residual analysis
- Cross validation

Results:

- Mean Squared Error (Ridge): 259.3152630035354
- Cross-Validated MSE (Ridge): 255.92104593176796
- Best Ridge Alpha: {'alpha': 10}
- Mean Squared Error (Best Ridge): 259.31526059872346





- Results show MSE is very high despite hyperparameters tuning, poly models are more suitable for this data compared to linear regression.

4.1.2Decision Tree

Data Processing:

- Identify the overview of the data to understand the type of data and the structure.
- Remove the duplicate rows and verify the missing values in the data.
- To focus on mainstream music, remove "Children's Music" from the genre list.
- Categorical and numerical data were found in the data. To process identifying the feature's correlation with popularity, we transformed the categorical columns into numerical
- Apply the IQR method with a multiplier of 3.0 to remove extreme outliers, keeping the dataset focused on meaningful data for better accuracy.
- Standardize and normalize the features for balanced model training.

Data Analysis

To understand the feature distribution, relationship, and important features in the data, we have visualized the data using distribution plots, scatter plots, and correlation matrix.

Analysis:

- **Distribution plots**: show that loudness, energy, and danceability are important for determining song popularity.
- **Scatter plots:** show that based on the frequency, the genres are an important feature.
- **Correlation Matrix: The** distributions of danceability, energy, and loudness show that popular songs often fall within moderate levels of these features.

Experimental Setup

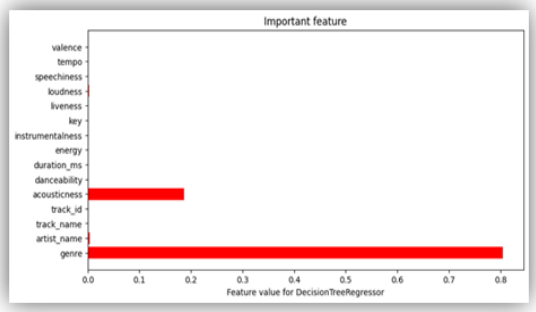
- With a high importance score near 0.8, the feature importance chart indicates that genre has the greatest influence on song popularity prediction.
- we have created a set of data as 90%-10%, 80%-20%, 70%-30%, and 60%-40% samples, to experiment with decision tree with different ratios of train–test data.
- Also, to evaluate module performance for overfitting and generalization we have used:
    - Tree Depth (max_depth): [e.g., 10, 15, 20, None]
    - Minimum Samples per Split (min_samples_split): [e.g., 2, 5, 10]
    - Minimum Samples per Leaf (min_samples_leaf): [e.g., 1, 5, 10]

```
[92] regressor = DecisionTreeRegressor(max_depth=5, random_state=42)
     regressor.fit(X_train, y_train)# Train
     y_pred = regressor.predict(X_test)#predictions

     # Evaluate the model
     mae = mean_absolute_error(y_test, y_pred)
     mse = mean_squared_error(y_test, y_pred)
     r2 = r2_score(y_test, y_pred)

     print(f"Mean Absolute Error: {mae}")
     print(f"Mean Squared Error: {mse}")
     print(f"R^2 Score: {r2}")

     Mean Absolute Error: 0.5484847119325812
     Mean Squared Error: 0.5667530154590109
     R^2 Score: 0.4376812604702127
```



- With a high importance score near 0.8, the feature importance chart indicates that genre has the greatest influence on song popularity prediction.
- we have created a set of data as 90%-10%, 80%-20%, 70%-30%, and 60%-40% samples, to experiment decision tree with different ratios of train–test data.

- Also, to evaluate module performance for overfitting and generalization we have used:
    - Tree Depth (max_depth): [e.g., 10, 15, 20, None]
    - Minimum Samples per Split (min_samples_split): [e.g., 2, 5, 10]
    - Minimum Samples per Leaf (min_samples_leaf): [e.g., 1, 5, 10]

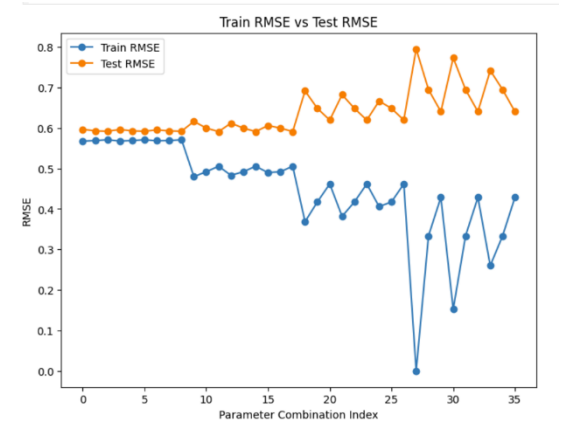
Experimental Results:

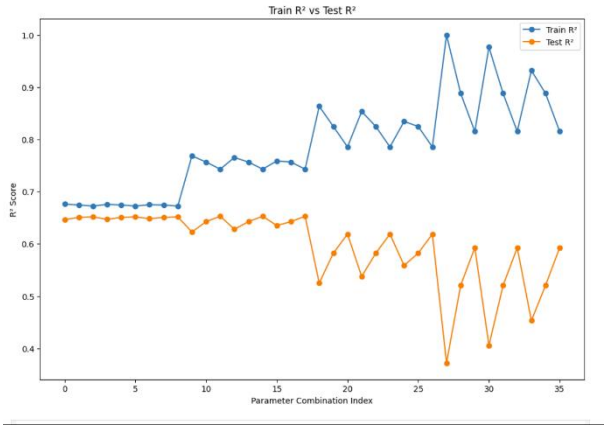
Multiple analyses has been performed to explore important factors in the decision tree:

- We have split the train and test data into multiple sizes, and where we have come up with the smaller test sizes (like 90%-10%) gave better results on the training data but struggled to generalize well to unseen data.
- The large size (60%-40%) of terrain-test data reduced training performance but did a much better job in generalizing to new data.
- Increasing max_depth enhanced the model's performance on the training set, as shown in the results of higher Train R2 and lower Train RMSE, which results in overfitting. We have defined from the results the difference between Train R2 and Test R2 and a larger Test RMSE, which suggests poor generalization to unknown data. As a result, we can see in Max_depth the train R2-1.000 but Test R2 = 0.372
- The parameters min_samples_split and min_samples_leaf were essential in balancing the model and streamlining the tree structure. By avoiding excessively complicated splits and smaller leaves. However, the larger values for min_samples_split and min_samples_leaf increased generalization and decreased overfitting. For example, the difference between Train RMSE (0.506) and Test RMSE (0.591) at max_depth = 15, min_samples_split = 5, and min_samples_leaf = 10.

Important Features:

The most powerful feature for predicting popularity was identified in predicting the genre, acousticness, and loudness based on the results we defined from our models in predicting. This will define the listener preferences, which help marketers and music producers create content that will be based on audience preference.

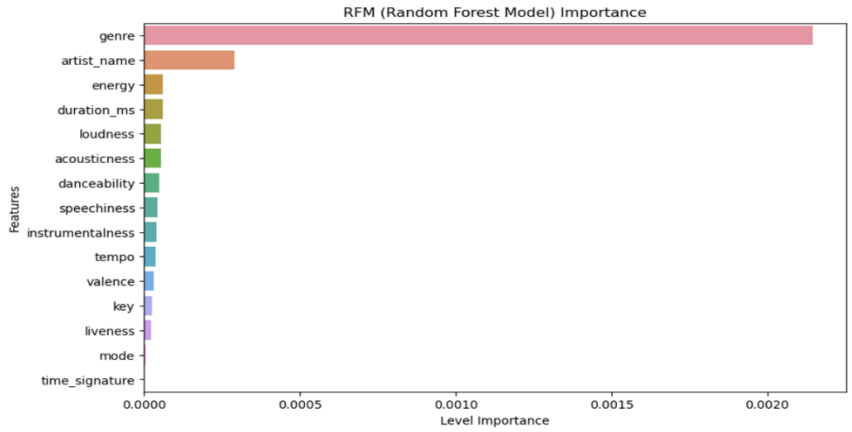




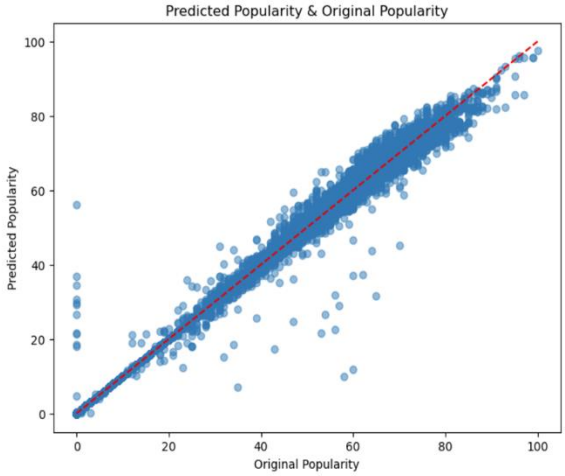
4.1.3 Random Forest

The main reason the Random Forest Regression is better for prediction is the fact that it is robust and handles both the classification and regression tasks, and through aggregation from several decision trees, it can reduce the risk of overfitting.

- **Data loading, preprocessing and training the model**: The dataset was loaded from the .csv file that contained all of the Spotify features. “Popularity” being the “target” variable was separated from the features. Using target encoding to clean, the values that were categorical were then changed to numerical. Afterwards the data was split into 80% training and the other 20% for testing. Using the Random Forest to train and then be evaluated through MSE (Mean Square Error) and R-Square
- Used the GridSearchCV to get the correct hyperparameters: number of trees, minimum samples, maximum depths, etc.
- After the training to understand the features extracted a lot better, we have made sure to conduct visualization methods, a bar plot to showcase the importance of each feature and a scatter plot to showcase the performance of the model.




```
Mean Squared Error: 0.9738963271501905
R-squared: 0.9970499210481792
```



	Metric	Value
1	Mean Squared Error (MSE)	0.9738963271501905
2	Root Mean Squared Error (RMSE)	0.9868618581899852
3	R-squared (R^2)	0.9970499210481792

Challenges:

While working on the dataset, some challenges faced were coming across the categorical features of the dataset (song names, genres, etc.) as they cannot be used directly by a machine-learning algorithm, therefore applying the “Target Encoding” method helped convert the data and make it usable for the Random Forest Regression. Since using the Random Forest model, if not looked after, it has a tendency of using features that are not needed which led to overfitting. To fix this, “feature importances” was added to carefully understand the contribution of each feature to the prediction of the model.

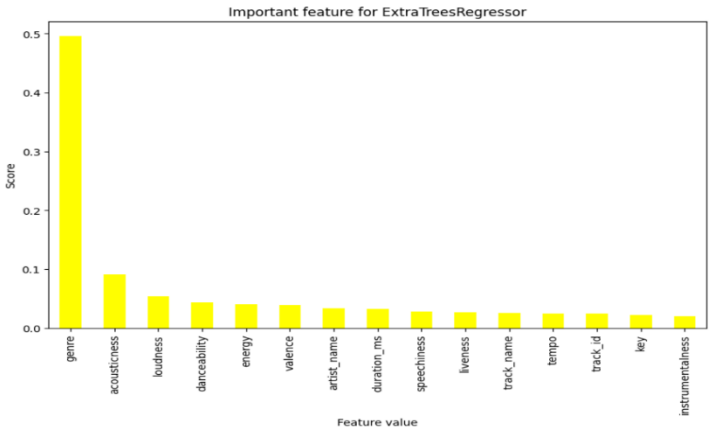
The challenges faced while working with the Spotify dataset using the flexible “Random Forest” model addressed issues regarding overfitting, encoding and more, and it was important to overcome these issues to ensure it was effective while conducting predictions. Some key insights observed were how the model captured good results while calculating the MSE (Mean Square Error) and R-Square in song popularity, while some analysis in features showcased how energy, loudness, and danceability were the highest, giving more insight on its popularity features. Finally, including a graph and a scatter plot provided with more visual understanding of the difference between true and predicted popularity. To conclude, the

Random Forest Regression model proved its effectiveness, providing us with good insights into the Spotify dataset.

4.1.4 ExtraTree Regressor

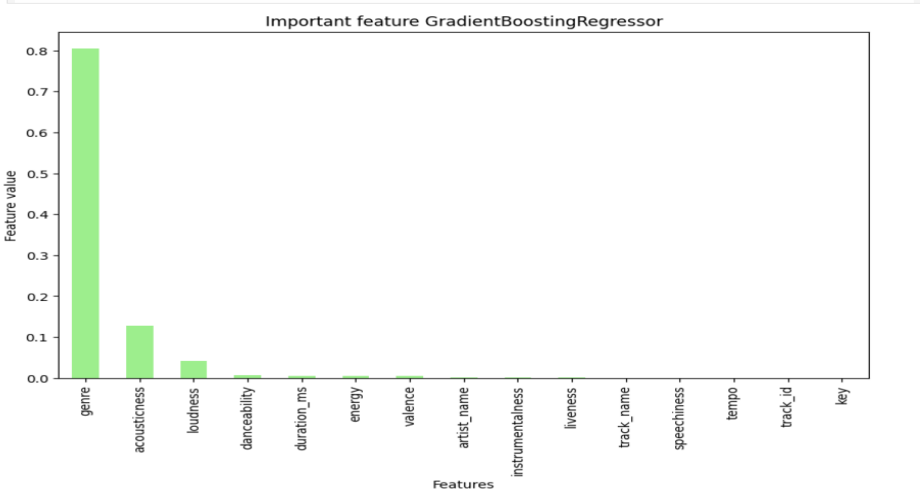
Genre is the most influential feature, with an importance score of about 0.5. Acousticness, loudness, and danceability follow secondary factors, indicating their roles in predicting song popularity, but to a smaller extent.

```
Mean Absolute Error: 0.4263120201303713
Mean Squared Error: 0.31057918489251823
R^2 Score: 0.6916343135908426
RMSE: 0.5652746706775484
```



4.1.5 Gradient Boosting Regressor

```
Mean Absolute Error: 0.45031961736584397
Mean Squared Error: 0.3480128568426264
R^2 Score: 0.6544674314969732
RMSE: 0.5652746706775484
```



In this module, the most important feature is showing genre, with a score of 0.8. Followed by Acousticness and loudness, which are secondary factors. These features Represent the importance of predicting the song's popularity.

4.2 Clustering

Working on clustering Spotify tracks audio features (acousticness , danceability, energy, instrumentalness , liveness ,loudness ,speechiness ,tempo ,key ,Mode, valence) to identify patterns songs, using advanced clustering techniques such as K-Means and DBSCAN to help generate playlists

Data cleaning: Exploring the genre column and fixing the "children's music "genre, removing duplicate tracks while maintaining the highest popularity value for these tracks, remove null values.

Data preprocessing: normalize numerical features and encode categorical features such as key, mode

Feature Engineering: Using dimensionality reduction with PCA for 2d , 3d & compare the results with original features.

Models:

1- Kmeans Clustering: both with PCA & original dimensions
   - Elbow method was used to determine the optimal number of clusters
   - Silhouette score was used to decide the best number of clusters from (3-5)
   - Davies-Bouldin Index & Calinski-Harabasz Index metrics were used to evaluate between PCA and original dimensions clustering to reduce processing time taken by silhouette score as the data is large.

2- DBscan Clustering: "Density-based spatial clustering of applications with noise " Using KNN with 4 neighbours to determine value of epsilon parameter& min point is set to 2*dim (working on the original data because it works best with high dimensions and did not work on external experiment with PCA)

4.3 Classification (Neural Networks)

The experiment was to classify album cover images by TensorFlow. The objective was to optimize the machine learning model with iterative improvement in design and training strategies. The project moved from a simple architecture to leverage pre-trained models and parameter tweaking that resulted in significant strides in accuracy.

Experiment Steps:

1. Initial Try: Simple CNN Without Pre-train Support
    a. A simple CNN was implemented from scratch.
    b. The basic architecture consisted of convolutional, pooling, and dense layers.
    c. Results after training:
        Accuracy: Only 16.45% was achieved, leaving much room for improvement.
2. MobileNetV2 Usage
    d. Adopted the pre-trained version of MobileNetV2 as the feature extractor.
    e. Data augmentation was used to increase variety in the dataset, including rotation, flipping, and zooming.
    f. An increase in the number of epochs allowed the model to learn better.
    g. Results after Training:
        Accuracy Improved to 24.65%, demonstrating the benefit of pre-trained networks and augmentation.
2. Applying Early Stopping and Preventing Overfitting
    a. Early stopping was introduced to halt training if validation performance ceased improving, reducing overfitting.
    b. The number of epochs was further adjusted in order to maximize learning.
    c. Training Results:
        Accuracy: Improved marginally to 24.97%, which showed a plateau in performance with current configuration.
3. Reducing Genre Categories
    d. To simplify the task and better focus the model, the genre list was reduced to a random subset of five classes.
    e. Training Results:
        Accuracy: Jumped to 47.17%, highlighting the impact of reducing the classification complexity.
4. Final Task: Classical vs. Non-Classical Classification
    f. The experiment was restructured into a binary classification task: Classical versus Non-Classical album covers.
    g. The dataset was balanced between these two classes.
    h. Training and Testing Results:
        Accuracy of Test: Well, 95.22%, which showed just how strong this model was for simple, binary tasks.

5. Results

5.1 Regression:

Regression models were applied to predict song popularity, highlighting the following results:

- **Linear Regression:**
  MSE: 259.32; limited by simplicity in capturing data complexity.
- **Decision Tree:**
  Overfitted on training data (R^2: 1.000) with poor generalization (R^2: 0.372).
  Key features: genre, acousticness, and loudness.
- **Random Forest:**
  Best performance with lower MSE and robust generalization.
  Important predictors: energy, loudness, and danceability.
- **ExtraTrees Regressor:**
  Genre was the most influential feature (importance ~0.5), followed by acousticness, loudness, and danceability.
  MSE: 196.75, MAE: 4.26, R^2: 0.55.
- **Gradient Boosting Regressor:**
  Genre had the highest importance score (0.8), followed by acousticness and loudness. MSE: 184.38, MAE: 4.58, R^2: 0.56.

Summary: Ensemble methods like Random Forest and Gradient Boosting delivered the best results, with genre and audio features like loudness and acousticness consistently emerging as key predictors of song popularity.

5.2 Clustering:

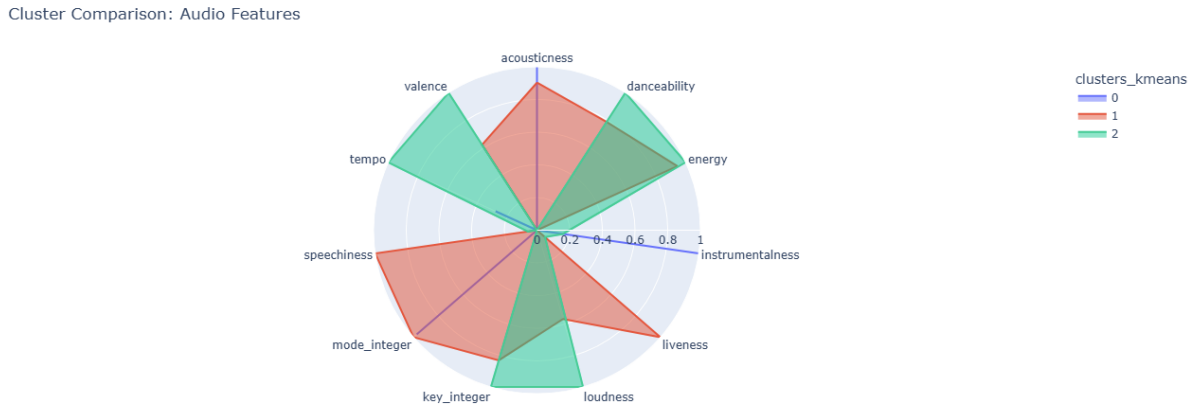
Optimal Clustering Method: KMeans performed better in separating the data into three distinct clusters, achieving a good silhouette score when applied to the 2D PCA-transformed, scaled audio features. In contrast, DBSCAN struggled to maintain cohesion within clusters but succeeded in identifying some noise points.

KMEANs silhouette score	DBscan silhouette score
0.597	0.1130

Clusters Analysis based on kmeans :

- **Cluster 0**: is defined by classical music, soundtracks, and opera, creating a reflective and calm atmosphere with mellow, contemplative tracks that lean towards relaxation. The high acousticness and lower energy and tempo enhance the peaceful nature of the music, while valence reflects mid-range emotions.
- **Cluster 1**: includes comedy tracks and less mainstream genres, resulting in a light and upbeat mood with moderate danceability and tempo. Valence is higher, reflecting positive emotions, and the tracks are of varied popularity and short in duration.

- **Cluster 2** :features modern, mainstream genres like electronic and hip-hop, offering an energetic and vibrant party-like mood with high danceability, fast tempos, and high energy and loudness. Valence is high, indicating celebratory emotions, and the tracks are vocal-heavy and under 5 minutes, suitable for commercial radio play and energetic playlists.



* Further visualizations for clustering are on GitHub

5.3 Classification:

Summary Results:

- No Pre-trained Support-Initial CNN: 16.45%
- MobileNetV2 with Data Augmentation: 24.65%
- Early Stopping with Increased Epochs: 24.97%
- Reduced Classes: 5 - Genre: 47.17%
- Classical vs. Non-Classical: 95.22%

6. Discussion

Comparing the results with some related work:

Benchmarks on the album cover genre classification part :

This was a challenging experiment because of a lack of sufficient data and the complication brought in by the number of genres. Starting without support from pre-trained models was tough as the custom CNN struggled to extract good features, resulting in accuracy as low as 16.45%.

The large genre list made this task quite complicated, since the differences between different visual features of album covers for some categories were really hard to learn by the model.

No doubt, similar studies across the internet highlights the same challenges :

- CSE 546 Final Paper Predicting Musical Genre from Album Cover Art:

  In this paper , researchers found that decreasing the dimensions of the pic would improve the accuracy (even with smaller dimensions than our study) . Also, the study showed that using a pretrained models would also be beneficial.

- CS230 - Genre Classification via Album Cover:

  In this paper, researchers used VGG16 for Transfer Learning, while we used MobileNetV2 . Depending on the data, But MobileNetV2 has been demonstrating better performance and accuracy. Ref (https://www.kaggle.com/code/arnavr10880/vgg16-vs-mobilenetv2-who-wins)

Comparing some results from a published paper in clustering:

Machine learning Based Clustering Using Spotify Audio Features: In comparison to this paper, which evaluated clustering methods (K-means, Fuzzy C-means, and PCM) on Spotify audio features and found K-means (n=6) with a silhouette score of 0.465 as the most effective, our approach also validated K-means as the optimal method but with fewer clusters (n=3) achieving a silhouette score of 0.597. While their study highlighted PCM's stability, our findings suggested DBSCAN struggled with noise handling. Both studies emphasize K-means' superior clustering efficiency for audio features, though the optimal cluster number varied based on dataset scope and features.

7. Conclusion

The project successfully demonstrated the applicability of machine learning in music analytics. The prediction of song popularity, clustering for playlist generation, and genre classification by album covers provided valuable insights. Advanced models used for ensemble regressors, K-Means clustering, and pre-trained neural networks were crucial for treading over the obvious challenges inherent in the data.

This report repeats the importance of strong preprocessing and feature engineering, along with the right selection of models, in enhancing prediction and clustering efficiency in music datasets.

8. References

CS230 - Genre Classification via Album Cover:
Reference (https://cs230.stanford.edu/projects_spring_2020/reports/38953572.pdf)

CSE 546 Final Paper Predicting Musical Genre from Album Cover Art:
Reference (https://rjbaraldi.github.io/assets/websitefiles/paper.pdf)

S. Shinde, S. Kulkarni, P. Kulkarni and A. Bhatt, "Machine learning Based Clustering Using Spotify Audio Features," 2023 1st DMIHER International Conference on Artificial Intelligence in Education and Industry 4.0 (IDICAIEI), Wardha, India, 2023 (https://ieeexplore.ieee.org/abstract/document/10406332)