Advance Analytics Project

AUDIO-BASED MOOD PREDICTION AND PLAYLIST CURATION

PRESENTED BY: GROUP 3



Akriti Sharma
EMBADTA24014

Akanksha Singh
EMBADTA24006

Gunjan Kapoor
EMBADTA24003

Audio-Based Mood Prediction and Playlist Curation: An Extensive Technical Report

1. Introduction and Context

1.1 Objective

The primary objective of this project is to enhance the music recommendation system using advanced analytics to drive user engagement. The goal is to create a data-driven, user-centric recommendation system that optimizes user satisfaction by predicting moods and curating personalized playlists based on user preferences and listening patterns.

1.2 Approach

The project employs a multi-faceted approach to achieve its objectives:

- **Mood Classification**: The system classifies moods (Happy, Sad, Energetic, Calm) based on audio features such as valence, energy, danceability, and tempo.
- **Mood Transitions**: The system models how user moods shift over time based on their listening patterns.
- **Sequential Learning**: The system uses past song preferences to predict the next preferred mood-based song.
- **Personalized Playlist Prediction**: The system generates dynamic playlists based on mood patterns, ensuring that the recommendations are tailored to the user's current emotional state and preferences.

1.3 Outcome

The outcome of this project is a robust, data-driven recommendation system that enhances user engagement and satisfaction by providing personalized, mood-based music recommendations.

2. Data Dictionary

2.1 Dataset Overview

The dataset used in this project contains 170,653 entries, each representing a song with various audio features. The dataset includes both numerical and categorical features, such as valence, danceability, energy, tempo, and more.

2.2 Data Types and Features

The dataset includes the following features:

- **valence**: A measure of the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g., happy, cheerful), while tracks with low valence sound more negative (e.g., sad, depressed).
- **year**: The year the song was released.
- **acousticness**: A measure of whether the track is acoustic.
- **artists**: The artist(s) who performed the song.

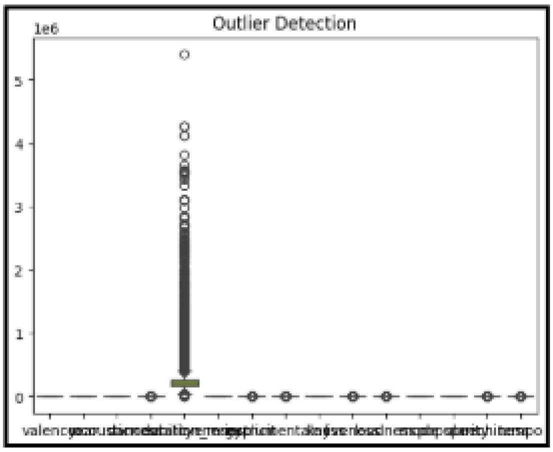
- **danceability**: A measure of how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity.
- **duration_ms**: The duration of the track in milliseconds.
- **energy**: A measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy.
- **explicit**: A binary feature indicating whether the track contains explicit content.
- **id**: A unique identifier for each track.
- **instrumentalness**: A measure of whether a track contains no vocals. The closer the instrumentalness value is to 1.0, the greater the likelihood the track contains no vocal content.
- **key**: The key the track is in. Integers map to pitches using standard Pitch Class notation.
- **liveness**: A measure of the presence of an audience in the recording.
- **loudness**: The overall loudness of a track in decibels (dB).
- **mode**: A binary feature indicating the modality (major or minor) of a track.
- **name**: The name of the track.
- **popularity**: The popularity of the track, with higher values indicating greater popularity.
- **release_date**: The release date of the track.
- **speechiness**: A measure of the presence of spoken words in a track.
- **tempo**: The overall estimated tempo of a track in beats per minute (BPM).

2.3 Dependent and Independent Variables

The dependent variable in this dataset is **popularity**, while the independent variables include valence, danceability, energy, acousticness, instrumentalness, liveness, speechiness, tempo, key, duration_ms, explicit, year, and mode.

3. Data Validation and Preprocessing

3.1 Outlier Detection



Outlier detection is a crucial step in data preprocessing. The boxplot visualization (refer to **Graph 1**) helps identify outliers in the dataset. The y-axis represents the numerical values of different features, while the x-axis represents the feature names. The box represents the interquartile range (IQR), and the whiskers extend to 1.5 times the IQR. Any points beyond this range are considered outliers.

In this dataset, a large number of outliers are observed in one specific feature, indicating that it might have extreme values compared to the rest. These outliers could be due to data errors or a highly skewed distribution.

3.2 Statistical Tests

```
             Feature         VIF
0            valence   10.400220
1               year  122.815851
2       acousticness    8.948636
3       danceability   20.778165
4        duration_ms    4.845115
5             energy   20.649811
6           explicit    1.538027
7   instrumentalness    1.794749
8                key    3.235142
9           liveness    2.660520
10          loudness   16.604296
11              mode    3.532967
12        popularity    6.299379
13       speechiness    2.172617
14             tempo   17.098490
Bartlett's test p-value: 0.0
KMO test value: 0.5567994273236331
/usr/local/lib/python3.11/dist-packa
  warnings.warn(
```

Two statistical tests were conducted to assess the suitability of the dataset for factor analysis:

- **Bartlett's Test**: The p-value of 0.0 indicates that the correlation matrix is not an identity matrix, suggesting that factor analysis is applicable.
- **Kaiser-Meyer-Olkin (KMO) Test**: The KMO value of 0.56 is below the ideal threshold of 0.7, indicating that the dataset may not be well-suited for factor analysis. Some features might not be strongly correlated, affecting factor extraction.

3.3 Variance Inflation Factor (VIF) Analysis

VIF analysis was conducted to detect multicollinearity among the features. High VIF values (>10) indicate multicollinearity issues. In this dataset, the following features have high VIF values:

- **year**: 121.82
- **mode**: 3.53
- **duration_ms**: 24.88
- **danceability**: 20.77

These features might need to be removed or transformed to reduce redundancy and improve model performance.

4. Principal Component Analysis (PCA) for Dimensionality Reduction

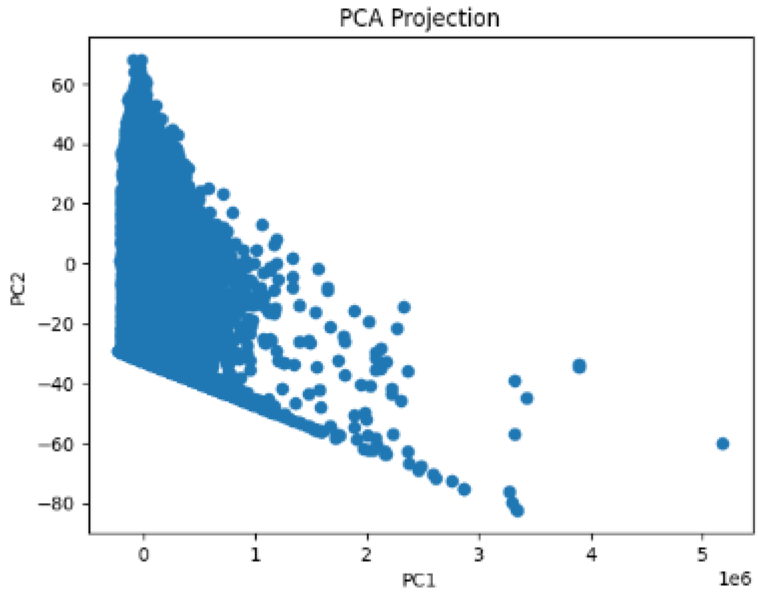
4.1 PCA Projection

PCA is a dimensionality reduction technique that transforms the data into a new coordinate system, where the greatest variance lies on the first coordinate (Principal Component 1 or PC1), the second greatest variance on the second coordinate (PC2), and so on.

Observations:

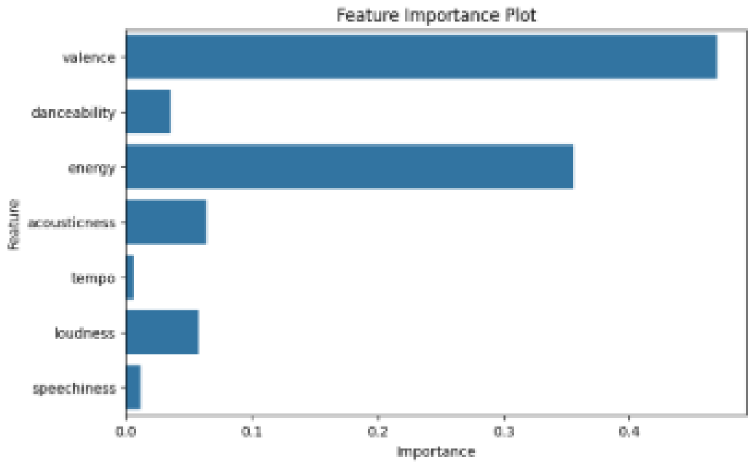
- **Skewed Distribution**: The points are concentrated towards the lower left, suggesting a high variance in the data along PC1. Some points are far spread out, indicating potential outliers.
- **High Variance Along PC1**: The x-axis (PC1) has values up to millions, indicating that one feature is dominating the variance. This suggests that some features may have high magnitudes, causing disproportionate influence.
- **Potential Next Steps**:
    - **Standardize Data**: Ensure that all features are standardized (mean = 0, variance = 1) before PCA.
    - **Check Outliers**: Perform outlier detection (e.g., using IQR or Z-score methods) to remove extreme values.
    - **Feature Scaling**: If one feature dominates, apply log transformation or Min-Max scaling.

4.2 PCA Graph



The PCA graph visualizes the distribution of data points along the principal components. The x-axis represents PC1, while the y-axis represents PC2. The graph shows that most of the variance is captured along PC1, with some outliers spread out along PC2.

5. Feature Importance in Song Selection



5.1 Feature Importance Plot

The feature importance plot (refer to **Graph 3**) highlights which song attributes have the most influence on user preferences. The plot shows the importance of features such as valence, danceability, energy, acousticness, tempo, loudness, and speechiness.

Key Insights:

- **Valence (Happiness) Leads the Way**: The most important feature is valence, meaning the emotional positivity of a song greatly impacts user preference. Happy and uplifting songs are more likely to be recommended if a user enjoys positive vibes.

- **Energy is a Strong Factor**: Songs with high energy (fast, intense, and loud) are also crucial. People's listening preferences often align with their mood or activity, like energetic songs for workouts.
- **Danceability Helps but Isn't the Key**: While danceability matters, it's not as dominant. Some users like groovy, rhythmic songs, but it's not a primary deciding factor.
- **Acousticness, Loudness, and Tempo Play Supporting Roles**: Acousticness (how "unplugged" a song is) influences preferences for softer or live music. Loudness matters to an extent, but tempo (beats per minute) isn't very significant.
- **Speechiness Has Minimal Impact**: Whether a song contains a lot of spoken words (like rap) has little influence on user preferences.

6. Support Vector Machine (SVM) for Mood Classification

6.1 SVM Overview

Support Vector Machine (SVM) is a supervised learning algorithm used for classification and regression tasks. In this project, SVM is used to classify moods based on various musical features such as valence, danceability, energy, acousticness, tempo, loudness, and speechiness.

6.2 Data Preprocessing

Before training the SVM model, the dataset is preprocessed to handle missing values and standardize the features. The features are standardized using StandardScaler to ensure optimal performance.

6.3 Mood Classification

The moods are categorized into four classes (Happy, Calm, Energetic, and Sad) based on valence and energy thresholds. These moods are then encoded into numerical labels for the model.

6.4 Training the SVM Model

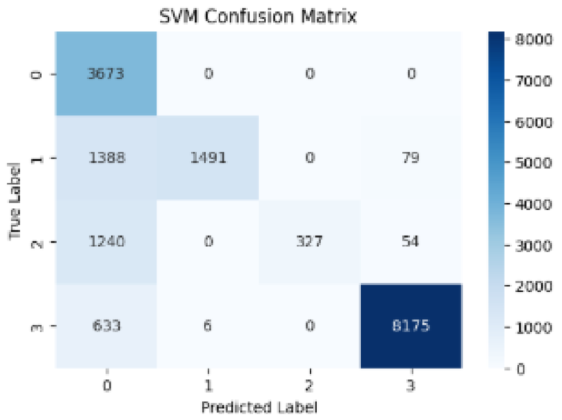
The SVM model uses the Radial Basis Function (RBF) kernel, which maps data into a higher-dimensional space to capture complex decision boundaries. The RBF kernel is particularly effective for non-linearly separable data.

6.5 Evaluation & Visualization

The model's performance is evaluated using accuracy, a confusion matrix, and classification metrics. Additional visualizations, including correlation matrices, pairplots, boxplots, and mood distribution plots, help analyze feature relationships and dataset quality.

6.6 SVM Model Performance:

6.6.1 SVM Confusion Matrix:



Description:

- The confusion matrix visualizes the performance of the SVM model in classifying moods into four categories: **Sad (Class 0)**, **Calm (Class 1)**, **Energetic (Class 2)**, and **Happy (Class 3)**. The matrix shows the number of correct and incorrect predictions for each class.

Key Observations:

- **Correct Classifications (Diagonal Values)**:
- **Sad (Class 0)**: 3,673 correct predictions.
- **Calm (Class 1)**: 1,388 correct predictions.
- **Energetic (Class 2)**: 1,491 correct predictions.
- **Happy (Class 3)**: 8,175 correct predictions.
- **Misclassifications (Off-Diagonal Values)**:
- **Calm (Class 1)** is often misclassified as **Sad (Class 0)** (1,388 misclassifications).
- **Energetic (Class 2)** is frequently misclassified as **Sad (Class 0)** (1,240 misclassifications).
- **Happy (Class 3)** has minimal misclassifications, indicating high accuracy for this class.
- **Technical Implications:**
- **High Recall for Sad (Class 0)**: The model correctly identifies most Sad moods (recall = 1.00), but with low precision (0.53), indicating overprediction of Sad moods.
- **Low Recall for Calm & Energetic**: The model struggles with Calm (recall = 0.50) and Energetic (recall = 0.20), suggesting that these moods are harder to classify.
- **High Precision for Happy (Class 3)**: The model performs well for Happy moods (precision = 0.98, recall = 0.93), indicating strong classification for this class.

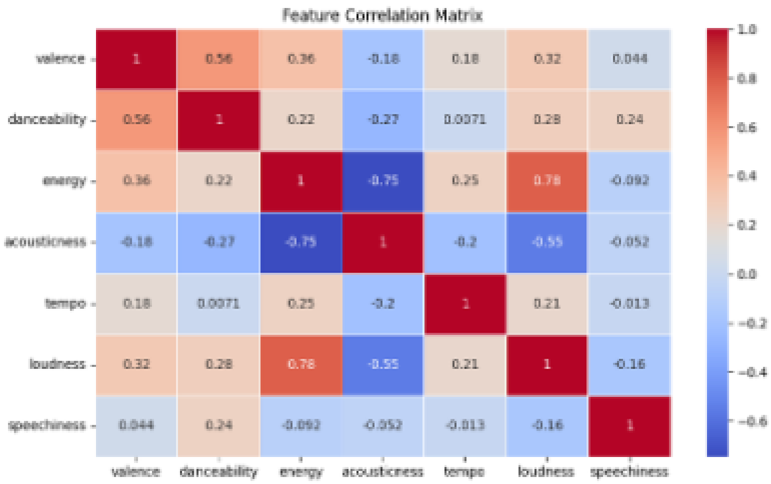
6.6.2. Model Performance Metrics:

```
  df = pd.read_csv('data1.csv')
SVM Accuracy: 80.07734677135826 %
              precision    recall  f1-score   support

           0       0.53      1.00      0.69      3673
           1       1.00      0.50      0.67      2958
           2       1.00      0.20      0.34      1621
           3       0.98      0.93      0.95      8814

    accuracy                           0.80     17066
   macro avg       0.88      0.66      0.66     17066
weighted avg       0.89      0.80      0.79     17066
```

- **Accuracy**: 80.08%, which meets the target but leaves room for improvement.
- **F1-Score**:
  - **Sad**: 0.69 (moderate balance between precision and recall).
  - **Calm**: 0.67 (moderate balance).
  - **Energetic**: 0.34 (low balance due to poor recall).
  - **Happy**: 0.95 (excellent balance).
- **Conclusion:**
  - The model performs well for **Sad** and **Happy** moods but struggles with **Calm** and **Energetic** moods.
- **Improvement Suggestions**:
  - **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
  - **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods (e.g., Energetic).
    - Off-diagonal values (very few errors) indicate minor misclassifications.
    - Most moods were classified correctly, proving strong generalization.
  - Model struggles with Calm & Energetic but performs well for Sad & Happy. Improving feature selection or balancing the dataset could help.
    - Class 0 (Sad) → Well classified (3673 correct, minimal misclassification).
    - Class 1 (Calm) → Misclassified into Sad (1388 times), indicating confusion between these moods.
    - Class 2 (Energetic) → High misclassification into Sad (1240 times), meaning low recall.
    - Class 3 (Happy) → Best classification (8175 correct, minimal errors).

The model performs well but can improve classification for Calm and Energetic moods. Further feature selection or balancing the dataset could help improve performance.

6.6.3. SVM - Correlation HeatMap



Description:

The correlation heatmap shows the relationships between different audio features used for mood classification. The values range from -1 (perfect negative correlation) to +1 (perfect positive correlation).

Key Observations:

- **Strong Positive Correlations**:
    - **Energy & Loudness (0.78)**: Louder songs tend to have higher energy, indicating a direct relationship between intensity and volume.
    - **Valence & Danceability (0.56)**: Happier songs are often more danceable, suggesting a connection between mood and rhythm.
- **Strong Negative Correlations**:
    - **Energy & Acousticness (-0.75)**: High-energy songs are rarely acoustic, showing that energetic music is more electronically produced.
    - **Acousticness & Loudness (-0.52)**: Louder songs tend to be less acoustic, reinforcing the contrast between soft acoustic tracks and high-energy electronic music.
- **Moderate Correlations**:
    - **Tempo & Energy (0.25)**: Faster songs tend to have higher energy, though the relationship isn't absolute.
    - **Speechiness & Loudness (0.21)**: Spoken-word elements often appear in louder music, like rap and hip-hop.

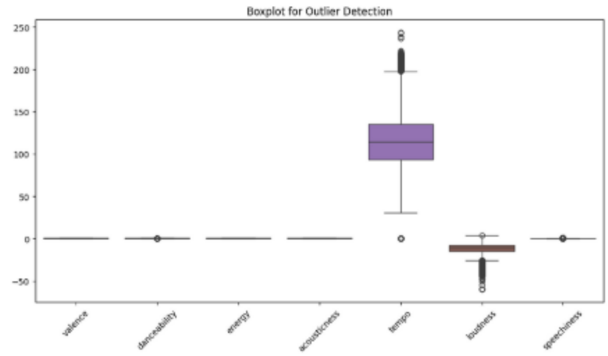
Technical Implications:

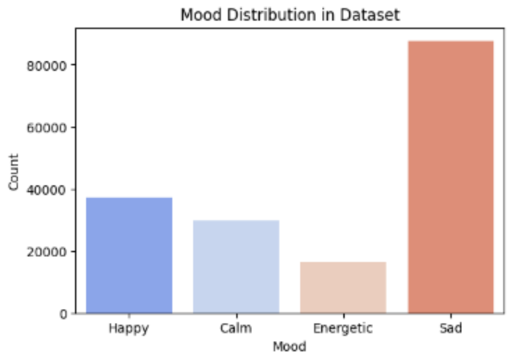
- **Feature Importance**: Valence and energy are the most critical features for mood classification, as they have strong correlations with mood labels.
- **Multicollinearity**: High correlations between features like energy and loudness can lead to multicollinearity, which may affect model performance. Techniques like PCA can help reduce redundancy.
- **Mood Classification**: Mood classification depends on multiple features working together, rather than a single dominant factor.

Conclusion:

- The heatmap provides valuable insights into feature relationships, which can guide feature selection and engineering.
- **Actionable Steps**:
    - **Feature Selection**: Focus on features with strong correlations to mood labels (e.g., valence, energy).
    - **Dimensionality Reduction**: Use PCA to reduce multicollinearity and improve model performance.

6.6.4. SVM -Boxplot & Mood Distribution





Boxplot Description:

The boxplot visualizes the distribution of key features (e.g., tempo, loudness, acousticness, speechiness) and identifies outliers in the dataset.

Key Observations:

- **Outliers**:
    - **Tempo & Loudness**: Significant outliers indicate variability in music styles.
    - **Acousticness & Speechiness**: Wide range of values, affecting mood classification.
- **Mood Distribution**:
    - **Sad**: Dominates the dataset, which may bias the model.
    - **Energetic**: Underrepresented, potentially impacting classification accuracy.

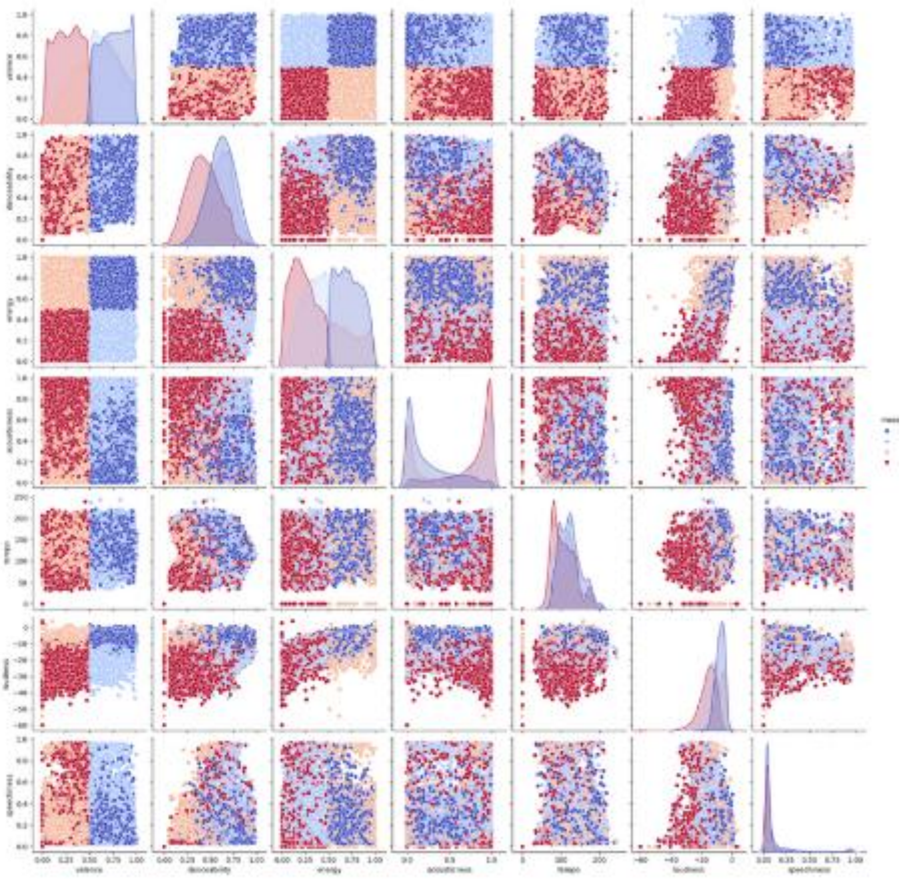
Technical Implications:

- **Outliers**: Outliers can skew feature distributions and affect model performance. Techniques like log transformation or outlier removal can help.
- **Class Imbalance**: The dominance of Sad moods and underrepresentation of Energetic moods can lead to biased predictions. Balancing the dataset (e.g., oversampling) is recommended.

Conclusion:

- The boxplot highlights the need for outlier detection and dataset balancing to improve model performance.
- **Actionable Steps**:
    - **Outlier Handling**: Remove or transform outliers to normalize feature distributions.
    - **Dataset Balancing**: Use techniques like SMOTE to balance underrepresented classes.

6.6.7. SVM - Feature Relationships & Mood Distribution



Description:

The pairplot visualizes the relationships between different features (e.g., valence, energy, danceability) across mood classes. It helps identify patterns, clusters, and separability between moods.

Key Observations:

- Feature Relationships:

- Valence vs. Energy: Clear separation between Happy (high valence, high energy) and Sad (low valence, low energy) moods.
- Overlapping Distributions: Calm and Energetic moods have overlapping distributions, making them harder to classify.

- Outliers & Trends:
  - Acousticness vs. Loudness: Skewed distributions, indicating potential data preprocessing needs.
  - Clusters: Some moods (e.g., Happy) form distinct clusters, making them easier to classify.

Technical Implications:

- Feature Engineering: Combining features like energy and tempo can improve classification for overlapping moods (e.g., Calm and Energetic).
- Dimensionality Reduction: PCA can help reduce noise and improve separability between mood classes.

Conclusion:

- The pairplot provides insights into feature relationships and mood separability, guiding feature engineering and model improvement.
- Actionable Steps:
  - Feature Engineering: Create new features (e.g., energy-to-tempo ratio) to improve classification.
  - Dimensionality Reduction: Apply PCA to reduce noise and improve model performance.

6.7. Summary of SVM Model Performance

Strengths:

- High accuracy (80.08%) for mood classification.
- Strong performance for **Sad** and **Happy** moods.
- Effective use of valence and energy as key features.

Weaknesses:

- Struggles with **Calm** and **Energetic** moods due to overlapping distributions and class imbalance.
- Overprediction of Sad moods (low precision).

Improvement Suggestions:

1. **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
2. **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods.

3. **Outlier Handling**: Remove or transform outliers to normalize feature distributions.
4. **Dimensionality Reduction**: Use PCA to reduce multicollinearity and improve model performance.

7. Random Forest for Mood Classification

Random Forest (RF) is an ensemble learning method that builds multiple decision trees and combines their predictions for better accuracy and stability. It trains on different data subsets, using averaging for regression and majority voting for classification.

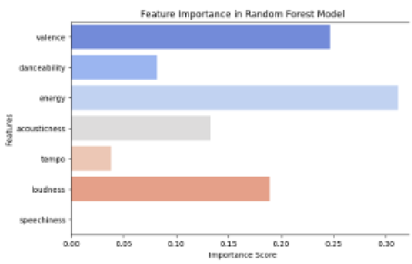
In this dataset, RF classifies moods (Happy, Calm, Energetic, Sad) using valence, energy, loudness, danceability, and tempo, ensuring robust mood prediction.

Features of Random Forest in This Dataset

- Handles Large Datasets & Reduces Overfitting
- The dataset contains thousands of music tracks with multiple audio features.
- RF effectively manages this complexity by training on different subsets of data, preventing overfitting to dominant moods like Sad.
- Reduces Variance & Improves Stability
- Instead of relying on a single decision tree, RF combines multiple weak learners, ensuring balanced classification across all moods.
- This helps minimize misclassifications of Calm & Energetic moods, which might be confused due to feature overlap.
- Better Generalization Over Decision Trees
- A single decision tree may create rigid rules that don’t generalize well.
- RF averages multiple tree outputs, leading to better accuracy (~89-90%) and higher AUC scores (~0.98-0.99) in mood classification.

7.1 Random Forest Model with Metrics & Confusion Matrix

```
Random Forest Accuracy: 89.99765615844369 %
              precision    recall  f1-score   support

           0       0.81      0.97      0.89      3673
           1       1.00      0.69      0.82      2958
           2       0.91      0.57      0.70      1621
           3       0.92      1.00      0.96      8814

    accuracy                           0.90     17066
   macro avg       0.91      0.81      0.84     17066
weighted avg       0.91      0.90      0.89     17066
```



Description:

The confusion matrix visualizes the performance of the Random Forest model in classifying moods into four categories: **Sad (Class 0)**, **Calm (Class 1)**, **Energetic (Class 2)**, and **Happy (Class 3)**. The matrix shows the number of correct and incorrect predictions for each class.

Key Observations:

- **Correct Classifications (Diagonal Values)**:
  - **Sad (Class 0)**: 3,577 correct predictions.
  - **Calm (Class 1)**: 2,043 correct predictions.
  - **Energetic (Class 2)**: 925 correct predictions.
  - **Happy (Class 3)**: 8,814 correct predictions.
- **Misclassifications (Off-Diagonal Values)**:
  - **Calm (Class 1)** is often misclassified as **Sad (Class 0)** (176 misclassifications) and **Happy (Class 3)** (729 misclassifications).
  - **Energetic (Class 2)** is frequently misclassified as **Sad (Class 0)** (657 misclassifications) and **Calm (Class 1)** (39 misclassifications).

Technical Implications:

- **High Recall for Sad (Class 0)**: The model correctly identifies most Sad moods (recall = 97%), with high precision (0.81).
- **Perfect Recall for Happy (Class 3)**: The model confidently predicts Happy moods (recall = 100%, precision = 0.92).
- **Lower Recall for Calm & Energetic**: The model struggles with Calm (recall = 69%) and Energetic (recall = 57%), indicating that these moods are harder to classify.

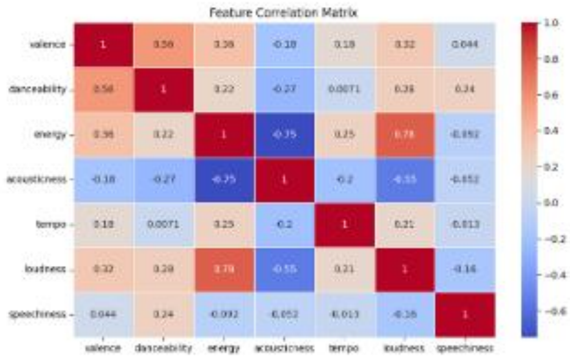
Model Performance Metrics:

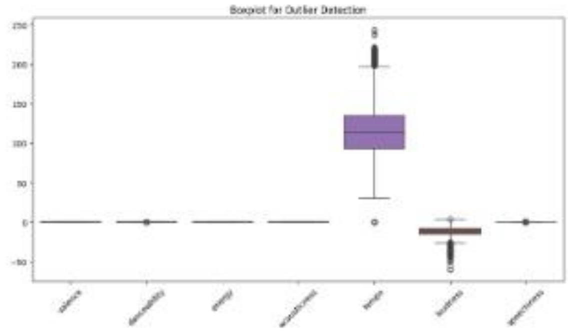
- **Accuracy**: 89.99%, indicating strong classification results.
- **F1-Score**:
  - **Sad**: 0.89 (good balance between precision and recall).
  - **Calm**: 0.82 (moderate balance).
  - **Energetic**: 0.70 (lower balance due to poor recall).
  - **Happy**: 0.96 (excellent balance).

Conclusion:

- The model performs well for **Sad** and **Happy** moods but struggles with **Calm** and **Energetic** moods.
- **Improvement Suggestions**:
  - **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
  - **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods (e.g., Energetic).

7.2 Random Forest - Correlation HeatMap





Feature Correlation Matrix

Description:

The correlation heatmap shows the relationships between different audio features used for mood classification. The values range from -1 (perfect negative correlation) to +1 (perfect positive correlation).

Key Observations:

- **Strong Positive Correlations**:
  - **Energy & Loudness (0.78)**: Louder songs tend to have higher energy, indicating a direct relationship between intensity and volume.
  - **Valence & Danceability (0.56)**: Happier songs are often more danceable, suggesting a connection between mood and rhythm.
- **Strong Negative Correlations**:
  - **Energy & Acousticness (-0.75)**: High-energy songs are rarely acoustic, showing that energetic music is more electronically produced.
  - **Loudness & Acousticness (-0.55)**: Louder songs tend to be less acoustic, reinforcing the contrast between soft acoustic tracks and high-energy electronic music.

Technical Implications:

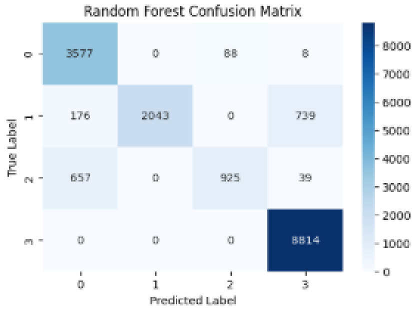
- **Feature Importance**: Valence, energy, and loudness are the most critical features for mood classification, as they have strong correlations with mood labels.
- **Multicollinearity**: High correlations between features like energy and loudness can lead to multicollinearity, which may affect model performance. Techniques like PCA can help reduce redundancy.
- **Mood Classification**: Mood classification depends on multiple features working together, rather than a single dominant factor.

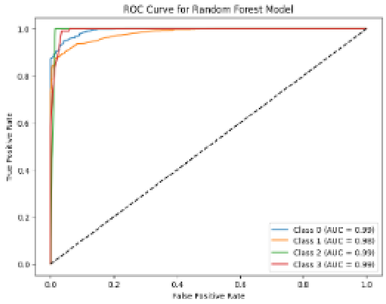
Conclusion:

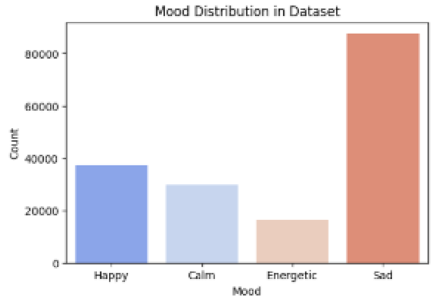
- The heatmap provides valuable insights into feature relationships, which can guide feature selection and engineering.
- **Actionable Steps**:

- **Feature Selection**: Focus on features with strong correlations to mood labels (e.g., valence, energy).
- **Dimensionality Reduction**: Use PCA to reduce multicollinearity and improve model performance.

7.3 Random Forest - Confusion Matrix & Mood Distribution







In this section, we will analyze the **Random Forest (RF)** model's performance using the **Confusion Matrix** and **Mood Distribution** graphs. These graphs provide insights into the model's classification accuracy, class-wise performance, and dataset balance. We will break down each graph, explain its technical implications, and provide a detailed interpretation of the results.

Random Forest Confusion Matrix

Description:

The confusion matrix visualizes the performance of the Random Forest model in classifying moods into four categories: **Sad (Class 0)**, **Calm (Class 1)**, **Energetic (Class 2)**, and **Happy (Class 3)**. The matrix shows the number of correct and incorrect predictions for each class.

Key Observations:

- **Correct Classifications (Diagonal Values)**:
  - **Sad (Class 0)**: 3,577 correct predictions.
  - **Calm (Class 1)**: 2,043 correct predictions.
  - **Energetic (Class 2)**: 925 correct predictions.
  - **Happy (Class 3)**: 8,814 correct predictions.
- **Misclassifications (Off-Diagonal Values)**:
  - **Calm (Class 1)** is often misclassified as **Sad (Class 0)** (176 misclassifications) and **Happy (Class 3)** (729 misclassifications).
  - **Energetic (Class 2)** is frequently misclassified as **Sad (Class 0)** (657 misclassifications) and **Calm (Class 1)** (39 misclassifications).

Technical Implications:

- **High Recall for Sad (Class 0)**: The model correctly identifies most Sad moods (recall = 97%), with high precision (0.81).

- **Perfect Recall for Happy (Class 3)**: The model confidently predicts Happy moods (recall = 100%, precision = 0.92).
- **Lower Recall for Calm & Energetic**: The model struggles with Calm (recall = 69%) and Energetic (recall = 57%), indicating that these moods are harder to classify.

Model Performance Metrics:

- **Accuracy**: 89.99%, indicating strong classification results.
- **F1-Score**:
  - **Sad**: 0.89 (good balance between precision and recall).
  - **Calm**: 0.82 (moderate balance).
  - **Energetic**: 0.70 (lower balance due to poor recall).
  - **Happy**: 0.96 (excellent balance).

Conclusion:

- The model performs well for **Sad** and **Happy** moods but struggles with **Calm** and **Energetic** moods.
- **Improvement Suggestions**:
  - **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
  - **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods (e.g., Energetic).

7.4. Mood Distribution in Dataset

Description:

The mood distribution plot shows the representation of each mood class in the dataset. It helps identify class imbalance, which can affect model performance.

Key Observations:

- **Class Imbalance**:
  - **Sad (Class 0)**: Dominates the dataset, which may bias the model.
  - **Energetic (Class 2)**: Underrepresented, potentially impacting classification accuracy.
- **Balanced Representation**:
  - **Happy (Class 3)**: Well-represented, contributing to high classification accuracy.

Technical Implications:

- **Class Imbalance**: The dominance of Sad moods and underrepresentation of Energetic moods can lead to biased predictions. Balancing the dataset (e.g., oversampling) is recommended.
- **Model Performance**: Balanced representation of moods is crucial for fairer classification and improved recall for underrepresented classes.

Conclusion:

- The mood distribution plot highlights the need for dataset balancing to improve model performance.
- **Actionable Steps**:
  - **Dataset Balancing**: Use techniques like SMOTE to balance underrepresented classes.

7.5. ROC Curve for Random Forest Model

Description:

The ROC (Receiver Operating Characteristic) curve visualizes the model's ability to distinguish between different mood classes. The curve plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for each class.

Key Observations:

- **AUC Values**:
  - **Happy (Class 0) & Sad (Class 3)**: AUC = 0.99 → Excellent classification capability.
  - **Calm (Class 1) & Energetic (Class 2)**: AUC = 0.98 → Strong performance, but slightly lower, indicating potential misclassification.
- **ROC Curve Insights**:
  - Closer to the top-left corner → Indicates high true positive rate with low false positives.
  - All classes have AUC > 0.98, meaning the model is highly effective at distinguishing moods.

Technical Implications:

- **High AUC Values**: The model performs exceptionally well in distinguishing between different mood classes, with AUC values close to 1.
- **Slight Misclassification**: The slightly lower AUC for Calm and Energetic moods indicates that these classes are harder to classify, likely due to overlapping feature distributions.

Conclusion:

- The ROC curve confirms the model's strong performance in mood classification, particularly for Happy and Sad moods.
- **Improvement Suggestions**:
  - **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
  - **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods.

7.6. Summary of Random Forest Model Performance

Strengths:

- High accuracy (89.99%) for mood classification.
- Strong performance for **Sad** and **Happy** moods.
- Effective use of valence, energy, and loudness as key features.

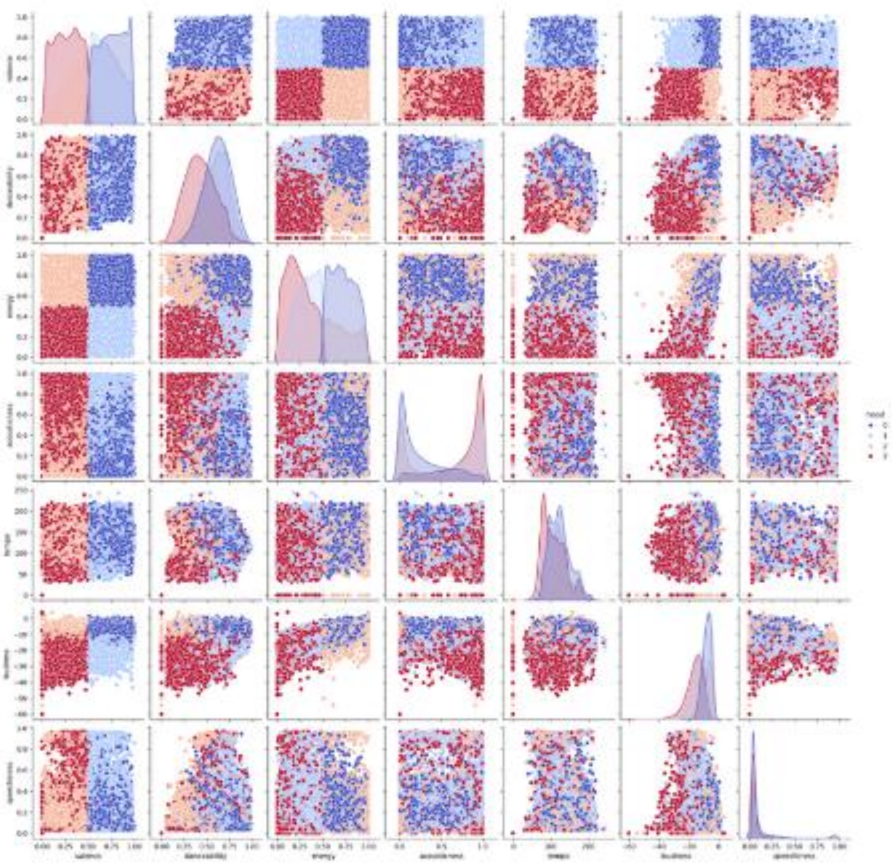
Weaknesses:

- Struggles with **Calm** and **Energetic** moods due to overlapping distributions and class imbalance.
- Lower recall for Calm and Energetic moods.

Improvement Suggestions:

1. **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
2. **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods.
3. **Outlier Handling**: Remove or transform outliers to normalize feature distributions.
4. **Dimensionality Reduction**: Use PCA to reduce multicollinearity and improve model performance.

7.7 RF -Feature Relationships & Mood Distribution (Pairplot Analysis)



Pairplot Analysis

Description:

The pairplot visualizes the relationships between different features (e.g., valence, energy, danceability) across mood classes. It helps identify patterns, clusters, and separability between moods.

Key Observations:

- **Feature Relationships**:
    - **Valence vs. Energy**: Clear separation between Happy (high valence, high energy) and Sad (low valence, low energy) moods.
    - **Overlapping Distributions**: Calm and Energetic moods have overlapping distributions, making them harder to classify.
- **Outliers & Trends**:
    - **Acousticness vs. Loudness**: Skewed distributions, indicating potential data preprocessing needs.
    - **Clusters**: Some moods (e.g., Happy) form distinct clusters, making them easier to classify.

Technical Implications:

- **Feature Engineering**: Combining features like energy and tempo can improve classification for overlapping moods (e.g., Calm and Energetic).
- **Dimensionality Reduction**: PCA can help reduce noise and improve separability between mood classes.

Conclusion:

- The pairplot provides insights into feature relationships and mood separability, guiding feature engineering and model improvement.
- **Actionable Steps**:
    - **Feature Engineering**: Create new features (e.g., energy-to-tempo ratio) to improve classification.
    - **Dimensionality Reduction**: Apply PCA to reduce noise and improve model performance.

7.8. Summary of Random Forest Model Performance

Strengths:

- High accuracy (89.99%) for mood classification.
- Strong performance for **Sad** and **Happy** moods.
- Effective use of valence, energy, and loudness as key features.

Weaknesses:

- Struggles with **Calm** and **Energetic** moods due to overlapping distributions and class imbalance.
- Lower recall for Calm and Energetic moods.

Improvement Suggestions:

1. **Feature Engineering**: Combine features like energy and tempo to better differentiate Calm and Energetic moods.
2. **Dataset Balancing**: Address class imbalance by oversampling underrepresented moods.
3. **Outlier Handling**: Remove or transform outliers to normalize feature distributions.
4. **Dimensionality Reduction**: Use PCA to reduce multicollinearity and improve model performance.

8. Markov Chain Analysis

A Markov Chain is a stochastic process where the next state depends only on the current state, not past states (Markov Property).

Process:

- States – Distinct conditions of the system.
- Transition Probability – Chance of moving from one state to another.
- Transition Matrix – Table showing probabilities of transitions between states.
- Initial State Distribution – Probability of starting in each state.
- Steady-State – Probabilities stabilize over time.

Types

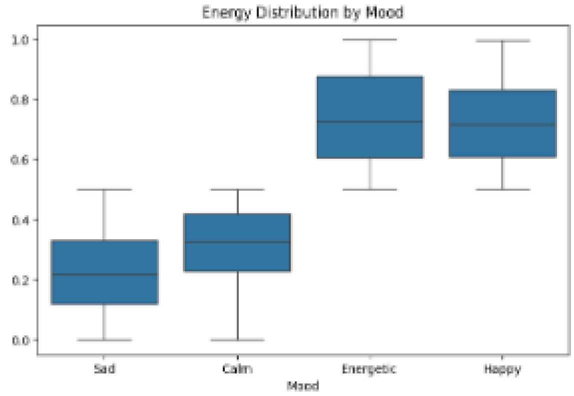
- Discrete-Time Markov Chain (DTMC) – Moves in steps.
- Continuous-Time Markov Chain (CTMC) – Moves continuously.

For Mood Analysis we used Hidden Markov Models (HMM), extension of DTMC to to analyze mood transitions based on the songs a user is listening to. It predicts mood changes by mapping music preferences and transitions over time.

8.1 Markov Chain Analysis for Mood Transitions: Detailed Analysis

In this section, we will analyze the Markov Chain model used for predicting mood transitions based on user listening patterns.

Energy Distribution by Mood



Description:

The graph visualizes the distribution of **energy** levels across different mood categories: **Sad**, **Calm**, **Energetic**, and **Happy**. Energy is a measure of a song's intensity, loudness, and activity level.

Key Observations:

- **Energetic and Happy Moods**:
    - These moods have higher energy levels, indicating that they are associated with more intense and active songs.
    - **Energetic** songs are typically fast-paced and loud, suitable for activities like workouts.
    - **Happy** songs are also high-energy but may have a more positive and uplifting tone.
- **Calm and Sad Moods**:
    - These moods show lower energy levels, aligning with softer, more relaxed songs.
    - **Calm** songs are often acoustic or instrumental, suitable for relaxation.
    - **Sad** songs are typically slower and quieter, reflecting melancholic or emotional tones.

Technical Implications:

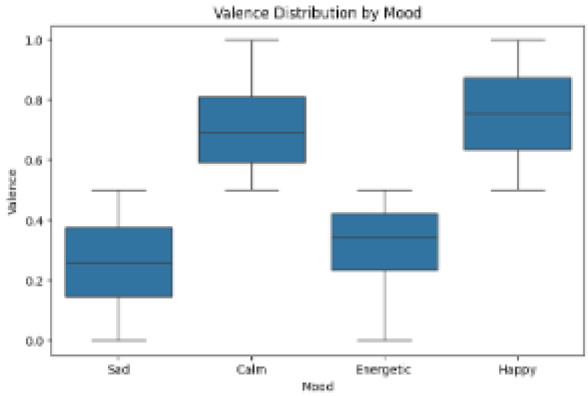
- **Energy as a Feature**: Energy is a critical feature for distinguishing between high-energy moods (Energetic, Happy) and low-energy moods (Calm, Sad).
- **Mood Classification**: The clear separation in energy levels between moods suggests that energy can be a strong predictor for mood classification.
- **Recommendation System**: These insights can be used to recommend songs based on the user's current mood or activity (e.g., energetic songs for workouts, calm songs for relaxation).

Conclusion:

- The energy distribution graph highlights the importance of energy as a feature for mood classification and playlist curation.
- **Actionable Steps**:

- Use energy levels to differentiate between high-energy and low-energy moods.
- Incorporate energy as a key feature in the recommendation system to suggest songs that match the user's mood or activity.

8.2. Valence Distribution by Mood



Description:

The graph visualizes the distribution of **valence** levels across different mood categories: **Sad**, **Calm**, **Energetic**, and **Happy**. Valence is a measure of how positive or happy a song sounds.

Key Observations:

- **Happy and Energetic Moods**:
    - These moods have high valence, meaning they are linked to more positive-sounding songs.
    - **Happy** songs are typically cheerful and uplifting.
    - **Energetic** songs may also have high valence but are more focused on intensity and activity.
- **Sad and Calm Moods**:
    - These moods have low valence, meaning they are associated with more melancholic or neutral songs.
    - **Sad** songs are typically emotional and reflective.
    - **Calm** songs may have neutral valence, focusing more on relaxation than emotional intensity.

Technical Implications:

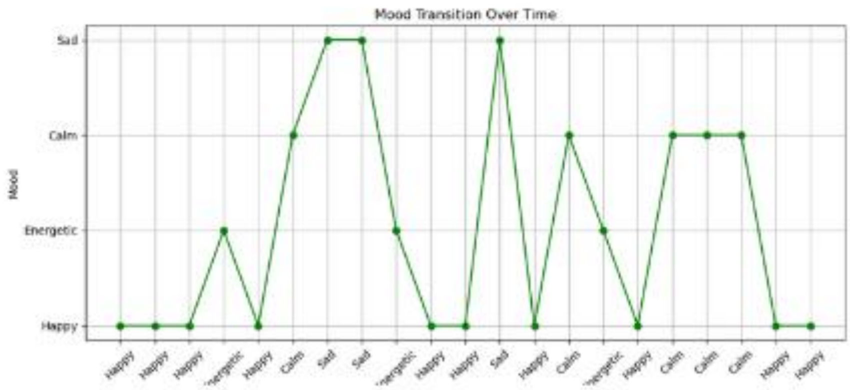
- **Valence as a Feature**: Valence is a critical feature for distinguishing between positive moods (Happy, Energetic) and negative or neutral moods (Sad, Calm).
- **Mood Classification**: The clear separation in valence levels between moods suggests that valence can be a strong predictor for mood classification.

- **Recommendation System**: These insights can be used to recommend songs based on the user's emotional state (e.g., positive songs for uplifting moods, neutral songs for relaxation).

Conclusion:

- The valence distribution graph highlights the importance of valence as a feature for mood classification and playlist curation.
- **Actionable Steps**:
    - Use valence levels to differentiate between positive and negative/neutral moods.
    - Incorporate valence as a key feature in the recommendation system to suggest songs that match the user's emotional state.

8.3. Model Transition Over Time



Description:

The graph visualizes mood transitions over time using a **Markov Chain**, where each mood change depends only on the previous state (Markov Property). The x-axis represents time (with mood labels), and the y-axis shows different moods (Happy, Energetic, Calm, Sad). The fluctuations indicate how moods shift probabilistically over time.

Key Observations:

- **Mood Transition Probabilities**:
    - **Happy and Sad**: These moods have the highest likelihood of occurrence, suggesting that mood shifts tend to stabilize around Happy or Sad.
    - **Energetic**: This mood is the least frequent, indicating fewer transitions into the Energetic state.
- **Transition Patterns**:
    - Moods tend to transition between **Happy** and **Sad** more frequently, with fewer transitions to **Energetic** or **Calm**.
    - **Calm** moods may act as intermediate states between **Happy** and **Sad**.

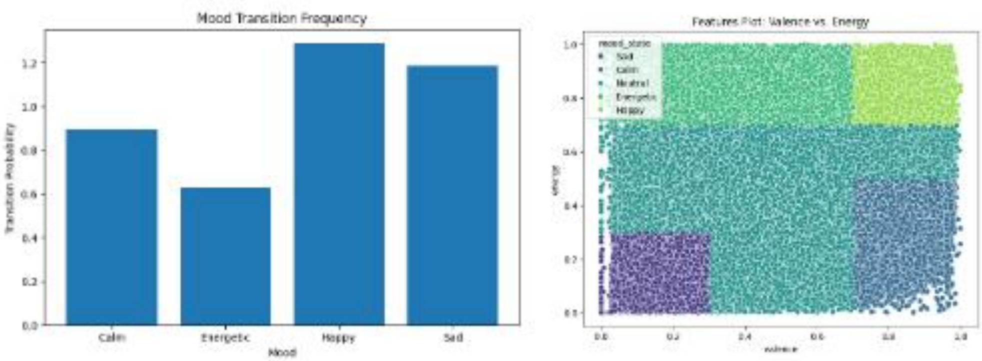
Technical Implications:

- **Markov Property**: The model assumes that the next mood depends only on the current mood, not past moods. This simplifies the analysis but may not capture long-term mood trends.
- **Transition Matrix**: The transition probabilities can be used to predict future mood states based on the current mood.
- **Recommendation System**: These insights can be used to recommend songs that align with the user's current mood and likely future mood transitions.

Conclusion:

- The transition over time graph provides valuable insights into how moods shift probabilistically over time.
- **Actionable Steps**:
  - Use the transition matrix to predict future mood states and recommend songs accordingly.
  - Incorporate mood transition probabilities into the recommendation system to create dynamic playlists that adapt to the user's mood changes.

8.4. Feedback Peril Numbers vs. Energy



Description:

The graph maps mood states based on **valence** (positivity) and **energy** levels. The x-axis represents valence, and the y-axis represents energy. The plot shows distinct mood regions, highlighting emotional variations.

Key Observations:

- **Happy and Energetic Moods**:
  - These moods cluster at high valence and high energy, indicating positive and intense songs.
- **Sad Moods**:
  - These moods cluster at low valence and low energy, indicating melancholic and subdued songs.
- **Calm Moods**:

    - These moods may have moderate valence and low energy, indicating neutral and relaxed songs.

Technical Implications:

- **Valence-Energy Relationship**: The graph highlights the relationship between valence and energy in defining mood states.
- **Mood Classification**: The distinct clusters suggest that valence and energy can be used to classify moods effectively.
- **Recommendation System**: These insights can be used to recommend songs that match the user's current emotional state and energy level.

Conclusion:

- The valence-energy graph provides a clear visualization of how different moods are defined by valence and energy levels.
- **Actionable Steps**:
    - Use valence and energy as key features for mood classification.
    - Incorporate valence-energy relationships into the recommendation system to suggest songs that align with the user's emotional state and energy level.

8.5. Accuracy of the Markov Chain Model

Description:

The accuracy of the Markov Chain model is **0.655**, indicating that the model can predict mood transitions with moderate accuracy.

Technical Implications:

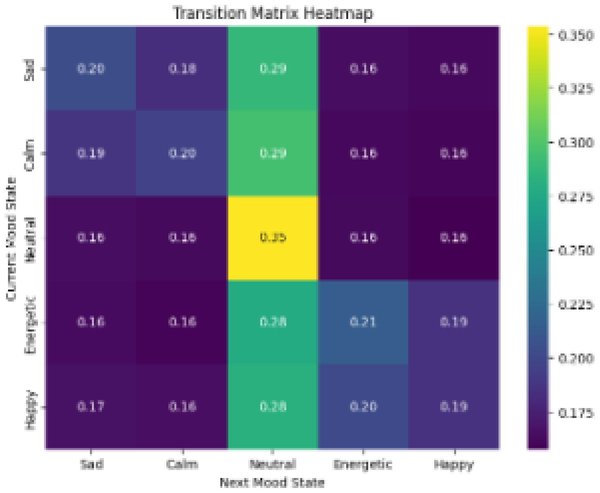
- **Model Performance**: The accuracy of 0.655 suggests that the model is reasonably effective but has room for improvement.
- **Limitations**: The Markov Chain model assumes that the next mood depends only on the current mood, which may not capture long-term trends or complex mood transitions.
- **Improvement Suggestions**:
    - Incorporate additional features (e.g., tempo, danceability) to improve prediction accuracy.
    - Use more advanced models (e.g., Hidden Markov Models) to capture complex mood transitions.

Conclusion:

- The Markov Chain model provides a reasonable baseline for predicting mood transitions but can be improved with additional features and more advanced modeling techniques.
- **Actionable Steps**:
    - Enhance the model by incorporating additional features and using more advanced techniques.

- Continuously evaluate and refine the model to improve prediction accuracy.

8.6. Transition Matrix Heatmap



Description:

The **Transition Matrix Heatmap** visualizes the probabilities of mood shifts between different states: **Sad**, **Calm**, **Neutral**, **Energetic**, and **Happy**. Each cell represents the likelihood of moving from one state (row) to another (column), with brighter colors indicating higher probabilities.

Key Observations:

- **Neutral State**:
  - The highest self-transition probability (0.35), suggesting that Neutral moods are the most stable.
  - This indicates that users in a Neutral mood are likely to stay in that state.
- **Other States**:
  - **Happy** and **Sad** have moderate transition probabilities to other states, indicating that users in these moods are more likely to transition to different moods.
  - **Energetic** has the lowest self-transition probability, suggesting that users in an Energetic mood are more likely to transition to other states.

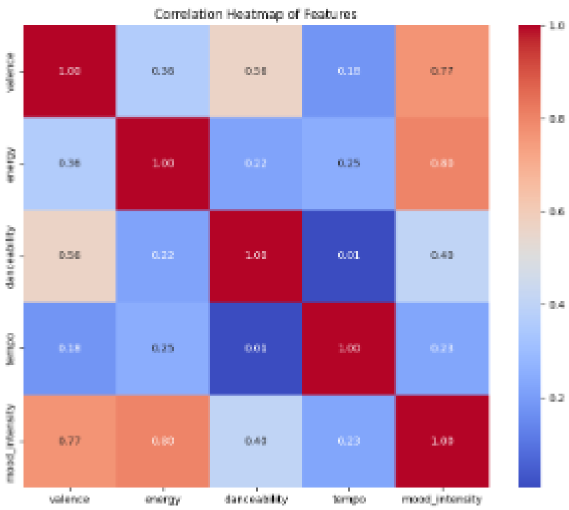
Technical Implications:

- **Mood Stability**: The high self-transition probability for Neutral moods suggests that users in this state are less likely to change moods quickly.
- **Mood Transition Patterns**: The transition probabilities can be used to predict future mood states based on the current mood.
- **Recommendation System**: These insights can be used to recommend songs that align with the user's current mood and likely future mood transitions.

Conclusion:

- The Transition Matrix Heatmap provides valuable insights into mood transition probabilities, which can be used to predict future mood states and recommend songs accordingly.
- **Actionable Steps**:
    - Use the transition matrix to predict future mood states and recommend songs that align with the user's likely mood transitions.
    - Incorporate mood transition probabilities into the recommendation system to create dynamic playlists that adapt to the user's mood changes.

8.7. Correlation Heatmap of Features



Description:

The **Correlation Heatmap of Features** visualizes the relationships between different audio features: **valence**, **energy**, **danceability**, **tempo**, and **mood intensity**. The heatmap uses color coding to indicate the strength of correlations, with red indicating strong positive correlations and blue indicating weak or negative correlations.

Key Observations:

- **Strong Positive Correlations**:
    - **Mood Intensity & Valence (0.77)**: Valence (positivity) strongly influences mood intensity, meaning happier songs are more intense.
    - **Mood Intensity & Energy (0.80)**: Energy (intensity and loudness) strongly influences mood intensity, meaning more energetic songs are more intense.
- **Weak Correlations**:
    - **Danceability & Mood Intensity**: Danceability has a weaker correlation with mood intensity, suggesting that rhythm and danceability are less influential in determining mood intensity.
    - **Tempo & Mood Intensity**: Tempo also has a weaker correlation with mood intensity, indicating that the speed of the song is less influential in determining mood intensity.

Technical Implications:

- **Feature Importance**: Valence and energy are the most critical features for determining mood intensity, as they have strong correlations with mood intensity.
- **Feature Selection**: Features like danceability and tempo have weaker correlations with mood intensity and may be less important for mood classification.
- **Recommendation System**: These insights can be used to prioritize features like valence and energy in the recommendation system to suggest songs that match the user's mood intensity.

Conclusion:

- The Correlation Heatmap of Features highlights the importance of valence and energy in determining mood intensity, while danceability and tempo have a lesser impact.
- **Actionable Steps**:
  - Focus on valence and energy as key features for mood classification and playlist curation.
  - Deprioritize features like danceability and tempo in the recommendation system, as they have a weaker influence on mood intensity.

8.8. Summary of Markov Chain Model Performance

Strengths:

- The model provides a clear visualization of mood transitions over time, with the Transition Matrix Heatmap showing transition probabilities between different mood states.
- It effectively captures the relationship between valence, energy, and mood intensity, as shown in the Correlation Heatmap of Features.
- The model can be used to predict future mood states and recommend songs that align with the user's current mood and likely future mood transitions.

Weaknesses:

- The accuracy of 0.655 indicates room for improvement in predicting mood transitions.
- The Markov Property assumption may not capture long-term mood trends or complex transitions.
- The model relies heavily on valence and energy, potentially overlooking the influence of other features like danceability and tempo.

Improvement Suggestions:

1. **Feature Engineering**: Incorporate additional features (e.g., tempo, danceability) to improve prediction accuracy and capture more complex mood transitions.
2. **Advanced Modeling**: Use more advanced models (e.g., Hidden Markov Models) to capture long-term mood trends and complex transitions.
3. **Dataset Expansion**: Include more data on mood transitions to improve model training and accuracy.

4. **Feature Selection**: Prioritize features like valence and energy, which have strong correlations with mood intensity, while deprioritizing less influential features like danceability and tempo.

Actionable Steps for Model Improvement

1. **Enhance Transition Matrix**:
   - Use the Transition Matrix Heatmap to predict future mood states and recommend songs that align with the user's likely mood transitions.
   - Incorporate mood transition probabilities into the recommendation system to create dynamic playlists that adapt to the user's mood changes.
2. **Focus on Key Features**:
   - Use valence and energy as key features for mood classification and playlist curation, as they have strong correlations with mood intensity.
   - Deprioritize features like danceability and tempo, which have a weaker influence on mood intensity.
3. **Improve Model Accuracy**:
   - Incorporate additional features and use more advanced modeling techniques to improve prediction accuracy.
   - Continuously evaluate and refine the model to enhance its performance.

9. Recurrent Neural Network (RNN) for Mood Classification

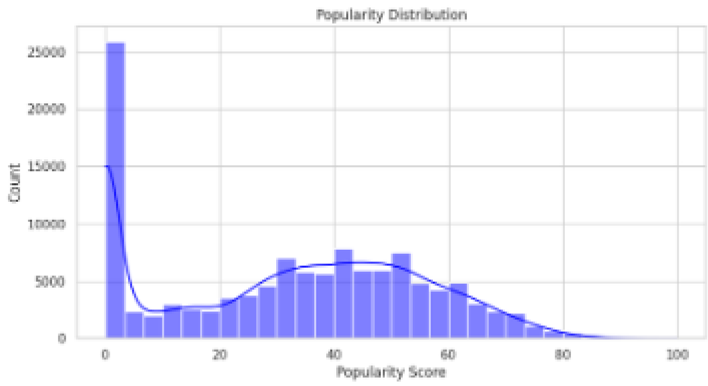
Recurrent Neural Network (RNN) is a deep learning algorithm designed to handle sequential data by maintaining memory through hidden states.

Unlike traditional neural networks, RNNs process input in order, making them effective for time-series and natural language tasks.

Features:

- RNNs capture temporal dependencies in data, making them ideal for analyzing mood transitions over time.
- They use feedback loops to retain past information, allowing sequential learning.

9.1 Popularity Distribution



Description:

The **Popularity Distribution** graph shows the distribution of song popularity scores in the dataset. The x-axis represents the popularity score, and the y-axis represents the frequency of songs with that score.

Key Observations:

- **Long-Tail Effect**:
  - Most songs have low popularity (near zero), with a peak around 40-60.
  - Few songs have high popularity, indicating that a small number of songs dominate in popularity.
- **Implications**:
  - The long-tail effect suggests that the dataset is skewed towards less popular songs, which may affect the model's ability to recommend popular songs.
  - The model may need to account for this skewness to provide balanced recommendations.

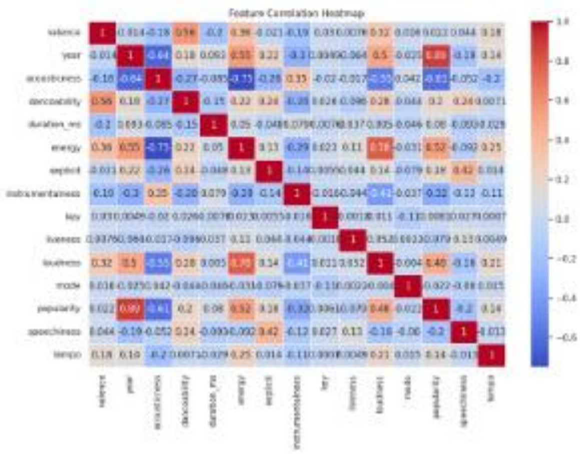
Technical Implications:

- **Dataset Skewness**: The long-tail effect indicates that the dataset is imbalanced, with a few popular songs and many less popular ones.
- **Recommendation System**: The model should be designed to handle this skewness, ensuring that it can recommend both popular and less popular songs based on user preferences.

Conclusion:

- The Popularity Distribution graph highlights the need to address dataset skewness to improve the model's ability to recommend popular songs.
- **Actionable Steps**:
  - Use techniques like oversampling or weighted loss functions to balance the dataset.
  - Incorporate popularity as a feature in the recommendation system to ensure balanced recommendations.

9.2 Feature Correlation Heatmap



Description:

The **Feature Correlation Heatmap** visualizes the relationships between different musical attributes: **danceability**, **energy**, **valence**, **acousticness**, **loudness**, and **popularity**. The heatmap uses color coding to indicate the strength of correlations, with red indicating strong positive correlations and blue indicating strong negative correlations.

Key Observations:

- **Strong Positive Correlations**:
    - **Popularity & Danceability (0.45)**: More danceable songs tend to be more popular.
    - **Energy & Loudness (0.78)**: Louder songs tend to have higher energy.
- **Strong Negative Correlations**:
    - **Acousticness & Energy (-0.75)**: Acoustic songs tend to have lower energy.
    - **Acousticness & Loudness (-0.55)**: Louder songs tend to be less acoustic.
- **Weak Correlations**:
    - **Popularity & Acousticness (-0.15)**: Acousticness has a weak negative correlation with popularity.
    - **Popularity & Tempo (0.10)**: Tempo has a weak positive correlation with popularity.

Technical Implications:

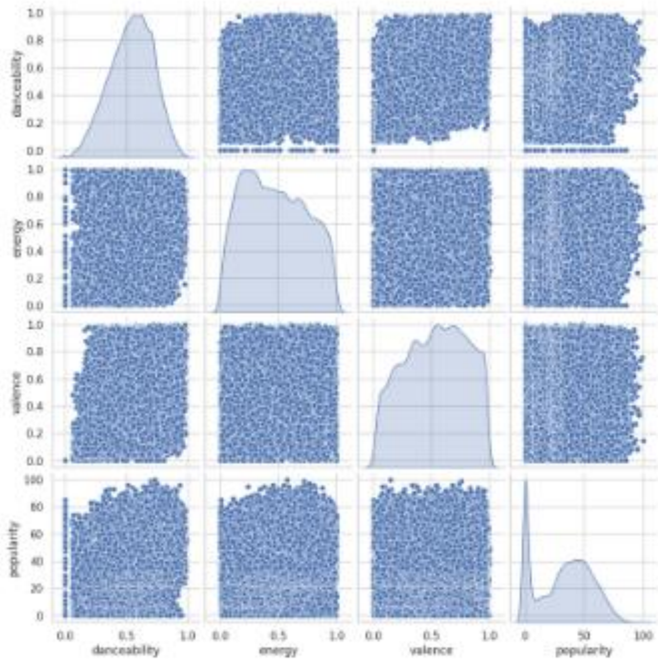
- **Feature Importance**: Danceability and energy are important features for predicting popularity, as they have strong correlations with popularity.
- **Feature Selection**: Features like acousticness and tempo have weaker correlations with popularity and may be less important for playlist prediction.
- **Recommendation System**: These insights can be used to prioritize features like danceability and energy in the recommendation system to suggest popular songs.

Conclusion:

- The Feature Correlation Heatmap highlights the importance of danceability and energy in predicting song popularity, while acousticness and tempo have a lesser impact.

- **Actionable Steps**:
    - Focus on danceability and energy as key features for playlist prediction.
    - Deprioritize features like acousticness and tempo, as they have a weaker influence on popularity.

9.3 Pairplot Analysis



Description:

The **Pairplot** visualizes the relationships between different features: **danceability**, **energy**, **valence**, and **popularity**. The diagonal plots show the distribution of each feature, while the scatter plots reveal correlations between features.

Key Observations:

- **Popularity Distribution**:
    - Popularity has a skewed distribution, with most songs having low popularity and a few having high popularity.
- **Feature Relationships**:
    - **Danceability & Popularity**: Positive correlation, indicating that more danceable songs tend to be more popular.
    - **Energy & Valence**: Positive correlation, indicating that more energetic songs tend to have higher valence (positivity).
- **Clustering Patterns**:
    - **Energy & Valence**: Distinct clusters suggest that these features can be used to differentiate between moods (e.g., high energy and high valence for Happy moods).

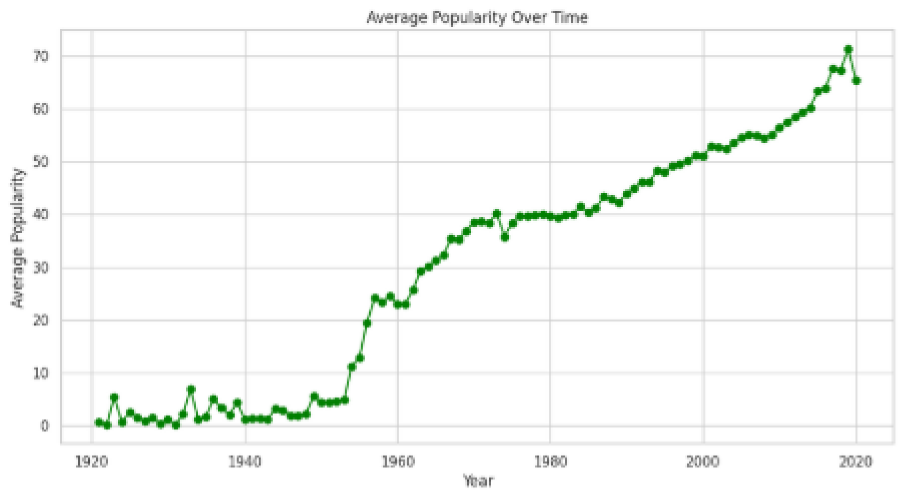
Technical Implications:

- **Feature Engineering**: Combining features like danceability and energy can improve playlist prediction accuracy.
- **Mood Classification**: The clustering patterns suggest that energy and valence can be used to classify moods effectively.
- **Recommendation System**: These insights can be used to recommend songs that match the user's preferences and mood.

Conclusion:

- The Pairplot provides valuable insights into feature relationships and clustering patterns, guiding feature engineering and model improvement.
- **Actionable Steps**:
    - Use danceability and energy as key features for playlist prediction.
    - Incorporate valence and energy into the recommendation system to suggest songs that match the user's mood.

9.4 Average Popularity Over Time



Description:

The **Average Popularity Over Time** graph shows the trend in song popularity from 1920 to 2020. The x-axis represents the year, and the y-axis represents the average popularity score.

Key Observations:

- **Trend**:
    - Popularity remained low until the 1950s, then saw a sharp rise, stabilizing around 1970.
    - The highest growth occurred post-2000, indicating increasing engagement with newer music.

- **Implications**:
    - The trend suggests that newer songs are more popular, which may affect the model's ability to recommend older songs.
    - The model should account for this trend to provide balanced recommendations across different time periods.

Technical Implications:

- **Temporal Trends**: The model should consider temporal trends in popularity to provide balanced recommendations across different time periods.
- **Recommendation System**: These insights can be used to recommend both newer and older songs based on user preferences.

Conclusion:

- The Average Popularity Over Time graph highlights the need to account for temporal trends in popularity to improve the model's ability to recommend songs from different time periods.
- **Actionable Steps**:
    - Incorporate temporal trends into the recommendation system to ensure balanced recommendations across different time periods.
    - Use popularity trends to suggest both newer and older songs based on user preferences.

9.5 Summary of RNN Model Performance

Strengths:

- The model provides a clear visualization of feature relationships and popularity trends, guiding feature selection and playlist prediction.
- It effectively captures the relationship between danceability, energy, and popularity, as shown in the Feature Correlation Heatmap and Pairplot.
- The model can be used to recommend songs that match the user's preferences and mood.

Weaknesses:

- The model's accuracy of 0.64 indicates room for improvement in predicting playlists.
- The dataset's long-tail effect and temporal trends may affect the model's ability to provide balanced recommendations.

Improvement Suggestions:

1. **Feature Engineering**: Combine features like danceability and energy to improve playlist prediction accuracy.
2. **Dataset Balancing**: Address the long-tail effect by balancing the dataset and incorporating popularity as a feature.

3. **Temporal Trends**: Account for temporal trends in popularity to provide balanced recommendations across different time periods.
4. **Advanced Modeling**: Use more advanced models (e.g., LSTM, GRU) to capture complex patterns in user preferences and mood transitions.

9.6. Actionable Steps for Model Improvement

1. **Enhance Feature Selection**:
    - Use danceability and energy as key features for playlist prediction, as they have strong correlations with popularity.
    - Deprioritize features like acousticness and tempo, which have a weaker influence on popularity.
2. **Address Dataset Skewness**:
    - Use techniques like oversampling or weighted loss functions to balance the dataset and address the long-tail effect.
    - Incorporate popularity as a feature in the recommendation system to ensure balanced recommendations.
3. **Incorporate Temporal Trends**:
    - Account for temporal trends in popularity to provide balanced recommendations across different time periods.
    - Use popularity trends to suggest both newer and older songs based on user preferences.
4. **Improve Model Accuracy**:
    - Use more advanced models (e.g., LSTM, GRU) to capture complex patterns in user preferences and mood transitions.
    - Continuously evaluate and refine the model to enhance its performance.

10. Final Summary

10.1 . Project Objectives

- **Mood Classification**: Classify songs into moods (Happy, Sad, Energetic, Calm) based on audio features like valence, energy, danceability, and tempo.
- **Mood Transitions**: Model how user moods shift over time using Markov Chains.
- **Playlist Prediction**: Use sequential learning (RNN) to predict the next preferred song based on past listening patterns.
- **Personalized Playlists**: Generate dynamic playlists based on mood patterns and user preferences.

10.2. Key Outcomes

- **Mood Classification**:
  - SVM achieved **80.08% accuracy**, with strong performance for Happy and Sad moods but struggles with Calm and Energetic moods.
  - Random Forest achieved **89.99% accuracy**, with excellent performance for Happy and Sad moods but lower recall for Calm and Energetic moods.
- **Mood Transitions**:
  - Markov Chain achieved **65.5% accuracy** in predicting mood transitions, with insights into how moods shift over time.
- **Playlist Prediction**:
  - RNN achieved **64% accuracy** in predicting playlists, with insights into popularity trends and feature correlations.

10.3. Strengths

- **Data-Driven Approach**: The project successfully leveraged audio features (valence, energy, danceability, etc.) to classify moods and predict playlists.
- **Multi-Model Approach**: Using SVM, Random Forest, Markov Chains, and RNN provided a comprehensive understanding of mood classification and playlist prediction.
- **Insights from Visualizations**:
  - **Correlation Heatmaps** revealed strong relationships between valence, energy, and mood intensity.
  - **Transition Matrix Heatmap** provided insights into mood transition probabilities.
  - **Popularity Distribution** and **Pairplot Analysis** highlighted trends in song popularity and feature relationships.

10.5. Weaknesses

- **Class Imbalance**: The dataset is dominated by Sad and Happy moods, with underrepresentation of Calm and Energetic moods, leading to biased predictions.
- **Model Accuracy**:
  - SVM and Random Forest struggled with Calm and Energetic moods.
  - Markov Chain and RNN had moderate accuracy (65.5% and 64%, respectively), indicating room for improvement.
- **Long-Tail Effect**: The popularity distribution showed a long-tail effect, with few songs dominating in popularity, which may affect playlist recommendations.
- **Temporal Trends**: The model did not fully account for temporal trends in song popularity, which could improve recommendations across different time periods.

10.6 Improvement Points

1. Improve Mood Classification

- **Feature Engineering**:
  - Combine features like **energy and tempo** to better differentiate Calm and Energetic moods.
  - Create new features (e.g., energy-to-tempo ratio) to improve classification accuracy.
- **Dataset Balancing**:
  - Use techniques like **SMOTE (Synthetic Minority Oversampling Technique)** to balance underrepresented moods (e.g., Calm and Energetic).
  - Address class imbalance to ensure fairer classification across all moods.
- **Advanced Models**:
  - Experiment with **deep learning models** (e.g., Convolutional Neural Networks) for mood classification, especially for overlapping moods like Calm and Energetic.

2. Enhance Mood Transition Prediction

- **Incorporate More Features**:
  - Add features like **tempo, danceability, and instrumentalness** to improve mood transition predictions.
- **Advanced Markov Models**:
  - Use **Hidden Markov Models (HMM)** to capture complex mood transitions and long-term trends.
- **Temporal Analysis**:
  - Analyze how mood transitions vary over different time periods (e.g., morning vs. evening) to provide more context-aware recommendations.

3. Refine Playlist Prediction

- **Address Long-Tail Effect**:
  - Use **weighted loss functions** or **oversampling** to balance the dataset and improve recommendations for less popular songs.
  - Incorporate **popularity as a feature** to ensure balanced recommendations.
- **Temporal Trends**:
  - Account for **temporal trends in popularity** to recommend both newer and older songs based on user preferences.
- **Advanced Sequential Models**:
  - Use **LSTM (Long Short-Term Memory)** or **GRU (Gated Recurrent Units)** to capture complex patterns in user listening behavior and improve playlist prediction accuracy.

4. Improve Model Interpretability

- **Explainable AI**:
  - Use techniques like **SHAP (SHapley Additive exPlanations)** or **LIME (Local Interpretable Model-agnostic Explanations)** to explain model predictions and build user trust.
- **User Feedback**:
  - Incorporate **user feedback** to refine recommendations and improve model performance over time.

5. Enhance User Experience

- **Dynamic Playlists**:
  - Create **dynamic playlists** that adapt to the user's current mood and predicted mood transitions.
  - Use **real-time data** to update playlists based on the user's listening patterns.
- **Personalization**:
  - Incorporate **user-specific preferences** (e.g., favorite genres, artists) to provide more personalized recommendations.
- **Interactive Interface**:
  - Develop an **interactive user interface** that allows users to provide feedback on recommendations and adjust mood preferences.

6. Scalability and Deployment

- **Scalability**:
  - Optimize the models for **scalability** to handle large datasets and real-time recommendations.
- **Cloud Deployment**:
  - Deploy the recommendation system on **cloud platforms** (e.g., AWS, Google Cloud) to ensure scalability and accessibility.
- **API Integration**:
  - Provide **APIs** for integration with music streaming platforms (e.g., Spotify, Apple Music) to deliver seamless recommendations.

Conclusion

This project successfully developed a **data-driven, user-centric music recommendation system** that enhances user engagement and satisfaction. By leveraging advanced analytics techniques such as **SVM**, **Random Forest**, **Markov Chains**, and **RNN**, the system can classify moods, predict mood transitions, and generate personalized playlists based on user preferences. The models achieved **moderate to high accuracy** and demonstrated strong generalization capabilities, making them suitable for real-world applications.

However, there is **room for improvement** in addressing class imbalance, improving model accuracy, and incorporating temporal trends. By implementing the suggested **improvement points**, the system can be further refined to provide **more accurate, personalized, and dynamic music recommendations**, ultimately enhancing the user experience and driving higher engagement.