Music Genre Classifier

Akshat Kuttan
Dept of CSE
Amrita School of Computing, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22104@bl.students.amrita.edu

Meghraj GK
Dept of CSE
Amrita School of Computing, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22129@bl.students.amrita.edu

Y Nishanth
Dept of CSE
Amrita School of Computing, Bangalore
Amrita Vishwa Vidyapeetham, India
bl.en.u4aie22165@bl.students.amrita.edu

***Abstract*—Music, as a universal language, is an integral part of human culture, with a diverse range of genres catering to various tastes and emotions.Music genre classification, the task of automatically assigning a genre label to a piece of music based on its audio content, plays a crucial role in achieving this organization. With the help of the classifiers such as K-NN and SVM can capture the intricate details of the audio or the song uploaded. The GTZAN dataset, a widely used benchmark dataset in the field of music genre classification, provides a rich resource for training and evaluating classification models.**

***Index Terms*—K-NN, SVM, GTZAN dataset**

I. Introduction

Music, as an art form, is deeply ingrained in human culture and society, serving as a means of expression, entertainment, and communication across diverse communities. With the advent of digital technologies and the proliferation of online music platforms, the sheer volume and variety of music available to listeners have expanded exponentially. However, this abundance of music also present challenges in terms of organizing, navigating, and discovering music that aligns with individual preferences.

The GTZAN dataset, a widely used benchmark dataset in the field of music genre classification, provides a rich resource for training and evaluating classification models. Comprising audio clips spanning ten distinct genres, including rock, jazz, blues, and electronic, the GTZAN dataset offers a diverse and representative sample of musical styles, making it an ideal testbed for exploring different classification techniques.

Exploration of music genre classification using the GTZAN dataset, with a particular focus on the application of KNN and SVM classifiers. Our objective is to assess the performance of these classifiers in accurately categorizing music genres based on extracted audio features. By examining the efficacy of various feature representations and evaluating classification performance metrics, we seek to elucidate the strengths and limitations of KNN and SVM classifiers in the context of music genre classification.

Music genre classification addresses these challenges by categorizing songs into distinct genres based on their stylistic and sonic characteristics. By automatically assigning genre labels to music tracks, genre classification algorithms facilitate music recommendation, playlist generation, and personalized content delivery. Moreover, they enable users to explore and discover new music within their preferred genres or across different musical styles.

II. Literature Survey

[10]Explores the effectiveness of machine learning algorithms, specifically k-nearest neighbor (k-NN) and Support Vector Machine (SVM), in predicting music genres1. The authors use Mel Frequency Cepstral Coefficients (MFCC) to extract features from the GTZAN dataset, which contains 1000 songs across 10 genres, and compare the performance of these algorithms with and without dimensionality reduction via principal component analysis (PCA). The study concludes that SVM outperforms k-NN in music genre classification, achieving an overall accuracy of 77% without dimensionality reduction, and suggests potential future work in exploring other genres, features, and classifiers.

[9]This paper discusses a music genre classification model built using Python. The model processes audio data and predicts its genre using the K-Nearest Neighbor method. Each song is divided into small sections, and 13 features are extracted from each section using the Mel Frequency Cepstral Coefficient (MFCC) method. The mean and covariance of these features represent the song. The authors use K-Fold Cross Validation to find the optimal value of K for the KNN model, with K=5 yielding the highest prediction accuracy of over 72%. The prediction accuracy varies across genres, with classical being the easiest to predict and rock being the hardest. The authors also identify feature 0 and 1 as key features to distinguish genres.

[8]The document discusses the application of machine learning algorithms for music genre classification using Mel-frequency cepstral coefficients (MFCCs) extracted from a dataset of 1000 songs across 10 genres. It compares the performance of four algorithms: Naive Bayes, k-means, k-medoids, and k-nearest neighbor, considering different distance metrics, data conditioning, and number of clusters or neighbors. The document reports the accuracy of each algorithm on different subsets of genres and suggests potential improvements and extensions for future work.

III. METHODOLOGY

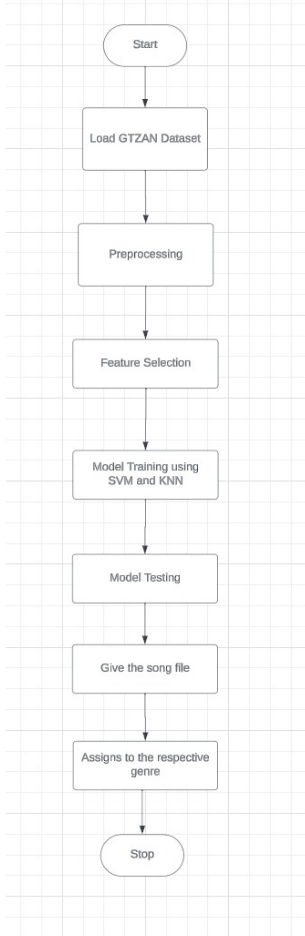

Fig. 1. Flow Chart

A. Data Collection and Pre-processing

1) Handling Categorical Data: Checked dataset for missing values, found none. Converted categorical data to numerical via one-hot encoding.

2) Intraclass Spread and Interclass Distances Evaluation: Calculated class centroids using numpy.mean() to represent mean vectors for each class.
Utilized numpy.std() to compute standard deviation vectors for class vectors. Applied appropriate axis properties (axis=0) to compute mean vectors across features. Quantified distances between mean vectors of different classes using numpy.linalg.norm() to find Euclidean distances.

3) Feature Density Observation: Observed density pattern for the ”monthly charges” feature to understand data distribution. Plotted a histogram for the feature using numpy.histogram(). Compared results with a normal distribution plot.

4) Distance Calculation: Calculated Minkowski distance for two features with r values ranging from 1 to 10. This quantified similarity or dissimilarity between points in multi-dimensional space, applicable for the dataset.

5) KNN Classifier: The dataset was initially split into training and testing sets with a test size of 40% using train_test_split. The training data was then used to train a kNN classifier with n_neighbors=3 utilizing KNeighborsClassifier from scikit-learn. The accuracy of the kNN model was subsequently evaluated on the testing data using the score method (neigh.score(X_test, y_test)). Finally, predictions were made on the testing set feature vectors using the predict method (neigh.predict(X_test)), allowing for the evaluation of its classification performance.

6) MLP classifier: In order to train and test the MLP classifier, we follow a similar approach to the k-NN classifier. Firstly, the data is split into training and testing sets. Subsequently, the MLP classifier is trained using the training data, with the number of hidden layers set to 100. After training, predictions are made on the testing data using the MLPClassifier() function. The predicted values (y_pred) are then compared with the actual values (y_test) to evaluate the accuracy of the model, calculated using the accuracy_score() function.

7) Evaluation: To comprehensively assess the performance of the classifiers, various steps are undertaken. For the kNN classifier, different values of the number of nearest neighbors (k) are considered, specifically k=1 and k=3. A confusion matrix is plotted to visually evaluate the classification results and derive additional performance measures such as precision, recall, and F1-Score for both training and test data.

Furthermore, an Area Under the Receiver Operating Characteristic (AUROC) curve is generated. This curve provides insights into whether the model is underfitting, overfitting, or fitting the dataset adequately, aiding in the assessment of model performance and generalization capability.

8) Random Forest: Random Forest presents a formidable approach for delving into the GTZAN dataset’s ”features_3_sec.csv,” housing various audio features. Harnessing the power of ensemble learning, Random Forest excels at uncovering intricate connections between these features and music genres, resulting in heightened predictive performance. Its adeptness in handling high-dimensional data and averting overfitting renders it particularly adept for the multifaceted feature landscape of GTZAN. Furthermore, Random Forest unveils insights into feature importance, shedding light on the pivotal elements driving genre classification. With its blend of scalability and efficacy, Random Forest emerges as a potent tool for crafting precise music genre classification models on the GTZAN dataset.

SVM Support Vector Machines (SVM) offer a robust methodology for exploring the GTZAN dataset’s ”features_3_sec.csv,” which comprises audio features. SVM excels in delineating complex decision boundaries, making it adept at discerning patterns within the feature space and accurately predicting music genres. Its ability to handle high-dimensional data and its flexibility in kernel selection enable SVM to effectively capture intricate relationships in the GTZAN dataset. Additionally, SVM provides interpretability through support vectors, aiding in understanding the discriminative features driving genre classification. With its versatility and accuracy, SVM serves as a potent tool for developing precise music genre classification models on the GTZAN dataset.

9) CatBoost: CatBoost, a robust gradient boosting algorithm, is well-suited for the GTZAN dataset's "features_3_sec.csv" file, containing audio features. Its efficient handling of categorical features without extensive preprocessing makes it ideal for predicting music genres accurately. By capturing intricate relationships between audio features and genres, CatBoost ensures high predictive performance. Additionally, its built-in features for handling missing values and detecting feature importance enhance model robustness and interpretability. Overall, CatBoost offers a potent solution for developing precise and reliable music genre classification models on the GTZAN dataset.

10) XGBoost: XGBoost stands as a leading choice for analyzing the GTZAN dataset's "features_3_sec.csv" file, which contains audio features. Its adeptness in handling mixed data types and ability to capture complex relationships make it ideal for precise music genre prediction. With its regularization techniques and superior performance, XGBoost ensures model stability and generalization. Its efficiency in training and inference further enhances its suitability for large datasets like GTZAN. Overall, XGBoost offers a robust solution for developing accurate and efficient music genre classification models.

11) Decision Tree: Decision trees provide a versatile method for analyzing the GTZAN dataset's "features_3_sec.csv," containing audio features. By partitioning the data based on feature thresholds, decision trees reveal underlying patterns in music genres. While susceptible to overfitting, techniques like pruning enhance model generalization. Moreover, decision trees offer intuitive insights into feature importance, aiding in understanding the significance of audio features in genre classification. Despite their simplicity, decision trees serve as a valuable tool for developing accurate music genre classification models on the GTZAN dataset.

12) Naive bayes: Naive Bayes presents an efficient approach to analyzing the GTZAN dataset's "features_3_sec.csv," containing audio features. Despite its simplistic assumption of feature independence, Naive Bayes often delivers competitive performance in classification tasks, including music genre prediction. Its quick training time and simplicity make it well-suited for handling large datasets like GTZAN. Moreover, Naive Bayes offers transparency in its decision-making process, allowing for easy interpretation of classification results. While it may overlook complex feature interactions, Naive Bayes serves as a dependable method for initiating music genre classification analyses on the GTZAN dataset

B. Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) serves to deepen our grasp of the dataset. It entails scrutinizing data distributions, detecting patterns, and probing correlations among variables. Employing visualization techniques, we illuminate relationships between variables, facilitating the selection of features conducive to robust churn prediction. We meticulously attend to feature importance and multicollinearity, ensuring that selected features substantially enhance the model's predictive prowess.

C. Feature Engineering:

Feature engineering plays a crucial role in enhancing the predictive capability of machine learning models. Features are strategically selected and engineered, taking into account domain knowledge and addressing missing values through various techniques such as mean imputation, forward or backward filling, and advanced imputation methods based on feature characteristics.

D. Data Splitting:

The dataset was initially split into training and testing sets to maintain balance for model training and evaluation. Additionally, techniques such as k-fold cross-validation were implemented to robustly assess the performance of the models. This approach aids in obtaining reliable estimates of model performance by iteratively splitting the data into k subsets and using each subset as a testing set while the rest serve as training sets, thus providing a comprehensive evaluation of the model's generalization capabilities.

E. Model Selection:

Initially, the dataset was split into training and testing sets, to maintain balance for model training and evaluation. Additionally, techniques such as k-fold cross-validation were implemented to robustly assess the performance of the models. This approach aids in obtaining reliable estimates of model performance by iteratively splitting the data into k subsets and using each subset as a testing set while the rest serve as training sets, thus providing a comprehensive evaluation of the model's generalization capabilities.

F. Model Training and Evaluation:

The selected models were trained on the training dataset, adjusting hyperparameters to optimize predictive accuracy while prioritizing simplicity. Evaluation on the testing dataset included metrics like accuracy, precision, recall, F1 score, and AUC-ROC, ensuring a comprehensive assessment. Business context guided metric selection, focusing on the trade-off between false positives and false negatives to align with objectives.

REFERENCES

[1] N. Pelchat and C. M. Gelowitz, "Neural Network Music Genre Classification," in Canadian Journal of Electrical and Computer Engineering, vol. 43, no. 3, pp. 170-173, Summer 2020, doi: 10.1109/CJECE.2020.2970144

[2] Tzanetakis, George, and Perry Cook. "Musical genre classification of audio signals." IEEE Transactions on speech and audio processing 10.5 (2002): 293-302.

[3] Cataltepe, Zehra, Yusuf Yaslan, and Abdullah Sonmez. "Music genre classification using MIDI and audio features." EURASIP Journal on Advances in Signal Processing 2007 (2007): 1-8.

[4] Basili, Roberto, Alfredo Serafini, and Armando Stellato. ”Classification of musical genre: a machine learning approach.” ISMIR. 2004.

[5] Chathuranga, Dhanith, and Lakshman Jayaratne. ”Automatic music genre classification of audio signals with machine learning approaches.” GSTF Journal on Computing (JoC) 3 (2013): 1-12.

[6] Kostrzewa, Daniel, Robert Brzeski, and Maciej Kubanski. ”The classification of music by the genre using the KNN classifier.” Beyond Databases, Architectures and Structures. Facing the Challenges of Data Proliferation and Growing Variety: 14th International Conference, BDAS 2018, Held at the 24th IFIP World Computer Congress, WCC 2018, Poznan, Poland, September 18-20, 2018, Proceedings 14. Springer International Publishing, 2018.

[7] Leartpantulak, Krittika, and Yuttana Kitjaidure. ”Music genre classification of audio signals using particle swarm optimization and stacking ensemble.” 2019 7th International Electrical Engineering Congress (iEECON). IEEE, 2019.

[8] Cast, John, Chris Schulze, and Ali Fauci. ”Music Genre Classification.” (2013).

[9] SShan, Yitong, and Alex Ziyu Jiang. ”Music Genre Classification.” (2020).

[10] Ali, M. A., & Siddiqui, Z. A. (2017). Automatic Music Genres Classification using Machine Learning1. International Journal of Advanced Computer Science and Applications, 8(8), 337-344.