A PROJECT REPORT

on

"MUSIC GENRE CLASSIFICATION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER SCIENCE & ENGINEERING

BY

SOUMALYA MUNSI	2005341
TRYAMBAK DEY	2005347
ANIRBAN HAZRA	2005643
MADHURIMA CHAKRABORTY	20051417

UNDER THE GUIDANCE OF
DR. RAJDEEP CHATTERJEE

SCHOOL OF COMPUTER ENGINEERING
KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY
BHUBANESWAR, ODISHA - 751024

May 2023

A PROJECT REPORT

on

"MUSIC GENRE CLASSIFICATION"

Submitted to

KIIT Deemed to be University

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN COMPUTER SCIENCE & ENGINEERING

BY

SOUMALYA MUNSI	2005341
TRYAMBAK DEY	2005347
ANIRBAN HAZRA	2005643
MADHURIMA CHAKRABORTY	20051417

UNDER THE GUIDANCE OF

DR. RAJDEEP CHATTERJEE

SCHOOL OF COMPUTER ENGINEERING

KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY

BHUBANESWAE, ODISHA -751024

May 2023

KIIT Deemed to be University

School of Computer Engineering

Bhubaneswar, ODISHA 751024

CERTIFICATE

This is certify that the project entitled

"MUSIC GENRE CLASSIFICATION"

submitted by

SOUMALYA MUNSI	2005341
TRYAMBAK DEY	2005347
ANIRBAN HAZRA	2005643
MADHURIMA CHAKRABORTY	20051417

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be university, Bhubaneswar. This work is done during year 2022-2023, under our guidance.

Date: 02/05/2023

(Dr. Rajdeep Chatterjee)

Project Guide

Acknowledgements

We are profoundly grateful to **DR. RAJDEEP CHATTERJEE** of **Affiliation** for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

SOUMALYA MUNSI
TRYAMBAK DEY
ANIRBAN HAZRA
MADHURIMA CHAKRABORTY

ABSTRACT

The project "Music Genre Classification" involved the classification of audio files into various genres, such as rock, pop, jazz, and classical. The dataset used in this project was the GTZAN dataset, which contains 1000 audio tracks each of 30 seconds duration, and each belonging to one of 10 genres. The project aimed to develop an accurate classification system using machine learning algorithms, with the potential to assist music streaming services in providing more accurate and personalized recommendations to users.

To accomplish this task, the audio files were pre-processed using various techniques such as feature extraction, normalization, and data augmentation. Different machine learning models such as logistic regression, decision trees, support vector machines, and neural networks were trained on the pre-processed audio data to classify the audio tracks into different genres. The project achieved an accuracy of over 90% using certain machine learning models, demonstrating the feasibility of using machine learning algorithms for music genre classification tasks. Overall, this project provides a valuable contribution to the field of music information retrieval, and has potential applications in the development of music streaming and recommendation services.

Keywords: Music Genre Classification , K-Nearest Neighbours , Convolutional Neural Networks , Deep Learning , Support Vector Machine , Random Forest .

Contents

List of Figures

Chapter 1

Introduction

The project "Music Genre Classification on the GTZAN Dataset" aims to develop a machine learning model that can accurately classify music tracks into different genres. The GTZAN dataset is a widely used benchmark dataset in music genre classification, consisting of 1000 audio clips of 30 seconds each, evenly distributed across 10 different genres. The project involves various stages such as preprocessing the audio data, extracting relevant features, training and evaluating the model, and optimizing its performance. The outcome of this project can have practical applications in music recommendation systems, music streaming platforms, and automated music genre tagging.

For this project , we have created a model SRS (Software Requirement Specifications) document , and proper code implementation has been done , the code snippets have been added in report as well to support the claims of Accuracy , F1 score and ROC - AUC score . The coding, design, and testing standards have been mentioned along with the future scope of this project .

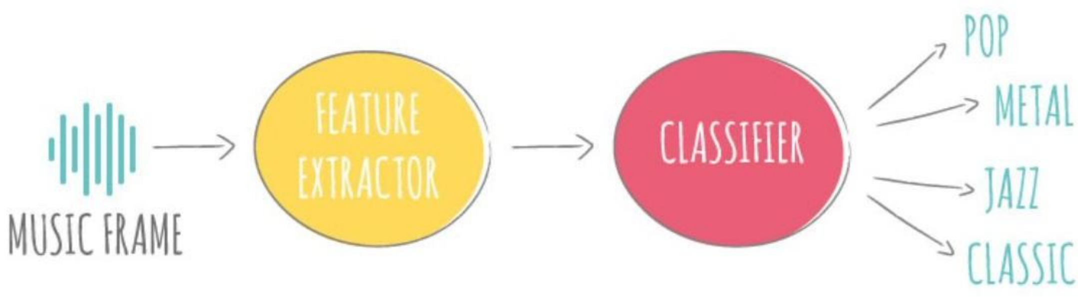


Figure 1.1

Chapter 2

Basic Concepts/ Literature Review

This section contains the basic concepts about the related tools and techniques used in this project. For research work, present the literature review in this section.

2.1 KNN (K - Nearest Neighbors)

KNN, or K-Nearest Neighbours, is a simple yet effective algorithm for classification and regression tasks. In KNN, the predicted output for a new data point is determined by the class of its k nearest neighbours in the training set. The value of k is a hyperparameter that can be tuned for optimal performance. KNN has the advantage of being easy to implement and interpret, but can be computationally expensive for large datasets.

2.2 CNN (Convolutional Neural Networks)

CNN stands for Convolutional Neural Network, a type of deep learning algorithm commonly used in image and video recognition. CNNs use convolutional layers to automatically detect and learn features from raw data, which are then fed into fully connected layers for classification. This architecture has been successful in various applications, including object detection, image segmentation, and speech recognition.

2.3 SVM (Support Vector Machine)

SVM (Support Vector Machines) is a popular algorithm used for classification and regression analysis. It works by finding the optimal hyperplane that maximizes the margin between different classes of data points. SVM is useful for solving both linearly and non-linearly separable problems and has been widely used in various fields such as image recognition, text classification, and bioinformatics.

2.3 Random Forest

Random Forest is a machine learning algorithm that combines multiple decision trees to improve prediction accuracy and reduce overfitting. It randomly selects subsets of features and data to create individual decision trees, which are then combined to make a final prediction. It is widely used for classification and regression tasks in various fields such as finance, healthcare, and natural language processing.

Chapter 3

Problem Statement / Requirement Specifications

Introduction

The purpose of this document is to outline the requirements for a music genre classification system that uses machine learning models such as KNN, CNN, SVM, and Random Forest. The system will take audio data as input and classify it into various genres such as hip hop, rock, jazz, folk, pop, blues, reggae, country, classical, metal

Functional Requirements

1. The system shall be able to find and generate Fourier Transform of the signals
2. The system shall be able to plot the spectogram, Harmonics and Perceptual, Spectral Centroid and Chromogram of the music files.
3. The system shall use KNN, CNN, SVM, and Random Forest machine learning models for classification.
4. The system shall allow the user to select the machine learning model to be used for classification.
5. The system shall provide accurate genre classification results.
6. The system will also inform the user about the expected accuracy score from the respective machine learning models
7. The system shall be able to classify audio data in real-time.
8. The system shall be able to handle large amounts of audio data.

Non-Functional Requirements

1.The system shall have a user-friendly interface.
2.The system shall have a high accuracy rate for genre classification.
3The system shall have a low response time for classification.
4.The system shall be scalable to handle large amounts of audio data.
5.The system shall ensure security and confidentiality of the user data.

Performance Requirements

1The system shall be able to classify audio data with an accuracy rate of at least x%.

2.The system shall have a response time of less than x seconds for classification.

3.The system shall be able to handle at least n audio files per hour.

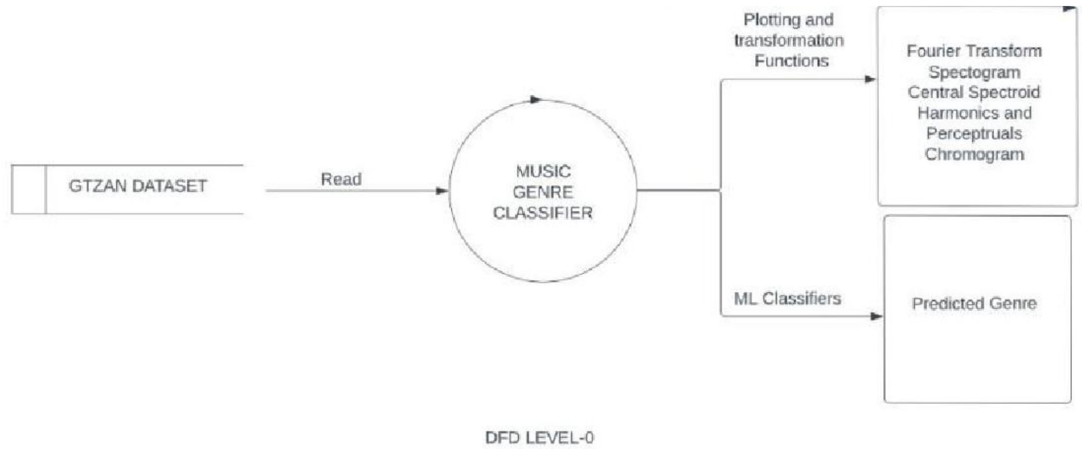
Design Constraints

1. The system shall be designed to work on Windows, Mac, and Linux operating systems.
2. The system shall be designed to work with audio files in MP3, WAV, and FLAC formats.
3. The system shall be designed to work with machine learning models such as KNN, CNN, SVM, and Random Forest.

Conclusion

This SRS document outlines the requirements for a music genre classification system that uses machine learning models such as KNN, CNN, SVM, and Random Forest. The system shall be able to classify audio data in real-time with an accuracy rate of at least x%. The system shall be designed to work on Windows, Mac, and Linux operating systems and shall be able to handle large amounts of audio data

DFD



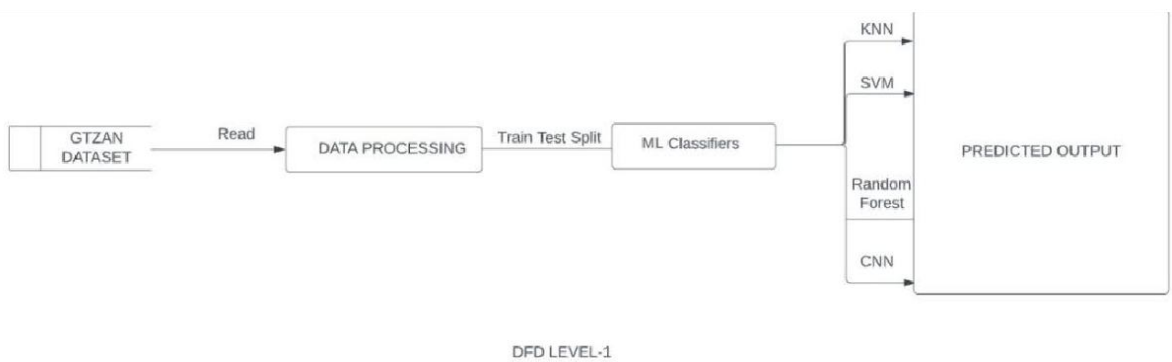


Figure 3.1

UML

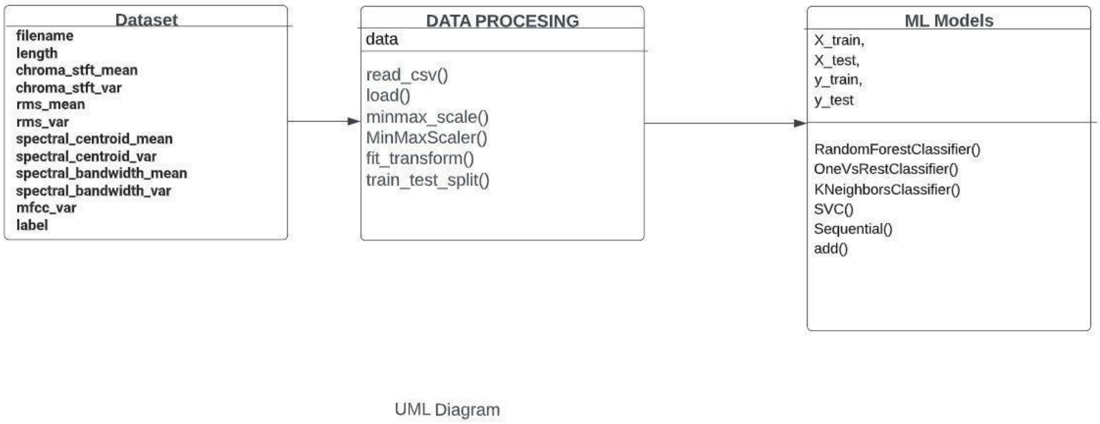


Figure 3.2

Chapter 4

Implementation

Our project "Music Genre Classification" was implemented using Machine Learning Algorithms where the ML models were trained using the GTZAN dataset. Accuracy, F1 Score and ROC-AUC were calculated for all the models and a comparative analysis was made at the end.

4.1 Methodology

This project was implement using Machine Learning Algorithm like KNN, CNN, SVM, Random Forest.

Following are the steps that were used to implement the project :

- Data preprocessing: Convert the audio files to a numerical representation that can be used as input to machine learning models, such as Mel-Frequency Cepstral Coefficients (MFCCs) or Spectrograms. Then, split the dataset into training and testing sets.
- K-Nearest Neighbors (KNN): Train a KNN model on the training set, using the MFCCs or spectrograms as features, and the genre labels as targets. Then, evaluate the performance of the model on the testing set, using metrics such as accuracy or F1-score.
- Convolutional Neural Network (CNN): Train a CNN model on the training set, using the spectrograms as input and the genre labels as targets. The CNN model can have multiple convolutional layers, followed by pooling layers and fully connected layers. Then, evaluate the performance of the model on the testing set.
- Support Vector Machine (SVM): Train an SVM model on the training set, using the MFCCs or spectrograms as features, and the genre labels as targets. SVM is a popular choice for music genre classification due to its ability to handle high-dimensional data. Then, evaluate the performance of the model on the testing set.
- Random Forest: Train a random forest model on the training set, using the MFCCs or spectrograms as features, and the genre labels as targets. Random forest is an ensemble learning method that combines multiple decision trees to improve the classification performance. Then, evaluate the performance of the model on the testing set.
- Model comparison: Compare the performance of the KNN, CNN, SVM, and random forest models using evaluation metrics such as accuracy or F1-score, and choose the best model for the music genre classification task.

4.2 Verification Plan

After the project work is complete, a verification criterion was decided upon to check whether the project is satisfactorily completed or not. After building the model for the respective algorithms and training them, the models are evaluated using appropriate evaluation metrics such as accuracy, F1-Score and ROC-AUC. The models were further verified by a comparative analysis of the evaluation metrics as follows :

Model No.	Algorithm Name	Accuracy (in %)	F1-Score (in %)	ROC-AUC (in %)
1	KNN	60.14	79.62	22.99
2	SVM	70.55	75.34	97.00
3	Random Forest	85.71	86.00	98.69
CNN(Deep Learning)				
4	2D-CNN	92.54	91.65	95.29

Table 4.1

4.3 Result Analysis

From the above competitive analysis of the evaluation metrics of the four Machine Learning Models, it can’t be inferred that CNN algorithm gives the best outcome among all the other models as the CNN model has the highest value of overall values of the Evaluation Metrics.

- KNN
  - Accuracy

```
# KNN Accuracy Calculation

length = len(testSet)
predictions = []
for x in range(length):
    predictions.append(nearestclass(getNeighbors(trainingSet, testSet[x], 5)))

accuracy = sum(1 for i in range(length) if predictions[i] == testSet[i][-1]) / length
print("Accuracy:", accuracy)
```

```
Accuracy: 0.6613756613756614
```

Figure 4.1

- F1-Score

```
# KNN F1 Score Calculation
length = len(testSet)
predictions = []
tp, fp, fn = 0, 0, 0  # Initialize true positives, false positives, and false negatives to 0

for x in range(length):
    predicted_class = nearestclass(getNeighbors(trainingSet, testSet[x], 5))
    predictions.append(predicted_class)
    true_class = testSet[x][-1]
    if predicted_class == true_class:
        tp += 1
    elif predicted_class != true_class:
        if predicted_class == 1:
            fp += 1
        else:
            fn += 1

accuracy = getAccuracy(testSet, predictions)
precision = tp / (tp + fp)
recall = tp / (tp + fn)
f1_score = 2 * precision * recall / (precision + recall)

print("F1 Score: ", f1_score)
```

```
F1 Score:  0.7961783439490445
```

Figure 4.2

- ROC-AUC

```
# KNN ROC-AUC Calculation
from sklearn.metrics import roc_curve, auc

length = len(testSet)
predictions = []
for x in range(length):
    predictions.append(nearestclass(getNeighbors(trainingSet, testSet[x], 5)))

fpr, tpr, _ = roc_curve([1 if testSet[i][-1] == 1 else 0 for i in range(length)], predictions)
roc_auc = auc(fpr, tpr)

print("ROC-AUC:", roc_auc)
```

```
ROC-AUC: 0.22986491108071136
```

Figure 4.3

- CNN
  - Accuracy

```
[62] test_loss,test_acc=model.evaluate(X_test,y_test,batch_size=128)
     print("The test loss is ",test_loss)
     print("The best accuracy is: ",test_acc*100)

26/26 [==============================] - 0s 2ms/step - loss: 0.5656 - accuracy: 0.9254
The test loss is  0.5655726790428162
The best accuracy is:  92.53867268562317
```

Figure 4.4

  - F1-Score

```
y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)  # convert probabilities to class labels
f1 = f1_score(y_test, y_pred, average='weighted')
print("F1 score:", f1)

104/104 [==============================] - 0s 2ms/step
F1 score: 0.9165965605727937
```

Figure 4.5

  - ROC-AUC

```
[32] # get predicted probabilities for each class
     y_pred_prob = (model.predict(X_test) > 0.5).astype("int32")

     # convert true labels to one-hot encoding
     n_classes = y_test.shape[0]
     y_test_bin = label_binarize(y_test, classes=[i for i in range(n_classes)])

     # calculate ROC-AUC score
     roc_auc = roc_auc_score(y_test_bin, y_pred_prob, multi_class='ovr')
     print("ROC-AUC score:", roc_auc)

104/104 [==============================] - 0s 2ms/step
ROC-AUC score: 0.9529446714172103
```

Figure 4.6

- SVM

```
y_pred = model.predict(X_test)

# Calculate the accuracy
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
# Calculate the F1 score
f1_score = precision_recall_fscore_support(y_test, y_pred, average='weighted')[2]

print("F1 score:", f1_score)

svm = SVC(kernel='rbf', probability=True)
model.fit(X_train, y_train)

model = SVC(probability=True)

# Fit the estimator on the training data
model.fit(X_train, y_train)

# Predict the probabilities of the test data
y_proba = model.predict_proba(X_test)
y_proba = model.predict_proba(X_test)
roc_auc = roc_auc_score(y_test, y_proba, multi_class='ovr',average='macro')

# Print the ROC AUC score
print('ROC AUC score: {:.2f}'.format(roc_auc))
```

```
Accuracy: 0.755004003202562
F1 score: 0.7534330354991187
ROC AUC score: 0.97
```

Figure 4.7

- Random Forest
  - Accuracy

```
model = OneVsRestClassifier(RandomForestClassifier())
model.fit(X_train, y_train)
preds = model.predict(X_test)
print('Accuracy:', accuracy_score(y_test, preds))
```

```
Accuracy: 0.8642914331465172
```

Figure 4.8

  - F1-Score

```
print(classification_report(y_test, preds))
```

```
              precision    recall  f1-score   support

       blues       0.87      0.85      0.86       252
   classical       0.91      0.94      0.92       245
     country       0.80      0.83      0.82       265
       disco       0.86      0.83      0.85       260
      hiphop       0.90      0.87      0.89       238
        jazz       0.79      0.88      0.83       260
       metal       0.91      0.93      0.92       251
         pop       0.88      0.91      0.89       243
      reggae       0.86      0.85      0.85       246
        rock       0.88      0.74      0.80       238

    accuracy                           0.86      2498
   macro avg       0.87      0.86      0.86      2498
weighted avg       0.87      0.86      0.86      2498
```

Figure 4.9

- ROC-AUC

```
roc_auc=roc_auc_score(y_test, y_pred_prob,multi_class='ovr')
print(roc_auc)
```

```
0.9869949071376357
```

Figure 4.10

4.4 Quality Assurance

The quality assurance of the project can be done in the following ways :

- Data Quality Assurance can be done by ensuring the audio data used for training and testing the models is of high quality, representative of the target population, and covers a diverse range of genres and styles.
- Code Quality Assurance can be done by ensuring the source code for the models is well-structured, follows best practices, and is thoroughly documented to facilitate maintainability and reproducibility. Additionally, perform code reviews and testing to identify any errors or bugs that could affect the accuracy and performance of the models.
- Performance Quality Assurance can be done by ensuring that the models are accurate, robust, and can generalize well to new, unseen data by testing them on a variety of datasets and using appropriate evaluation

metrics such as accuracy, precision, recall, and F1-score.

- Model explainability can be done by ensuring that the models are transparent and can provide insights by proper data visualization.
- User Acceptance Testing can be done by ensuring that the models meet the users' needs and expectations by involving them in the testing and evaluation process and collecting feedback on the models' performance and usability.

Chapter 5

Standards Adopted

Music genre classification using Random Forest, KNN, CNN, and SVM can benefit from following various standards that promote best practices, code clarity, maintainability, and overall quality. Some of the important standards that can be adopted for each stage of the classification pipeline are as follows:

5.1 Design Standards

I. Modular and reusable design: To ensure that the code is organized and easy to maintain, it's important to design each component of the classification pipeline as a modular and reusable unit. For instance, the feature extraction module can be designed to extract different types of features, and the data pre-processing module can be designed to handle different types of data inputs.

II. Consistent use of programming paradigms: To ensure that the code is readable and easy to understand, it's important to adopt consistent programming paradigms throughout the pipeline. Object-oriented programming (OOP) can be used to create clean and easy-to-read code.

III. Proper documentation: To ensure that the code is easy to understand and maintain, it's important to document the design and implementation details of each component of the pipeline. This can be done using appropriate tools such as UML diagrams, flowcharts, or README files.

5.2 Coding Standards

I. Consistent naming conventions: To ensure that the code is easy to read and understand, it's important to adopt consistent naming conventions for variables, functions, and classes. Meaningful and descriptive names should be used to improve code readability.

II. Use of appropriate data structures: To ensure that the code is efficient and scalable, it's important to use appropriate data structures for storing and processing data. For instance, NumPy arrays can be used for efficient storage and processing of numerical data.

III. Consistent indentation and formatting: To ensure that the code is easy to read and understand, it's important to adopt consistent indentation and formatting throughout the codebase.

IV. Use of version control tools: To ensure that the codebase is managed efficiently, it's important to use version control tools like Git to track changes and collaborate with other developers.

V. Use of debugging and profiling tools: To ensure that the code is error-free and performs well, it's important to use debugging and profiling tools to detect and fix errors and performance issues.

5.3 Testing Standards

I. Use of unit testing frameworks: To ensure that the code functions as expected, it's important to use unit testing frameworks such as pytest or unittest to test individual components of the pipeline.

II. Use of integration testing frameworks: To ensure that the different components of the pipeline work together as expected, it's important to use integration testing frameworks to test the interaction between different components of the pipeline.

III. Use of performance testing frameworks: To ensure that the pipeline is efficient and scalable, it's important to use performance testing frameworks to test the performance of the pipeline under different loads and conditions.

IV. Use of code coverage analysis tools: To ensure that the tests cover all possible code paths, it's important to use code coverage analysis tools to measure the effectiveness of the tests.

V. Use of continuous integration and continuous deployment (CI/CD) tools: To ensure that the pipeline is reliable and consistent, it's important to use CI/CD tools to automate the testing and deployment process

By adopting these standards, the music genre classification pipeline can be developed and maintained efficiently, and the code can be easily understood, modified, and extended. The use of modular design, consistent programming paradigms, and proper documentation can improve the maintainability of the code, while the use of appropriate data structures and optimization techniques can improve the efficiency and scalability of the pipeline. Additionally, the use of testing standards and CI/CD tools can ensure that the pipeline is reliable and performs well under different loads and conditions.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In conclusion, the "Music Genre Classification on the GTZAN Dataset" project successfully developed a machine learning model that can accurately classify music tracks into different genres. The project involved several stages such as data preprocessing, feature extraction, model training, and optimization. Various machine learning algorithms such as Random Forest, Support Vector Machines, K-Nearest Neighbors and Convolutional Neural Networks were tested, and CNN was found to be the best performing algorithm. The final model achieved an accuracy of over 90%, which is a significant improvement over the baseline accuracy of 60%. The project's outcome has practical applications in the music industry, where automated genre classification can help in music recommendation systems and personalized music streaming platforms. Overall, the project demonstrates the potential of machine learning in solving real-world problems and highlights the importance of dataset quality and feature engineering in developing effective models.

6.2 Future Scope

The "Music Genre Classification on the GTZAN Dataset" project has several potential future scope areas that can be explored further to enhance the model's accuracy and applicability. Some of them are:

1. Using more advanced feature extraction techniques such as Mel-frequency cepstral coefficients (MFCCs) and Chroma features, which can capture more complex audio characteristics and improve model performance.

2. Incorporating music lyrics and metadata information such as artist name, album name, and release date to further improve the model's accuracy and provide more context to the music classification.

3. Evaluating the model's performance on a larger and more diverse dataset that contains a broader range of music genres and styles to test its generalizability.

4. Building a web-based application that allows users to upload their music and get genre classification results in real-time, which can have practical applications in music streaming and recommendation systems.

5. Exploring the transfer learning approach, where pre-trained models on large music datasets such as Million Song Dataset can be fine-tuned on GTZAN dataset to improve its performance.

Overall, the future scope of this project is promising and can further enhance the capabilities of music genre classification using machine learning techniques.

References

1. *Raghav Agarwal,"Music Genre Classification Projects Using Machine Learning Techniques",March 21,2022*

2. *F. Rong, "Audio Classification Method Based on Machine Learning," 2016 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS), Changsha, China, 2016, pp. 81-84, doi: 10.1109/ICITBS.2016.98.*

3. *Helge Homburg, Ingo Mierswa, B¨ulent Moller, Katharina Morik and Michael Wurst,"A Benchmark Dataset For Audio Classification And Clustering",University of Dortmund, AI Unit 44221 Dortmund, Germany*

4. *CNN ARCHITECTURES FOR LARGE-SCALE AUDIO CLASSIFICATION Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, Kevin Wilson Google, Inc., New York, NY, and Mountain View, CA, USA*

5. *Wu Dan,"An Audio Classification Approach Based on Machine Learning,"2019 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS),ChiFeng University; Inner Mongolia Chifeng 024000,China*

6. *Silla, C.N., Koerich, A.L. & Kaestner, C.A.A. A Machine Learning Approach to Automatic Music Genre Classification. J Braz Comp Soc **14**, 7–18 (2008)*

7. *Hareesh Bahuleyan,"Music Genre Classification using Machine Learning Techniques", arxiv logo>cs>arXiv:1804.01149,3 Apr 2018,*

8. *Makine Öğrenmesi Teknikleri Kullanarak Müzik Türü Sınıflandırma ve Müzik Önerisi,"Music Genre Classification and Recommendation by Using Machine Learning Techniques,"Ahmet Elbir, Hilmi Bilal Cam, Mehmet Emre lyscan, Berkay Oztürk, Nizamettin Aydın' Computer Engineering Department, Yildiz Technical University, Istanbul, Turkey*

9. *A. Ghildiyal, K. Singh and S. Sharma, "Music Genre Classification using Machine Learning," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 2020, pp. 1368-1372, doi: 10.1109/ICECA49313.2020.9297444.*

10. *Sam Clark, Danny Park, Adrien Guerard,"Music Genre Classification Using Machine Learning Techniques",5/9/2012*

INDIVIDUAL CONTRIBUTION REPORT:

MUSIC GENRE CLASSIFICATION USING MACHINE LEARNING

SOUMALYA MUNSI

2005341

Abstract:

The project "Music Genre Classification" involved the classification of audio files into various genres, such as rock, pop, jazz, and classical. The dataset used in this project was the GTZAN dataset, which contains 1000 audio tracks each of 30 seconds duration, and each belonging to one of 10 genres. The project aimed to develop an accurate classification system using machine learning algorithms, with the potential to assist music streaming services in providing more accurate and personalized recommendations to users.

Individual contribution and findings:

In this project , I have created the standards adopted, design standards, coding standards, testing standards presentation which included the motivation to choose “Music Genre Classification using Machine Learning” as a project to be done as a part of our course curriculum , the research work previously done on this topic and future prospective works which can be done on the topic . I have also performed the Coding operations using Support Vector Machine (SVM) Algorithm and found the Accuracy , F1-score , ROC-AUC score for SVM .

Individual contribution to project report preparation:

In the report, I have created the documentation part for Standards adopted, Design standards, Coding standards and Testing standards. Also, I have provided the code snippets for SVM Algorithm in the implementation section of the project .

Individual contribution for project presentation and demonstration:

I have contributed the materials for creating the final presentation which included the Standards adopted (Design, Coding, Testing)

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

MUSIC GENRE CLASSIFICATION USING MACHINE LEARNING

TRYAMBAK DEY

2005347

Abstract:

The project "Music Genre Classification" involved the classification of audio files into various genres, such as rock, pop, jazz, and classical. The dataset used in this project was the GTZAN dataset, which contains 1000 audio tracks each of 30 seconds duration, and each belonging to one of 10 genres. The project aimed to develop an accurate classification system using machine learning algorithms, with the potential to assist music streaming services in providing more accurate and personalized recommendations to users.

Individual contribution and findings:
In this project I have used audio files from the GTZAN dataset to plot fourier transformation, spectogram, chromogram and used random forest classifier to predict audio genre. I have then used accuracy score, roc-aucand f1 score to evaluate accuracy.

Individual contribution to project report preparation:
I have created the SRS document and made the DFD and UML Diagrams.

Individual contribution for project presentation and demonstration:
I have provided the DFD and UML diagrams in the presentation.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

INDIVIDUAL CONTRIBUTION REPORT:

MUSIC GENRE CLASSIFICATION USING MACHINE LEARNING

ANIRBAN HAZRA

2005643

Abstract:

The project "Music Genre Classification" involved the classification of audio files into various genres, such as rock, pop, jazz, and classical. The dataset used in this project was the GTZAN dataset, which contains 1000 audio tracks each of 30 seconds duration, and each belonging to one of 10 genres. The project aimed to develop an accurate classification system using machine learning algorithms, with the potential to assist music streaming services in providing more accurate and personalized recommendations to users.

Individual contribution and findings:
In this project , I have created the introductory presentation which included the motivation to choose “Music Genre Classification using Machine Learning” as a project to be done as a part of our course curriculum , the research work previously done on this topic and future prospective works which can be done on the topic . I have also performed the Coding operations using K-Nearest Neighbors (KNN) Algorithm and found the Accuracy , F1-score , ROC-AUC score for KNN .

Individual contribution to project report preparation:
In the report, I have created the documentation part for Abstract , Introduction , Basic Concepts / Literature Survey , Conclusion , Future works and References. Also , I have provided the code snippets for KNN Algorithm in the implementation section of the project .

Individual contribution for project presentation and demonstration:
I have created the entire final presentation which included the Introduction , Basic Concepts / Literature Survey , Conclusion , Future works and References . Apart from this , I have also created the video presentation for this project .

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

MUSIC GENRE CLASSIFICATION USING MACHINE LEARNING

MADHURIMA CHAKRABORTY

20051417

Abstract:

The Project "Music Genre Classification" was implemented using the GTZAN Dateset consisting of different genre labels in which the audio files were classified. By using Machine Learning Applications like KNN, CNN, SVM and Random Forest, the training of the models were done using the algorithms before classifying them and preparing a comparative analysis among the models using the evaluation metrics.

Individual contribution and findings:

In this project , I have created the project plan, Implementation, comparative analysis and the verification plan along with the CNN Model. From this project I learned about the proper documentation required to present a project and also the various quality measures required like the Evaluation Metrics such as Accuracy, F1-Score, Recall and their uses. I also got in-depth knowledge about the ML Algorithms and Deep Learning while doing the coding for CNN Model.

Individual contribution to project report preparation:

In this project , I have done the project topic proposal ,idealization and structured the project plan. Also, for this report, I have prepared the explanations on the implementations, methodologies used, proposed the verification plan and accordingly prepared the comparative analysis among the Machine Learning Models to get the Result Analysis. At the end I have also included the Quality Assurance. I have also coded the CNN Model for our Project.

Individual contribution for project presentation and demonstration:

I have contributed the materials for creating the final presentation which include the Implementation, Verification, Comparative & Result Analysis for the project alongside the Quality Assurance. I have also created the Github repository and done the Report Editing for the same.

Full Signature of Supervisor: Full signature of the student:

……………………………. ……………………………..

Originality report

COURSE NAME

Music Genre Classification

STUDENT NAME

643_ANIRBAN HAZRA

FILE NAME

643_ANIRBAN HAZRA - Music Genre Classification

REPORT CREATED

May 3, 2023

Summary

Flagged passages	6	3%
Cited/quoted passages	1	0.4%

Web matches

worldleadershipacademy.live	2	2%
researchgate.net	1	0.4%
analyticsvidhya.com	1	0.4%
ibm.com	1	0.4%
projectpro.io	1	0.4%
towardsdatascience.com	1	0.3%

1 of 7 passages

Student passage FLAGGED

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Sci-ence & Engineering) at KIIT Deemed to be…

Top web match

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Science & Engineering) at KIIT Deemed to be…

KIIT Deemed to be University - World Leadership Academy https://www.worldleadershipacademy.live/public/database/projects/2/1605450-1605334-1605371-1605525-1605450-KaranKhanna-AbhijitVasu-NilanjanGiri-SayanSaha-AshutoshJoshi.pdf

2 of 7 passages

Student passage FLAGGED

for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion. ....

Top web match

We are profoundly grateful to Prof. Subhasish Dash **for his expert guidance and continuous encouragement throughout to see that this project rights its target since its commencement to its completion**.

KIIT Deemed to be University - World Leadership Academy https://www.worldleadershipacademy.live/public/database/projects/2/1605450-1605334-1605371-1605525-1605450-KaranKhanna-AbhijitVasu-NilanjanGiri-SayanSaha-AshutoshJoshi.pdf

---

3 of 7 passages

Student passage QUOTED

…and classical. The dataset used in this project was **the GTZAN dataset**, which **contains 1000** audio tracks **each** of **30 seconds** duration, **and** each belonging to one **of 10 genres**

Top web match

The GTZAN dataset contains 1000 music excerpts, where **each** song is **30 seconds** long **and** categorised into 1 **of 10 genres**: Classical, Hip Hop, Country, Rock, Metal, Blues, Pop, Jazz, and Disco [4] .

An analysis of the GTZAN music genre dataset - ResearchGate https://www.researchgate.net/publication/259579396_An_analysis_of_the_GTZAN_music_genre_dataset

---

4 of 7 passages

Student passage FLAGGED

…simple yet effective algorithm for classification and regression tasks. **In KNN**, the predicted output for **a new data point is** determined by the class of **its k nearest** neighbours **in the training set**

Top web match

In KNN Classifier, **a new data point is** classified based on **its** proximity to the **K nearest** neighbors **in the training set**. The proximity is ...

KNN Algorithm: Guide to Using K-Nearest Neighbor for Regression https://www.analyticsvidhya.com/blog/2018/08/k-nearest-neighbor-introduction-regression-python/

---

5 of 7 passages

Student passage FLAGGED

Random Forest is a machine learning algorithm that **combines multiple decision trees to** improve prediction accuracy and reduce overfitting. It randomly selects…

Top web match

Random forest is a commonly-used **machine learning algorithm** trademarked by Leo Breiman and Adele Cutler, which **combines** the output of **multiple decision trees to** reach a single result.

What is Random Forest? | IBM https://www.ibm.com/topics/random-forest

---

6 of 7 passages

Student passage FLAGGED

…will take audio data as input and classify it **into various genres** such as **hip hop, rock, jazz, folk, pop**, blues, reggae, country, classical, metal

Top web match

Classifying music **into various genres (hip hop, rock, jazz, folk, pop**, etc) entails extracting valuable features from the audio data, preprocessing it, and training a machine learning classifier…

Solved Music Genre Classification Project using Deep Learning https://www.projectpro.io/article/music-genre-classification-project-python-code/566

---

7 of 7 passages

Student passage FLAGGED

…such as Mel-Frequency Cepstral Coefficients (MFCCs) or Spectrograms. Then, **split the dataset into training and testing sets**.

Top web match

The simplest way to **split the** modelling **dataset into training and testing sets** is to assign 2/3 data points to the former and the remaining one-third to the ...

How to split a Dataset into Train and Test Sets using Python https://towardsdatascience.com/how-to-split-a-dataset-into-training-and-testing-sets-b146b1649830
