Music Genre Recognition System- Project Report

Team Members – **Santhosh S R23EK038**

Mahalakshmi J R23EK019

Vaishnavi M S R23EK017

Sai Shreya R23EK031

---

1. Project Overview

This project builds a machine learning system that can automatically identify the genre of a music track. The system analyzes various features of audio files and predicts which genre the music belongs to (like rock, jazz, classical, hip-hop, etc.).

Final Result: 99.5% Accuracy - The model correctly identifies the music genre 995 out of 1000 times!

---

2. What Problem Does This Solve?

Imagine you have thousands of songs without labels, or you're building a music streaming app that needs to automatically categorize songs. Manually classifying music is time-consuming and subjective. This system automates the process using machine learning.

---

3. The Dataset

The project uses a dataset (CSV file) containing:

- **Audio features**: Things like tempo, rhythm, pitch, frequency components, and other sound characteristics
- **Genre labels**: The actual genre of each song (the answer we want to predict)

---

4. How the System Works (Step-by-Step)

Step 1: Loading the Data

```
df = pd.read_csv("dataset.csv")
```

The system reads the music dataset from a CSV file containing audio features and genre labels.

Step 2: Data Cleaning

Problem: Some data had missing values (marked as '?') or blank entries.

Solution:

- For numeric features (like tempo, frequency): Filled missing values with the median (middle value)
- For text features: Filled missing values with the most common value (mode)

This ensures every song has complete information for analysis.

Step 3: Encoding Categorical Data

Machine learning models work with numbers, not text. So genre names like "Rock" or "Jazz" were converted to numbers:

- Rock → 0
- Jazz → 1
- Classical → 2
- And so on...

Step 4: Handling Class Imbalance with SMOTE

Problem: Some genres might have many more songs than others (e.g., 1000 rock songs but only 100 classical songs).

Solution: Used SMOTE (Synthetic Minority Over-sampling Technique)

- Creates synthetic (artificial but realistic) examples of underrepresented genres
- Balances the dataset so each genre has similar representation
- Prevents the model from being biased toward popular genres

Step 5: Splitting Data

The data was divided into two parts:

- **Training set (80%)**: Used to teach the model
- **Testing set (20%)**: Used to evaluate how well the model learned

This split was done carefully to maintain the same genre proportions in both sets (stratified split).

Step 6: Training the CatBoost Model

What is CatBoost? CatBoost is an advanced machine learning algorithm that:

- Builds multiple decision trees (like a forest of decision-makers)
- Each tree learns from the mistakes of previous trees
- Combines all trees' predictions for a final decision

Model Configuration:

- **Iterations: 500** - Maximum number of decision trees to build
- **Depth: 8** - How complex each decision tree can be
- **Learning rate: 0.05** - How quickly the model learns (slower = more careful)

- **Early stopping: 50 rounds** - Stops training if accuracy doesn't improve for 50 consecutive iterations (prevents overfitting)

Step 7: Making Predictions

After training, the model was tested on the 20% of data it had never seen before to check if it can correctly identify genres of new songs.

---

5. Results

Overall Performance

- **Accuracy: 99.5%** (0.995)
- This means out of every 1000 songs tested, the model correctly identified the genre of 995 songs

What This Means

The model is extremely accurate and reliable for:

- Automatically categorizing large music libraries
- Music recommendation systems
- Playlist generation
- Music streaming services

---

6. Why CatBoost Was Chosen

Advantages of CatBoost:

1. **High accuracy**: Consistently performs better than other algorithms
2. **Handles mixed data**: Works well with both numeric and categorical features
3. **Built-in regularization**: Prevents overfitting automatically
4. **Fast training**: Efficient even with large datasets
5. **Robust**: Handles missing data and outliers well
6. **No manual tuning needed**: Works great with default settings

---

7. Technical Components Used

Libraries

- **pandas**: Data manipulation and analysis
- **numpy**: Numerical operations
- **scikit-learn**: Data preprocessing and evaluation metrics

- **imbalanced-learn (imblearn)**: SMOTE for handling imbalanced data
- **CatBoost**: The machine learning algorithm

Key Techniques

1. **Data Preprocessing**: Cleaning and preparing data
2. **Label Encoding**: Converting text to numbers
3. **SMOTE**: Balancing underrepresented classes
4. **Train-Test Split**: Evaluating model on unseen data
5. **Stratified Sampling**: Maintaining class proportions
6. **Gradient Boosting**: Ensemble learning technique

---

8. Real-World Applications

This system can be used for:

1. **Music Streaming Platforms** (Spotify, Apple Music)
   - Auto-categorize newly uploaded songs
   - Generate genre-based playlists
2. **Radio Stations**
   - Automatically organize music libraries
   - Create genre-specific programming
3. **Music Production**
   - Help producers understand genre characteristics
   - Suggest similar tracks
4. **Digital Music Libraries**
   - Automatically tag and organize personal music collections
   - Fix incorrect genre labels
5. **Music Discovery Apps**
   - Recommend songs based on genre preferences
   - Create mood-based playlists

---

9. Potential Improvements

While 99.5% accuracy is excellent, the system could be enhanced with:

1. **Deep Learning**: Using neural networks for even more complex pattern recognition

2. **Feature Engineering**: Adding more audio features (spectral features, MFCCs)
3. **Cross-Genre Detection**: Identifying songs that blend multiple genres
4. **Confidence Scores**: Showing how confident the model is in each prediction
5. **Real-time Processing**: Making predictions on live audio streams

---

10. Conclusion

This music genre recognition system demonstrates the power of modern machine learning. With 99.5% accuracy, it can reliably automate the classification of music, saving time and providing consistent categorization across large music libraries.

The combination of proper data preprocessing, SMOTE for handling imbalanced data, and the powerful CatBoost algorithm resulted in a highly accurate and production-ready system.

Key Takeaway: Machine learning can effectively solve complex audio classification problems with minimal manual intervention, achieving near-perfect accuracy when proper techniques are applied.
