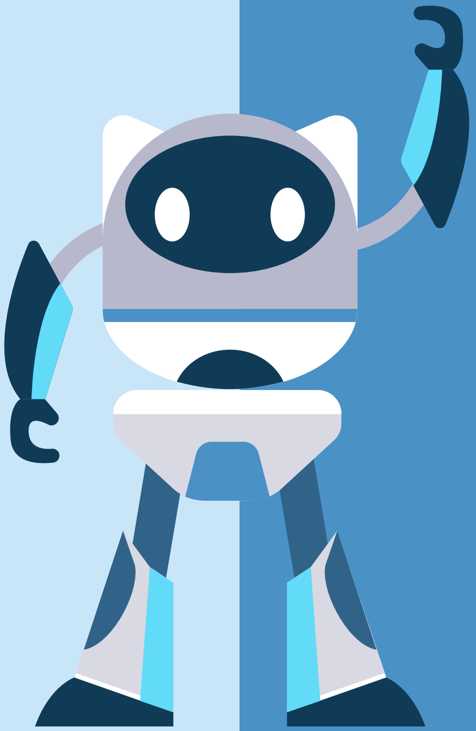
Music Genre Classification

Team Lead: Parijat Dhar

Team Members: Abhishek Shaw, Sumit Dey, Soham Dutta

Mentor: Sourav Goswami

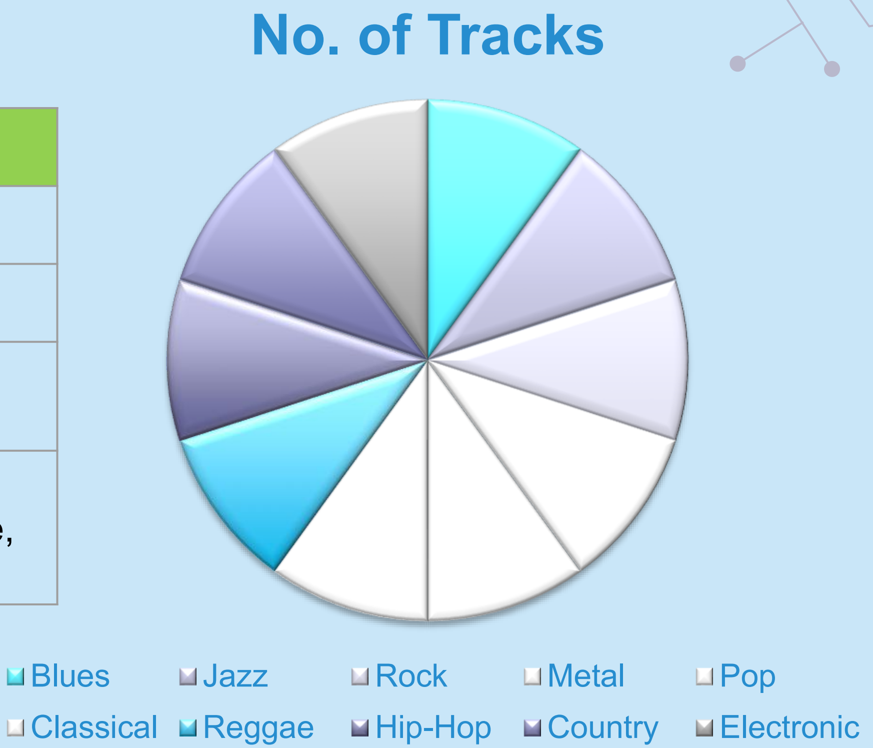


Motivation & Objective

- Growing music libraries require automatic categorization.
- Manual tagging is time-consuming and inconsistent.
- Objective: Develop a machine learning system to classify songs into genres automatically.

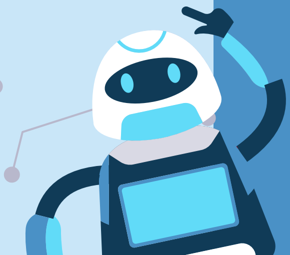
Dataset Overview

Attribute	Details
Dataset	GTZAN
Number of Tracks	1000 (30 seconds each)
Genres	10 (e.g., Blues, Jazz, Rock, Metal, etc.)
Features Extracted	MFCC, Chroma, Spectral Contrast, Zero Crossing Rate, Tempo



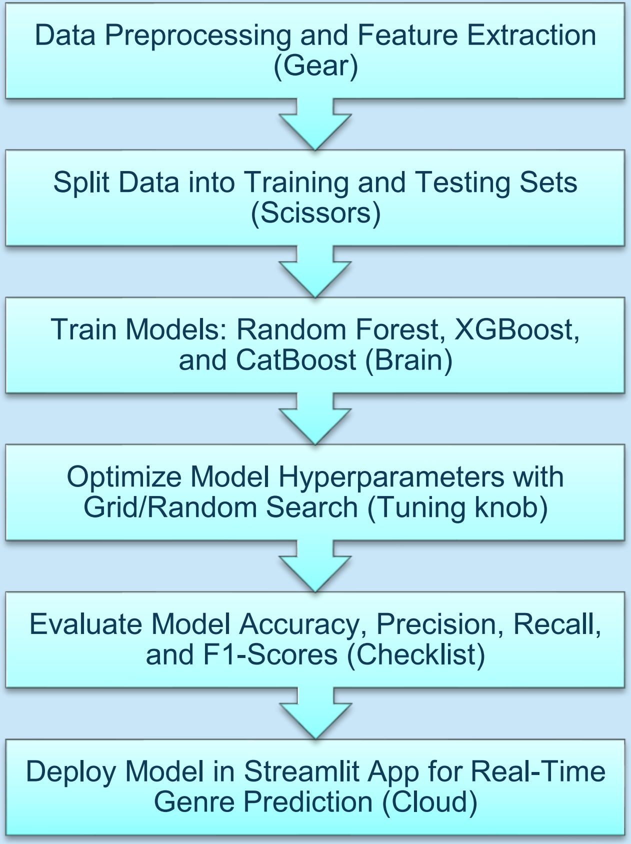
Audio Feature Explanation

Feature	Description	Visual Hint
MFCC	Captures the shape of the sound spectrum	Soundwave icon
Chroma	Identifies prominent pitch classes (notes)	Musical notes
Spectral Contrast	Measures difference between peaks and valleys in frequencies	Bar chart with peaks
Zero Crossing Rate	Counts how often signal changes sign, indicating noisiness/rhythm	Wave crossing zero line
Tempo	Overall speed of the song (beats per minute)	Metronome



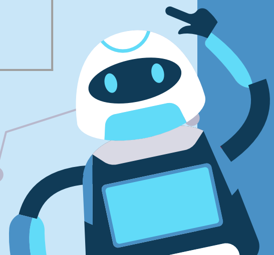
Methodology Workflow

1. Data preprocessing and feature extraction
2. Split data into training and testing sets
3. Train models: Random Forest, XGBoost, and CatBoost
4. Optimize model hyperparameters with grid/random search
5. Evaluate model accuracy, precision, recall, and F1-scores
6. Deploy model in Streamlit app for real-time genre prediction



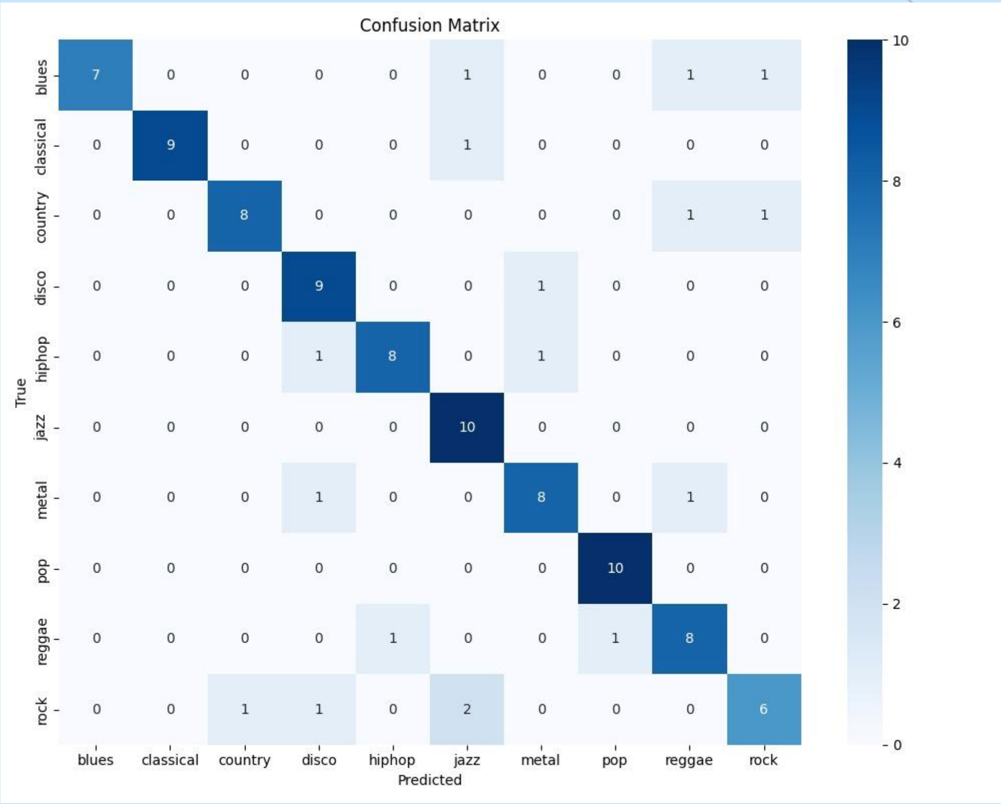
Model Comparison Table

Model	Accuracy	Advantages	Limitations
Random Forest	~66%	Easy to interpret, robust with noisy data	High-dimensional features cause overfitting
XGBoost	~79%	High accuracy, efficient	Sensitive to hyperparameters
CatBoost	~83%	Handles categorical features, reduces overfitting	Requires more compute, tuning effort



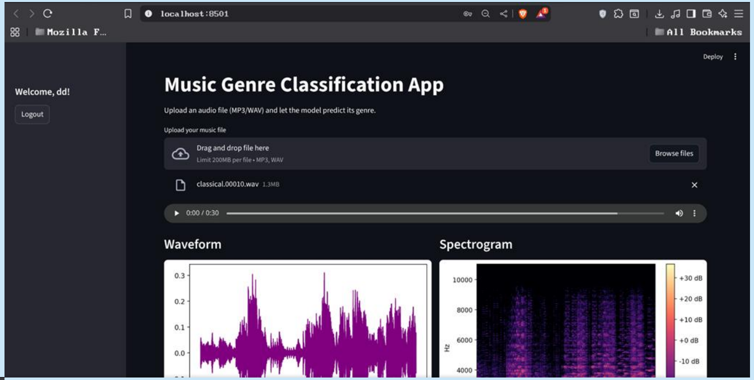
Performance Insights

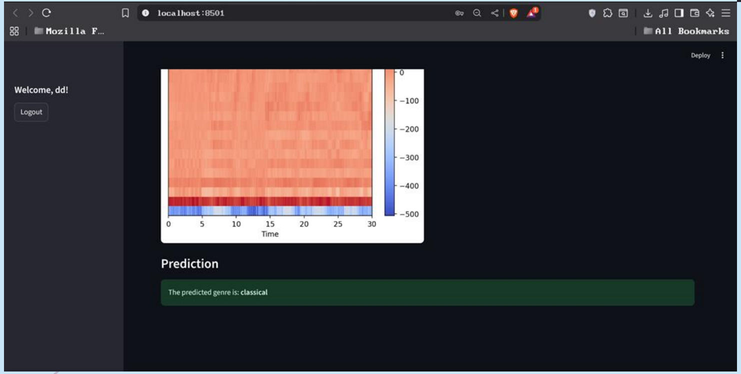
- Accuracy: 83% with CatBoost on test set
- Confusion Matrix: High precision in classical and jazz genres; more confusion among closely related genres like rock, metal, and reggae
- Feature Importance: Top 30 features (from MFCCs, chroma, tempo) drive final model predictions



Final Application – Streamlit Demo

- Upload audio (.mp3 or .wav)
- App visualizes waveform, spectrogram, MFCC heatmap
- Real-time prediction of song's genre with probability score





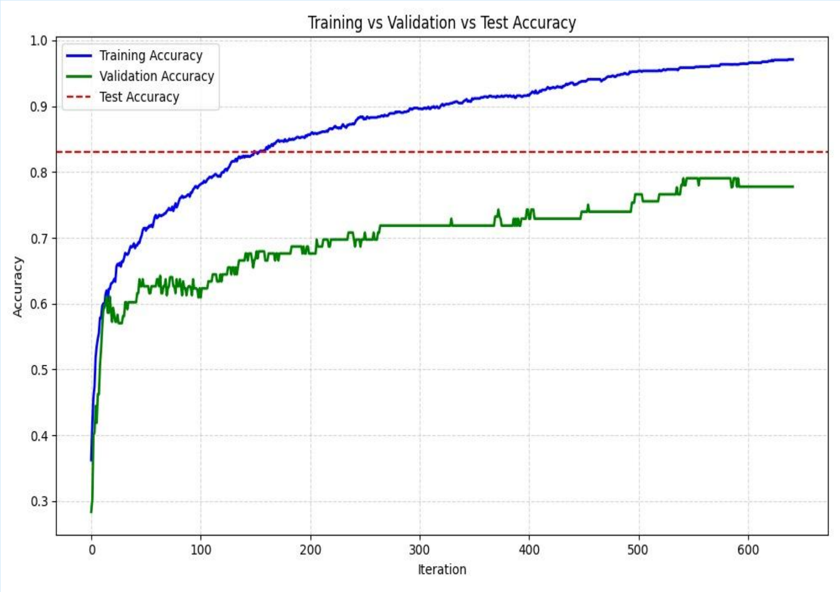
Limitations & Future Work

Limitations:

- Difficulty distinguishing similar genres (e.g., rock vs metal)
- Model relies on handcrafted audio features—may miss deep musical nuances
- Small dataset size limits generalizability

Future Directions:

- Expand dataset with more diverse music
- Explore deep learning models (CNNs for audio)
- Optimize app for mobile/web deployment

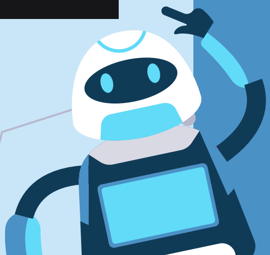


Summary & Takeaways

- Music genre classification helps organize massive audio databases
- Combined audio features and boosted tree models offer high accuracy
- CatBoost outperforms traditional classifiers by handling imbalanced classes and overfitting better
- Streamlined pipeline from audio processing to interactive web app showcase applied ML

Summary Bullet Points

- √ **Data Preprocessing and Feature Extraction:** Processed 1000 GTZAN tracks, extracted MFCC, Chroma, Spectral Contrast, Zero Crossing Rate, and Tempo.
- √ **Data Split:** Divided data into training and testing sets for model evaluation.
- √ **Model Training:** Trained Random Forest, XGBoost, and CatBoost models on the dataset.
- √ **Hyperparameter Optimization:** Fine-tuned models using grid/random search for improved performance.
- √ **Model Evaluation:** Assessed accuracy, precision, recall, and F1-scores on test data.
- √ **Deployment:** Integrated the best model into a Streamlit app for real-time genre prediction.



Thank You