Identification and Detection of Plagiarism in Music using Machine Learning Algorithms

MSc Research Project
Data Analytics

Rajesh Ramachandran Nair
Student ID: 20141289

School of Computing
National College of Ireland

Supervisor: Dr Catherine Mulwa

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Rajesh Ramachandran Nair
Student ID:	20141289
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr Catherine Mulwa
Submission Due Date:	16/08/2021
Project Title:	Identification and Detection of Plagiarism in Music using Machine Learning Algorithms
Word Count:	6084
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	23rd September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Identification and Detection of Plagiarism in Music using Machine Learning Algorithms

Rajesh Ramachandran Nair
20141289

Abstract

The occurrence of a substantial resemblance between two words is the closest description offered of plagiarism. It's no surprise because song similarity is based on so many distinct elements and their combinations that it's nearly difficult to combine them all into a single final rule of thumb. The combination of similarities between the rhythm and melody can be suspected of plagiarism. In this paper, we will do research based on the identification of plagiarism in music The dataset used here will be a simple set of MIDI files that have got only the melody track. First feature extraction has been performed here to extract the note or the chord progression then, the harmonic reduction is performed to understand the structure of the music and then using Word2Vec model is applied to get the relationship between similar chords to perform chord substitution which will be the final data that is extracted for the classifier models(KNN, Logistic Regression, Random Forest, Decision Tree, Gaussian Naive Bayes) to predict plagiarism and the results were obtained. After the training where quite interesting Naive Bayes performed poorly but among the 4 models, Random Forest performed with the highest accuracy of 98% after the model was trained for threshold value 0.5. These models have trained again with various other threshold values and the appropriate results were obtained.

1 Introduction

Music plagiarism is defined as the use or near copying of another author's music without appropriate acknowledgment. Every year, a significant number of new music songs are released all over the world, and certain portions of songs contain questionable parallels. Plagiarism is now apparent all across the world, not just among writers but also between languages and countries, due to the internet. In 2008, 1.4 billion music albums were sold worldwide. Since then, the figure has increased to more than 1.8 billion people. Recently, (Schuitemaker; 2020)Katy Perry has been sued for 2.8 million dollars for a lawsuit saying her Dark Horse Song is the same as the Joyful Noise of Marcus Grey. This event sparked a debate among artists who were unclear about the plagiarism rules. The law is a little unclear on this subject. The legislation stipulates that work is plagiarized if the claimant has 'access to the copyrighted music and the two songs are substantially identical' in the instance of melodic theft.

According to a study plagiarism can be detected concerning the similarity in melody and rhythm in some cases it is just the similarity in the melody that can be a key factor. Usually, it is the combination of the similarities in the melody and rhythm. In most cases, we cannot consider rhythm alone to be a key factor to decide whether the music is plagiarized. Music plagiarism can be suspected under the following 1) (Lee et al.; 2011)When two music has similar successive melody notes, plagiarism can be suspected. 2)When two music shares the unique parts of the melody which is rarely used in others, plagiarism can be suspected. 3) When music has similar melody progress with different keys and instruments, plagiarism can be suspected.

In this paper, we will be looking at the identification of plagiarism in music using Machine Learning Algorithms which are used to classify text. We will be dealing with Algorithms like Gaussian Naive Bayes, Logistic Regression, Random Forest classifier, Decision Tree classifiers, and KNN to make the classification in the text before feeding the data to the models we need to preprocess and extract Melody or the chord progression in the music. Since the dataset contains only the Electric Guitar instrument which makes the only melody which is the chord progression to extract, Then harmonic reduction and chord substitution are performed using the help of the Word2Vec model to find the relationship between similar chords to perform Chord Substitution. Then before feeding the data It is fed to it at 4 different threshold values to check whether at which threshold value the classification is performed with the highest Accuracy. Those values were plotted with appropriate graphs to understand the efficiency of the model.

1.1 Research Question

RQ: "To what extend can the detection of Plagiarism in Music be improved by calculating the Similarity Index to reduce plagiarism to support musicians to create music with complete novelty" ?
Sub-RQ: Can the identification methods be used to find the plagiarism threshold value for detecting plagiarism in music?

1.2 Research Objectives and Contributions

The Main objective is to identify music plagiarism
Obj1) Critical review of the literature related to detection of plagiarism in music.
Obj2) Implementation, Evaluation and results of the Classification models using machine learning technologies
Sub-Obj2.1) Perform MIR(Music Information Retrieval) to extract melody on all the MIDI(Music Instrumental Digital Interface) files using Word2Vec.
Sub-Obj2.2) Implement, evaluate and results of Gaussian Naive Bayes.
Sub-Obj2.3) Implement, evaluate and results of Logistic Regression.
Sub-Obj2.4) Implement, evaluate and results of Random Forest Classifier.
Sub-Obj2.5) Implement, evaluate and results of Decision Tree Classifier.
Sub-Obj2.6) Implement, evaluate and results of K-Nearest Neighbours
Obj3) Compare the results of the above algorithms to find which among them performs well to give the minimum similarity index value.
Obj4) Visualize the results obtained from the algorithm to understand at which plagiarism threshold value the models perform with higher accuracy.

The structure of the paper is written in the following format Section 2 presents the literature review on the models, Feature extraction and identification of plagiarism in the field of music, Section 3 deals with methodology approach in the field of detecting plagiarism in music. . Section 4 deals with Design Specification Implementation, Evaluation and results obtained after prepossessing the data and results obtained from training the model.Section 5 Confirms the results obtained from the trained model and insights are drawn and also the future work based on the topic is discussed.

2 Related Work

2.1 Review on Identification of plagiarism in music

Plagiarism has been a source of contention in recent years, particularly in industries that may generate large sums of money, such as music. However, present systems for detecting plagiarism, i.e., duplicating someone else's work and passing it off as your own, rely mostly on superficial and brute-force string matching approaches. (Li and Han; 2013)Such well-known metrics, which are frequently used to identify similarities in texts, could not be utilized to discover similarities in music compositions. Because the semantics of words are not taken into account, this might be a problem when the text is used to represent a piece of music, in which case the semantics of words (note sequence) is a crucial component to discover similarities.

Despite the popular perception that a few notes in common between two songs are enough to determine whether plagiarism exists, analyzing similarities is a fairly complicated procedure in bag-of-words encoding,(De Prisco et al.; 2017a)(Orkphol and Yang; 2019) this might be a problem when the text is used to represent a piece of music, in which case semantics of words (note sequence) is a crucial component to discover similarities. Despite the popular perception that a few notes in common between two songs are enough to determine whether plagiarism exists, analyzing similarities is a fairly complicated procedure.

This subsection will be utilized to review the techniques and methods used by the previous research papers about the detection of plagiarism in music.The study by (De Prisco et al.; 2017a)(Kadhim; 2019)(González-Carvajal and Garrido-Merchán; 2020) They show how the benefits of a textual representation of music can be exploited by a plagiarism detection system based on two computational intelligence modules: an unsupervised machine learning algorithm to retrieve similar melodies, and a fuzzy deep analyzer to disambiguate. The accuracy rate was 96.4 percent, according to the results.

Though there are many ways to find plagiarism in music like sampling plagiarism, rhythmic plagiarism, melodic plagiarism(Dittmar et al.; 2012) most of the studies use melodic as the main feature extracted from their respective datasets because according to the paper. [1], Rhythmic extraction alone won't help in determining the plagiarism in music because it might end up giving the wrong notion about whether the music is plagiarized or not. So it is better to extract the melody of the song because it is the best structure of the song for accurate results.

Identification of the melody track inside a MIDI file which is then a compressed file version of audio data. A paper by (De et al.; 2015) used NMF which is a non-negative

[1] https://www.icce.rug.nl/~soundscapes/VOLUME18/Plagiarism_or_inspiration.shtml

matrix factorization technique to store the features in the forms of vectors. They extracted the feature from the Audio signals using monoaural signal separation stored those values in the form of vectors these vectors were used for DTW(Dynamic Time Warping Algorithm) and go the results with an accuracy of 72.6%.The vector representation of the features extracted can be used to compare with other respective vectors using the similarity measurement algorithms like edit distance, Twersky feature-based similarity measurements, Ukkonen distance, sum common, cosine similarity measurements. A study by (De Prisco et al.; 2017b) used the fuzzy vectorial method to identify the similarity in music and used the above distance measurements to calculate the similarity measurement and was able to achieve 93% accuracy.

Regarding the calculation of the similarity between the melodies in music, another study by (Schuitemaker; 2020) used an edit distance algorithm to calculate. A comparison is made between the song and its remix. Extracting the melody is not an easy task even though it is expected there is no guarantee that the accuracy it achieves because the polyphonic music contains(Müllensiefen and Frieler; 2006) multiple tracks that are randomly aligned in the channels.The rhythmic similarity calculation is ignored in this case which can be considered a factor for plagiarism and the results were obtained using Edit distances. This study placed the boundary for plagiarism to be the similarity value they got for the song and its remix. and this study lacks in finding the right threshold value.

2.2 Review on the Feature extraction

Feature extraction is a tedious task that requires understanding the structure of the data and applying appropriate functions for the extraction to happen A study by(Velankar and Kulkarni; 2018) uses feature engineering to extract the melodies from the data to extract similarity values to find the plagiarism in the music.

ChordGAN is a generative adversarial network that transfers music genre style characteristics.(Lu and Dubnov; n.d.). By incorporating chroma feature extraction into the training process, ChordGAN aims to learn how to convert harmonic structures into sounds. The chroma representation approximates chord notation in notated music since it only considers the pitch class of musical notes, portraying many notes as a density of pitches within a short period. Pop, jazz, and classical datasets were employed in the study for training and transfer. Two measures were employed to assess the transfer's success. After the classification, they were able to get an accuracy of 68% for pop, 74% for jazz, and 64% for classical.

MIT developed a toolkit called as Music21 package for python to understand and extract features from MIDI files. The study did by the paper on the music 21 (Cuthbert et al.; 2011) The feature capabilities of music21, an open-source toolkit for analyzing, finding, and manipulating symbolic music data, are described in this article. The music21 features module incorporates typical feature-extraction tools offered by other toolkits, adds new tools, and lets researchers easily create new and powerful extraction algorithms. These enhancements take advantage of the system's built-in capabilities to read diverse data formats and alter complicated scores (for example, by reducing them to a sequence of chords, automatically identifying key or metrical strength, or incorporating audio data).

A study used melodic similarities to build a method for detecting plagiarism in music(Lee et al.; 2011). There are five major blocks in these stages. Extraction of Melody: the melody of the audio is extracted. Pitch candidate estimate and pitch sequence identification are the two phases in melody extraction. The harmonic structure model is used to determine the pitch candidates. A technique used by (Dittmar et al.; 2012) describes a toolbox that was created to make the investigation of potential music plagiarism cases easier. The basic idea is to use techniques from Music Information Retrieval to inspect original and suspicious songs semi-automatically. Plagiarism in music is broken down into its most basic forms. Several signal processing techniques are presented that can be used to reveal these categories. They're meant to be used under the guidance of a human expert. Evaluation of the techniques is omitted in this paper. a tool set designed to make the examination of possible cases of music copying easier. The fundamental concept is to examine original and suspect music semi-automatically using techniques from Music Information Retrieval.

Plagiarism in music may be divided into several categories. Several signal processing approaches that can be utilized to expose these categories are described. They should only be utilized under the supervision of a human expert. In this work, the approaches are not evaluated. The various types of music plagiarism, such as Sample plagiarism, Melody plagiarism, and Rhythmic Plagiarism, are discussed in-depth, as well as the respective inspection techniques, such as the Brute Force approach and Decomposition approach, which fall under the category of Sample similarity inspection. Rhythmic source separation, Tempo Alignment (which is inspected for Rhythmic similarity and pitch vector similarity), and Sequence alignment (which is inspected for Melody similarity) are all covered in depth.

2.3 Review on the existing Models For text classification

One of the most essential and common tasks in supervised machine learning is text categorization. Assigning categories to documents, which can include anything from a web page to a library book to media pieces to a gallery, has a variety of uses, including spam screening, email routing, sentiment analysis, and more. Here it is shown how to conduct text categorization with Python, Scikit-Learn, and a little bit of NLTK in this section. Mostly the text classification is done by classifier machine learning algorithms like SVM, Naive Bayes, KNN, (Razno; 2019)Logistic Regression, etc. A study by (Shah et al.; 2020) conducts a comparison of the performance of the Machine Learning models. (Wahdan et al.; 2020) As classification methods, we used logistic regression, random forest, and K-nearest neighbor. Then these classifiers were put to the test, analyzed, and compared to one another, and a result was reached. On the basis of algorithms evaluated on the data set, the experimental conclusion demonstrates that the BBC news text categorization model produces satisfactory results. The authors choose to compare the results using five criteria: precision, accuracy, F1-score, support, and confusion matrix. The best machine learning method for the BBC news data set is the classifier that achieves the greatest score across all of these factors. Logistic Regression scored 96% Random forest scored 94%, KNN scored 97%.

Generally, Naive Bayes doesn't work well with small datasets in one of the studies by(Chen et al.; 2019), they had to provide a correlation value of 0.1 to achieve better accuracy with a smaller dataset. Word2vec as mentioned in the previous section is a model that is a part of NLP which is used to find the relationship between similar words. Before that, it should be converted to a bag of vectors to process. In (Chen et al.; 2020) They predicted a total of 658 samples from a test set of 921 news texts, giving us an accuracy of 71.4 percent. To categorize Lao news texts, a KNN-based Chinese and Lao text classification technique is presented, which employs data normalization and data dimensionality reduction. The approach has produced good results, according to the findings of the experiments.

Decision tree classifiers are widely considered as one of the most well-known ways for representing data classification in classifiers. The challenge of expanding a decision tree using existing data has been studied by researchers from many areas and backgrounds, including machine learning, pattern recognition, and statistics. Decision tree classifiers have been suggested in a variety of domains, including medical illness analysis, text categorization, user smartphone classification, pictures, and many more. The article (Charbuty and Abdulazeez; 2021) takes a more in-depth look into decision trees. Furthermore, the contents of the study, such as the algorithms/approaches utilized, datasets, and outcomes attained, are thoroughly examined and explained. Furthermore, all of the techniques examined were explained to demonstrate the authors' themes and determine the most accurate classifiers. As a consequence, the applications of various types of datasets are addressed, and their results are examined. Finally, the best accuracy achieved for the decision tree algorithm was 99.93%.

3 Methodology

3.1 Introduction

Although the motivation for the research is to bring value to businesses related to plagiarism in music, it is not directly linked to any business, so a modified KDD approach is used here. Data mining-related research is usually done in KDD or CRISP-DM methodologies, but in this scenario, KDD fits best because the deployment of models in the business layer is not applicable here, and it is not directly linked to any business.

3.2 Plagiarism in Music Methodology Approach

The usual data mining methods to prepare models or to perform other exploratory analyses we use are KDD and Crisp-DM. In line with the objectives, certain modifications are made other than the conventional way, those include 1) Understanding certain aspects of the music which will help to identify the required features.2) Extract those features 3) The extracted features transformed to vectors which are suitable to be further used for training 4) The converted vectors are then passed to the different machine learning algorithms like Gaussian Naive Bayes, Logistic Regression, Decision Tree Algorithms, Random Forest classifier, and KNN for training. 5) The trained models are then evaluated based on their confusion matrices.

3.3 Business Understanding

With the advent of the Internet many industries, especially in the field of art and entertainment, have the platform to show their work publicly with just one click. So it is critical to avoid redundancy among the uploaded content especially in the field of music. If not handled properly it will be a waste of space for better content yet to be used. Now Music industry is considered to be a place billion fortune business there is a high chance for plagiarism. So by creating these models there is an added advantage for composers to identify whether the compositions are plagiarized or not.

3.4 Dataset

More than 450 Sonic MIDIs were downloaded. The source site offers a variety of fan-created versions of the same music, some are expanded with additional sections, while others are more close to the basic composition. Nonetheless, they should all have the same structure and essential elements. The files[2][3][4] were extracted using Web Scraping, a technique used to extract files from a web-page directly with the help of a package in python called Beautiful soup.

3.5 Data Transformation

Any data mining or machine learning process begins with data pre-processing. To make the data for implementing the models, it must be processed and changed. Here the data-set contains MIDI(Musical Instrument Digital Interface) files that contain useful information regarding the structure of the music. The MIDI can be viewed using software called MuseScore which shows the melody of the music and the list of the instruments and all the required information for building the model. Here a technique called harmonic reduction is used to extract the main feature which is the melody or the chord progression of each MIDI file. And those features are used to construct a Dataframe that will be used to train the developed models. All the processes related to the MIDI files are done with the help of a Package in Python called the Music21 developed by MIT, and the general process of retrieving Information about music is called MIR(Music Information Retrieval).

3.6 Project Flow

Data modeling occurs after the data has been processed and transformed to the necessary format.The models are built, trained, tested, and assessed at this phase.

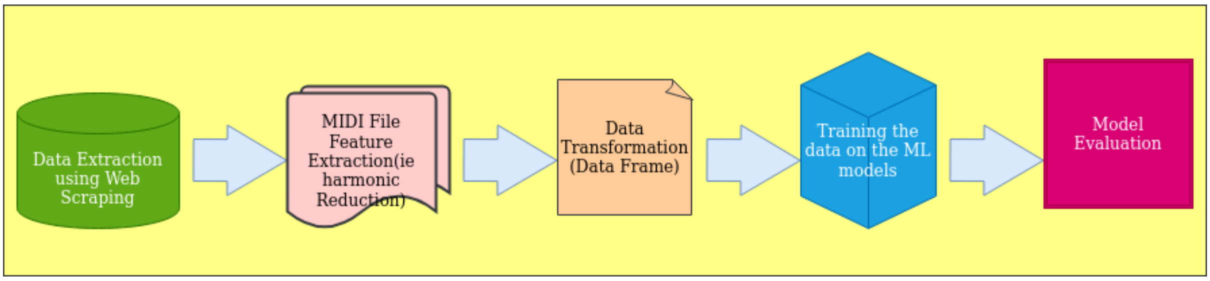

Data Extraction using Web Scraping
MIDI File Feature Extraction(ie harmonic Reduction)
Data Transformation (Data Frame)
Training the data on the ML models
Model Evaluation

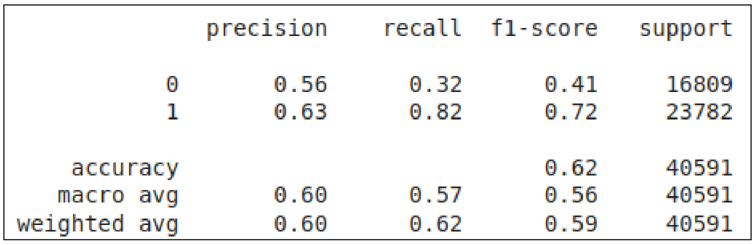
Figure 1: The flow indicating methods that is used to identify plagiarism in music

[2] "https://files.khinsider.com/midifiles/genesis/sonic-the-hedgehog"
[3] "https://files.khinsider.com/midifiles/genesis/sonic-the-hedgehog-2"
[4] "https://files.khinsider.com/midifiles/genesis/sonic-the-hedgehog-3

The pre-processed dataset contains the chord progression in the text format using the Word2Vec model these text-based chords are converted to vectors and it is further converted to the bag of words using NLP to be used to train using the different machine learning algorithms such as Gaussian Naive Bayes, KNN, Random Forest, Logistic Regression and Decision tree classifiers which are included in the python Sklearn package.1The above Architecture diagram shows the flow of the process.

3.7 Evaluation Metrics

To evaluate the accuracy, recall, precision, sensitivity, specificity, and F-1 measure, of the models in terms of classification and prediction.A confusion matrix is an important part of assessing a model since it allows you to accuracy, and recall.A sample Figure 2 is shown below.



```
              precision    recall  f1-score   support

           0       0.56      0.32      0.41     16809
           1       0.63      0.82      0.72     23782

    accuracy                           0.62     40591
   macro avg       0.60      0.57      0.56     40591
weighted avg       0.60      0.62      0.59     40591
```

Figure 2: Sample Confusion matrix

4 Design Specification,Implementation, Evaluation and Results of Plagirism in Music Classification Models

4.1 Design Specification

The interpretations from the classification models and exploratory analysis of the data are represented in utilizing MUSIC 21, a python library established by MIT, in the project design process of detecting plagiarism in music. Data selection, feature extraction, transformation, and training of classification models are all performed in the business logic tier, followed by model evaluation. It is a two-tier architecture Diagram as shown in Figure 3 where the visualization is done in the upper tier in the diagram and the rest of the back end process happens in the low-end tier.

4.2 Extraction of features from the data set

Here, the main task is to prepare the data for training our developed models, for this to be done, important features of the files are to be visualized to select the appropriate one for the model. Since there are a bunch of files to extract, it will be wise to see what all contents exist in a MIDI file, to do that start with analyzing one MIDI file. First, the number of instruments used to create the melody is known. As shown in Figure4

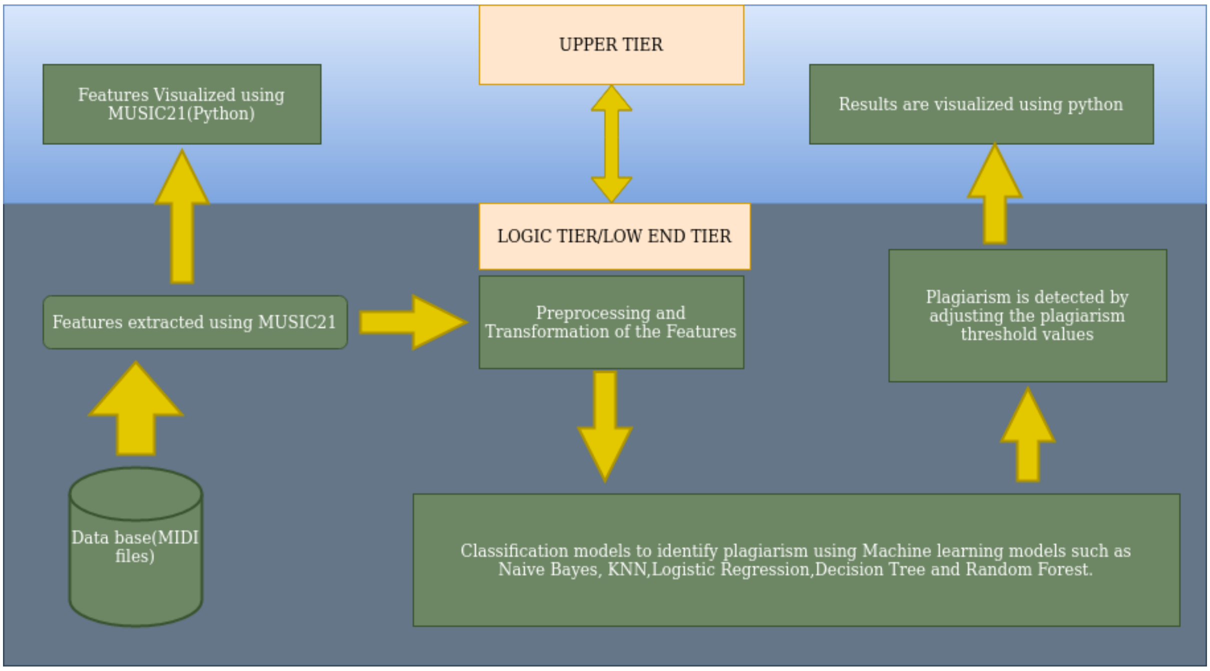

UPPER TIER
Features Visualized using MUSIC21(Python)
Results are visualized using python
LOGIC TIER/LOW END TIER
Features extracted using MUSIC21
Preprocessing and Transformation of the Features
Plagiarism is detected by adjusting the plagiarism threshold values
Data base(MIDI files)
Classification models to identify plagiarism using Machine learning models such as Naive Bayes, KNN,Logistic Regression,Decision Tree and Random Forest.

Figure 3: The flow indicating methods that is used to identify plagiarism in music

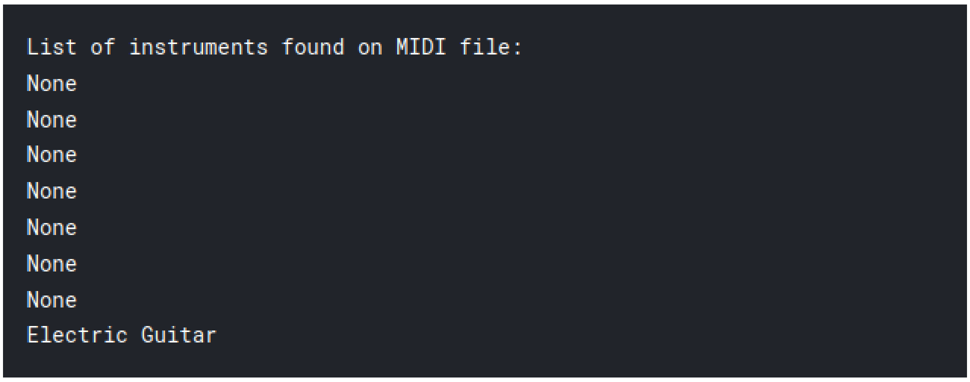
Then it is made sure the in the instrument list no percussion instruments are used because here the main aim is to extract the melody to confirm that the extracted one is the melody it is wise to delete the channel in the MIDI file which consist of the drum track which is usually assumed to be 10th track in the MIDI file.

The Figure5 is a depiction of how the notes exist in the MIDI file and each color in the plot represents each instrument. It is quite difficult for people who are not exposed to the music theory to understand the notes that are viewed using software like MuseScore because MIDIs are often created as LMMS on Digital Audio Workstations to produce musical audio. The presentation may be rather unpleasant in this manner.

If the original song is known, some known sections, like the first arpeggio at the beginning and the melody commencing after it, can be identified. The structure will be apparent with more measures to plot. However, extracting pitch information from it is difficult.

Look at Figure6 pitch histogram to discover which notes are being played the most. If you know a little music theory, you'll see that the seven more notes are all part of the C-major/A-minor Key, thus this is a decent method to figure out what key the song is in. The scatter plot7 demonstrates that the arrangement of notes appears to be stable throughout time, indicating that there are no significant changes in this work.

It would be useful to obtain the song's harmonic sequence to understand how it was constructed and compare it to other music. A reduction in music is a simplified arrangement or transcription of an existing score or work to make analysis, performance, or practice easier or clearer. The number of parts may be decreased, or the rhythm may be simplified, such as by the use of block chords. Begin by combining all voices into one and picturing the contents. Then combine all instruments such that each measure is saturated with chords. As previously stated, each measure comprises four beats, thus only the most often used chord for each measure is presented. Find the chord by counting the four most often used notes in each measure and making a chord out of it. For example,



```
List of instruments found on MIDI file:
None
None
None
None
None
None
None
Electric Guitar
```

Figure 4: The list of instruments in the MIDI File

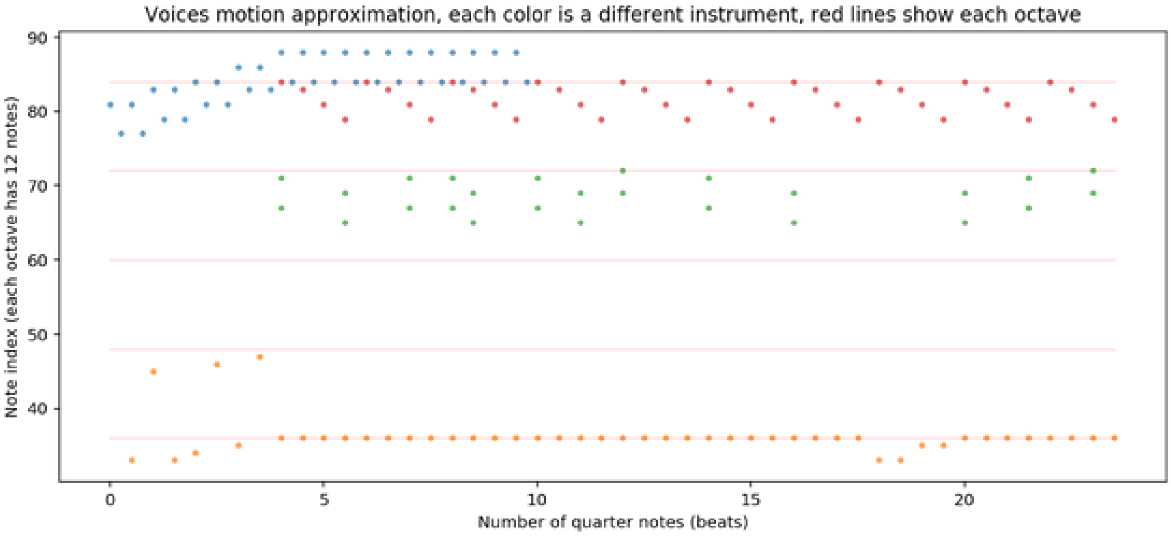


Figure 5: The Structure of the instruments placed in the MIDI file

if we discover (E4, G4, C5, G5), we may deduce that it is a C-major chord.

As shown in Figure8. Before converting the chords to strings a process called chord substitution is done it. chord substitution is a method of replacing a chord progression with another appropriate chord for this task a model called as Word2Vec is created it is an NLP tool used to identify the relationships between the chord progression and come up with a relationship value and the replacement character to process these chords in the model there is a necessity to convert them into vectors which are set of NumPy arrays. After the chord substitution is done for all the MIDI files these are saved in a Data Frame for further implementation. This sums the steps in feature extraction. The figure9 shows the harmonic reduction output of a single MIDI file. And the Figure shows the output of the Extracted Data of all the MIDI files in a Data Frame.

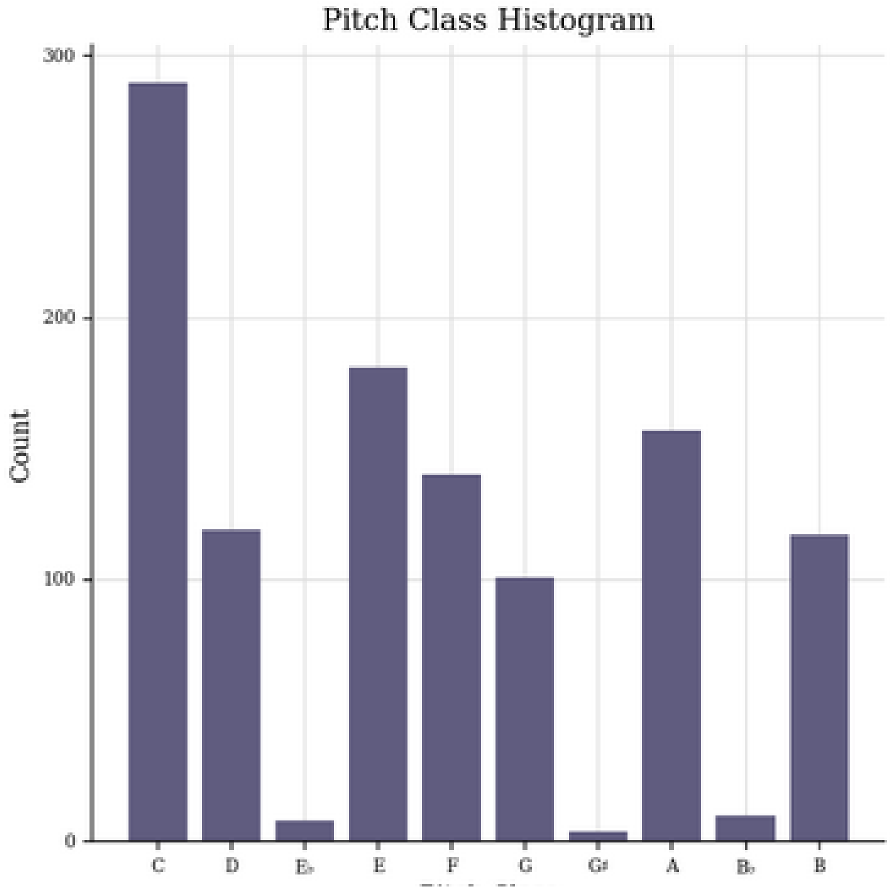


Figure 6: The Pitch histogram

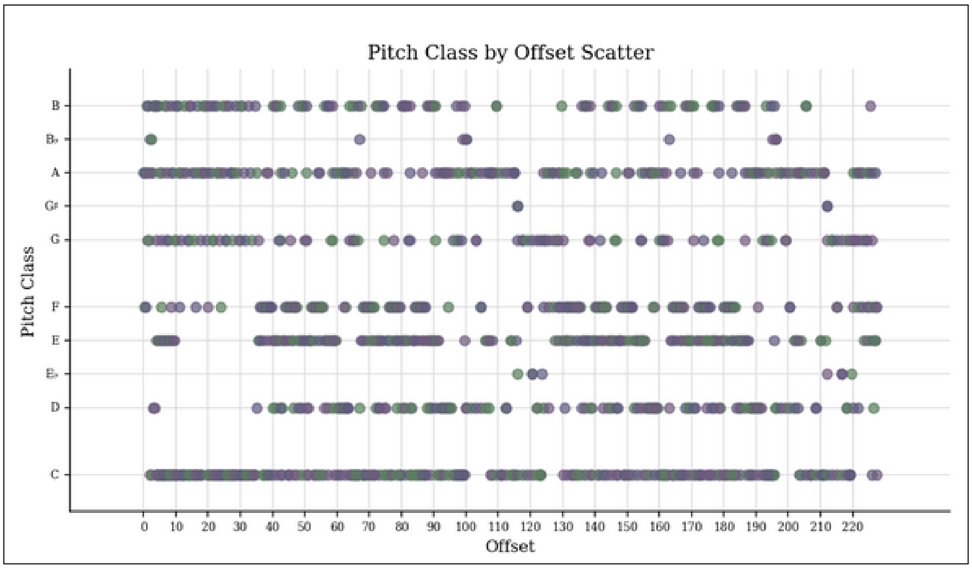
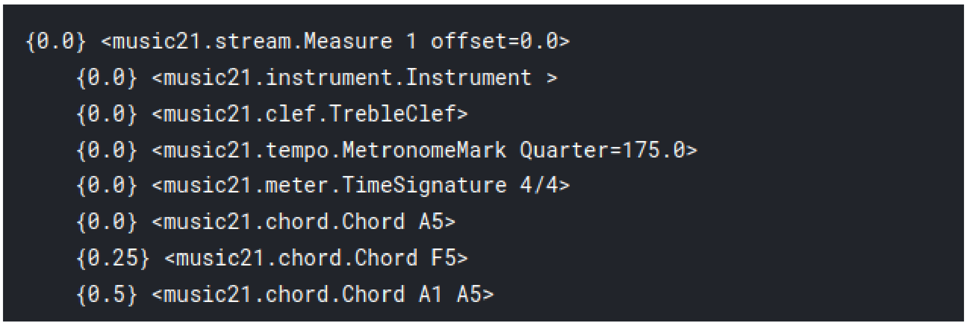
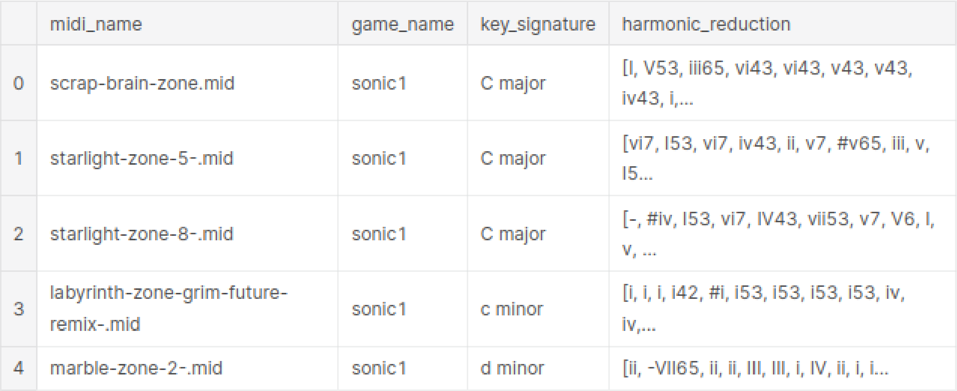


Figure 7: Pitch class scatter plot



```
{0.0} <music21.stream.Measure 1 offset=0.0>
    {0.0} <music21.instrument.Instrument >
    {0.0} <music21.clef.TrebleClef>
    {0.0} <music21.tempo.MetronomeMark Quarter=175.0>
    {0.0} <music21.meter.TimeSignature 4/4>
    {0.0} <music21.chord.Chord A5>
    {0.25} <music21.chord.Chord F5>
    {0.5} <music21.chord.Chord A1 A5>
```

Figure 8: The Chords in the file



	midi_name	game_name	key_signature	harmonic_reduction
0	scrap-brain-zone.mid	sonic1	C major	[I, V53, iii65, vi43, vi43, v43, v43, iv43, i,...
1	starlight-zone-5-.mid	sonic1	C major	[vi7, I53, vi7, iv43, ii, v7, #v65, iii, v, I5...
2	starlight-zone-8-.mid	sonic1	C major	[-, #iv, I53, vi7, IV43, vii53, v7, V6, I, v, ...
3	labyrinth-zone-grim-future-remix-.mid	sonic1	c minor	[i, i, i, i42, #i, i53, i53, i53, i53, iv, iv,...
4	marble-zone-2-.mid	sonic1	d minor	[ii, -VII65, ii, ii, III, III, i, IV, ii, i, i...

Figure 9: The pre-processed data frame

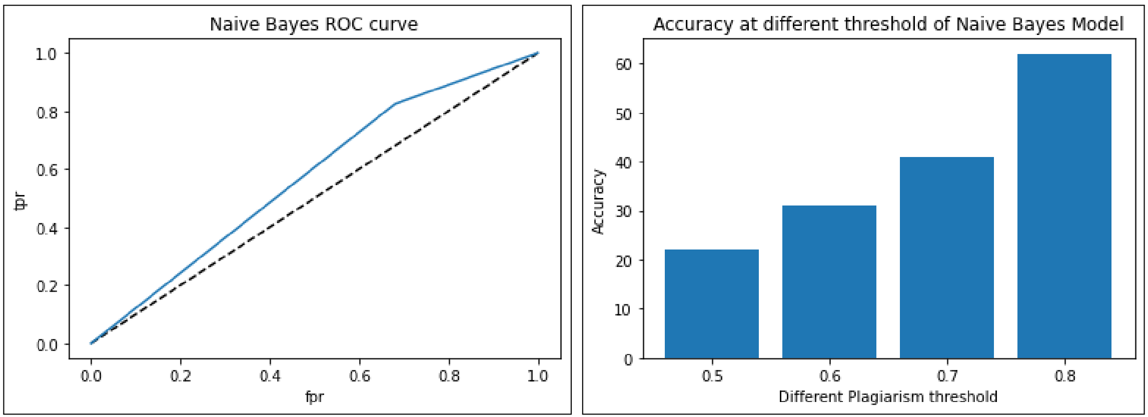
4.3 Implementation,Evaluation and Results of Naıve Bayes

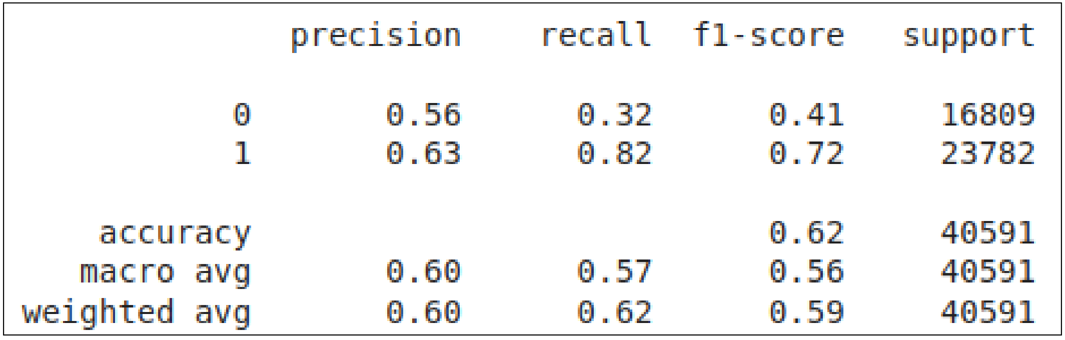
4.3.1 Implementation

Because the features are all continuous, we utilize Gaussian Naive Bayes instead of traditional Naive Bayes, which is an appropriate model to employ when categorical values are present in independent data. Naive Bayes is carried out by dividing the data into training and testing datasets in a 4:1 ratio, i.e. 80 percent of train data and 20% of test data. Gaussian Naıve Bayes is implemented using the sklearn library in python. The function used to implement is GaussianNB(). It is trained on the training data. The data that will be supplied to train will be the bag of words that are converted during the pre-processing stage and it will be classified based on the category where the similarity between two music, which is greater than 0.8 will be considered as plagiarised. To check the accuracy of the different plagiarism threshold levels where used (0.8,0.7,0.6,0.5) The vectors will be classified against the y variable that is duplicate in the data frame which has values 0 and 1. 0: not plagiarised and 1: plagiarised.

4.3.2 Evaluation and Results

Gaussian Naive Bayes was trained for 4 different threshold values(0.8,0.7,0.6,0.5) on the dataset and show more accuracy for the threshold value 0.8 with an accuracy of 62%. which means when the threshold for identifying similarity in melodies between the music was higher it was able to classify whether the music is plagiarised or not. As the threshold values have decreased the accuracy of the prediction is reduced. This means the model performed best at classification when the threshold was set as 0.8. the Trend in the accuracy is shown in the below plot.Check Figure10 for Plots and confusion matrix





```
              precision    recall  f1-score   support

           0       0.56      0.32      0.41     16809
           1       0.63      0.82      0.72     23782

    accuracy                           0.62     40591
   macro avg       0.60      0.57      0.56     40591
weighted avg       0.60      0.62      0.59     40591
```

Figure 10: Results of Naive Bayes

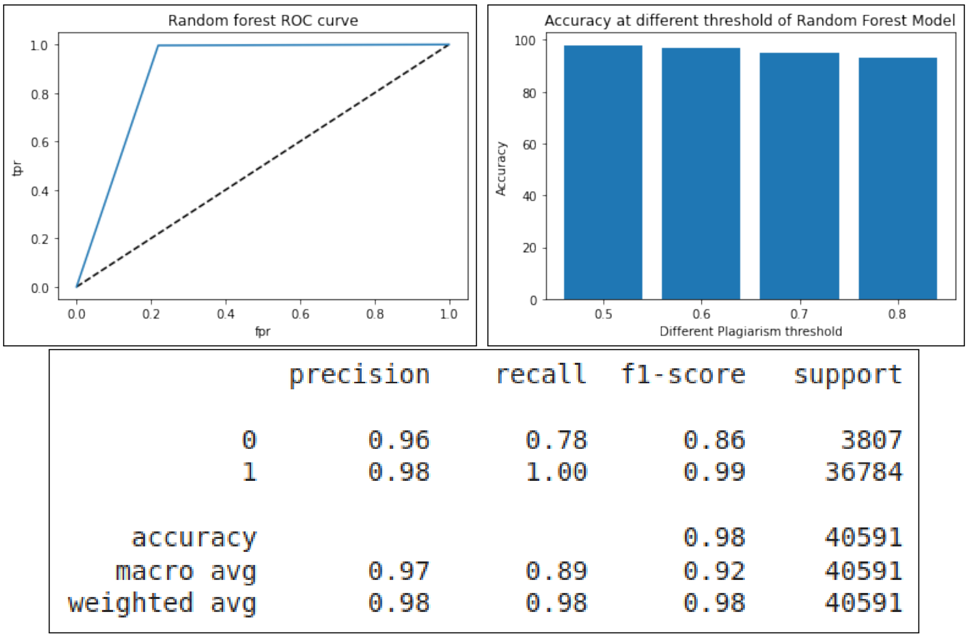
4.4 Implementation,Evaluation and Results of Random Forest

4.4.1 Implementation

A random forest is a meta estimator that utilizes averaging to enhance prediction accuracy and control over-fitting by fitting several decision tree classifiers on different sub-samples of the dataset. Random Forest is a collection of tree classification algorithms. The model is run by dividing the data into training and testing datasets in a 4:1 ratio, resulting in 80 percent train data and 20% test data. Random Forest is built in Python using the sklearn package. Random Forest Classifier() is the function that was used to implement it. The training data is used to train it. This model is created with a variety of feature combinations. The data that will be supplied to train will be the bag of words that are converted during the pre-processing stage and it will be classified based on the category where the similarity between two music, which is greater than 0.8 will be considered as plagiarised. To check the accuracy of the different plagiarism threshold levels where used (0.8,0.7,0.6,0.5) The vectors will be classified against the y variable that is duplicate in the data frame which has values 0 and 1. 0: not plagiarised and 1: plagiarised.

4.4.2 Evaluation and Results

Random Forest classifier was trained for 4 different threshold values(0.8,0.7,0.6,0.5) on the dataset and show more accuracy for the threshold value 0.5 with an accuracy of 98%. which means when the threshold for identifying similarity in melodies between the music was higher it was able to classify whether the music is plagiarised or not. As the threshold value increased the accuracy of the prediction reduced.But the decrease in the accuracy is no that major because it performed with 93% accuracy at the threshold level of 0.8 This means the model performed best at classification when the threshold was set as 0.8. the Trend in the accuracy is shown in Figure 11.



```
              precision    recall  f1-score   support

           0       0.96      0.78      0.86      3807
           1       0.98      1.00      0.99     36784

    accuracy                           0.98     40591
   macro avg       0.97      0.89      0.92     40591
weighted avg       0.98      0.98      0.98     40591
```

Figure 11: Results of Random Forest Classifier

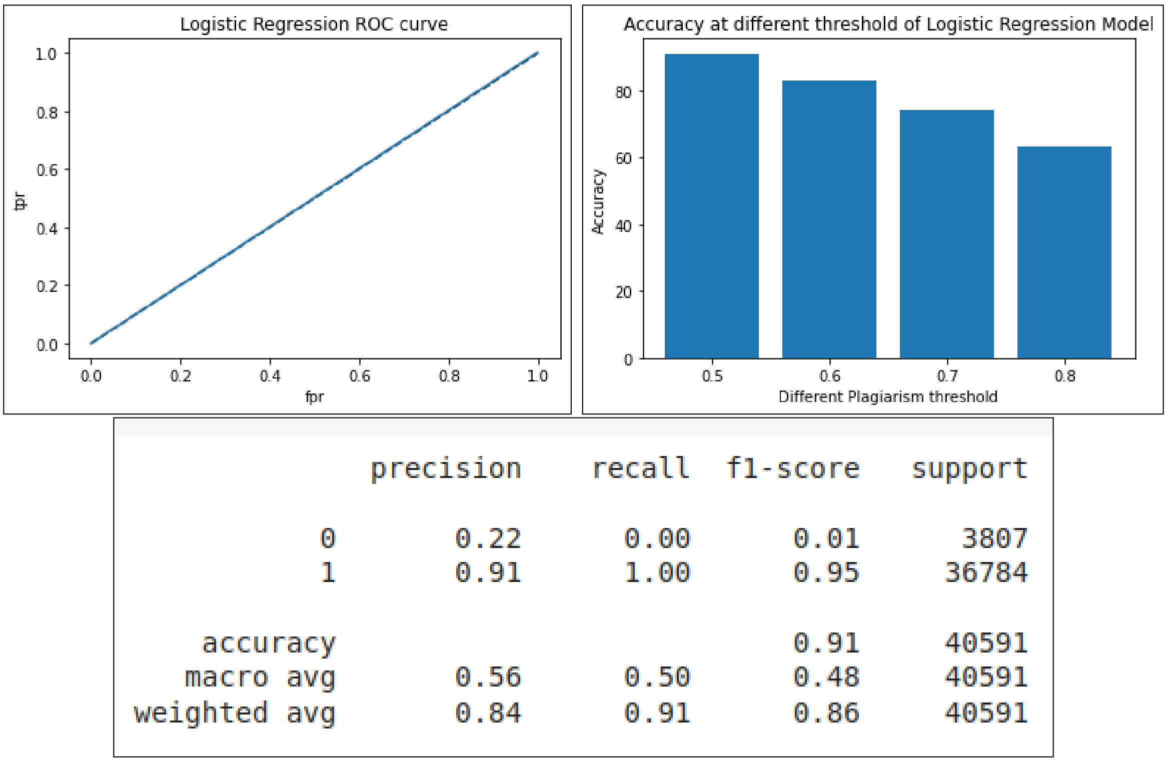
4.5 Implementation,Evaluation and Results of Logistic Regression

4.5.1 Implementation

If the ‘multi class’ option is set to OVR, the training algorithm utilizes the one-vs-rest (OvR) scheme, and if the ‘multi class’ option is set to multinomial, the training method employs the cross-entropy loss. It can work with both dense and sparse data. The model is implemented by splitting the data into training and testing datasets in a 4:1 ratio, yielding 80 percent train data and 20% test data. The sklearn package in Python is used to implement logistic regression. The function used to implement the pre-processing step is LogisticRegression(), and it will be categorized based on the category where similarity of higher than 0.8 between two pieces of music will be deemed plagiarized. The vectors will be categorized against the y variable that is duplicate in the data frame and has values 0 and 1. 0: not plagiarized, 1: plagiarized.

4.5.2 Evaluation and Results

Logistic Regression was trained for 4 different threshold values(0.8,0.7,0.6,0.5) on the dataset and show more accuracy for the threshold value 0.5 with an accuracy of 91%. which means when the threshold for identifying similarity in melodies between the music was higher it was able to classify whether the music is plagiarised or not. As the threshold values increased the accuracy of the prediction reduced. This means the model performed best at classification when the threshold was set as 0.5. the Trend in the accuracy is shown in the below plot. A classification matrix is shown in Figure12.



```
              precision    recall  f1-score   support

           0       0.22      0.00      0.01      3807
           1       0.91      1.00      0.95     36784

    accuracy                           0.91     40591
   macro avg       0.56      0.50      0.48     40591
weighted avg       0.84      0.91      0.86     40591
```

Figure 12: Results of Logistic Regression

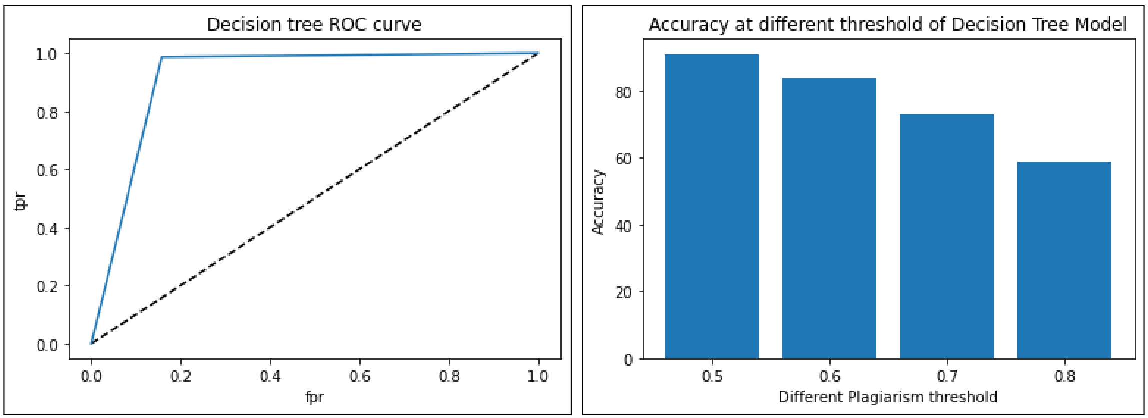
4.6 Implementation,Evaluation and Results of Decision Tree

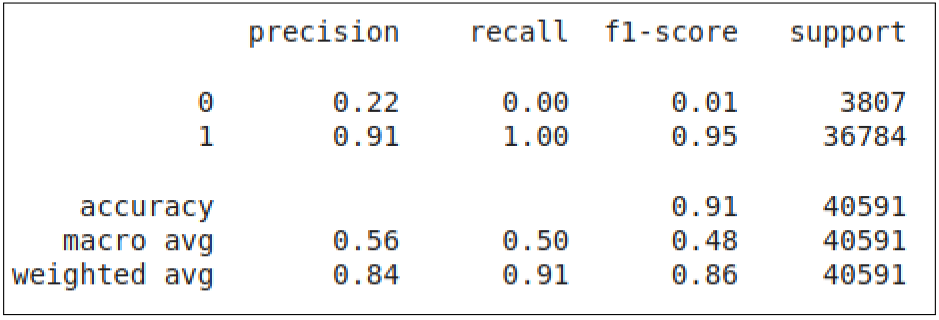
4.6.1 Implementation

For classification and regression, Decision Trees are a non-parametric supervised learning approach. The objective is to construct a model that predicts the value of a target variable by learning basic decision rules inferred from the data characteristics. A tree is an approximation of a piecewise constant. Decision Tree is carried out by dividing the data into training and testing datasets in a 4:1 ratio, i.e. 80 percent of train data and 20% of test data. Python's sklearn package is used to construct the Decision Tree classifier.Decisiontreeclassifier() is the function used to implement it ().It is trained using the practice data. The data provided to train will be a bag of words transformed during the pre-processing step, and it will be categorized based on the category where the similarity between two pieces of music is more than 0.8 will be regarded as plagiarised. To test the accuracy of the various plagiarism threshold levels (0.8,0.7,0.6,0.5), the vectors will be categorized against the y variable that is duplicate in the data frame, which has values 0 and 1. 0: not plagiarised, 1: plagiarised.

4.6.2 Evaluation and Results

Decision Tree was trained for 4 different threshold values(0.8,0.7,0.6,0.5) on the dataset and show more accuracy for the threshold value 0.5 with an accuracy of 91%. which means when the threshold for identifying similarity in melodies between the music was higher it was able to classify whether the music is plagiarised or not. As the threshold values have increased the accuracy of the prediction reduced to 59% for value 0.8. This means the model performed best at classification when the threshold was set as 0.5. the Trend in the accuracy is shown in Figure 13.





```
              precision    recall  f1-score   support

           0       0.22      0.00      0.01      3807
           1       0.91      1.00      0.95     36784

    accuracy                           0.91     40591
   macro avg       0.56      0.50      0.48     40591
weighted avg       0.84      0.91      0.86     40591
```

Figure 13: Results of Decision tree

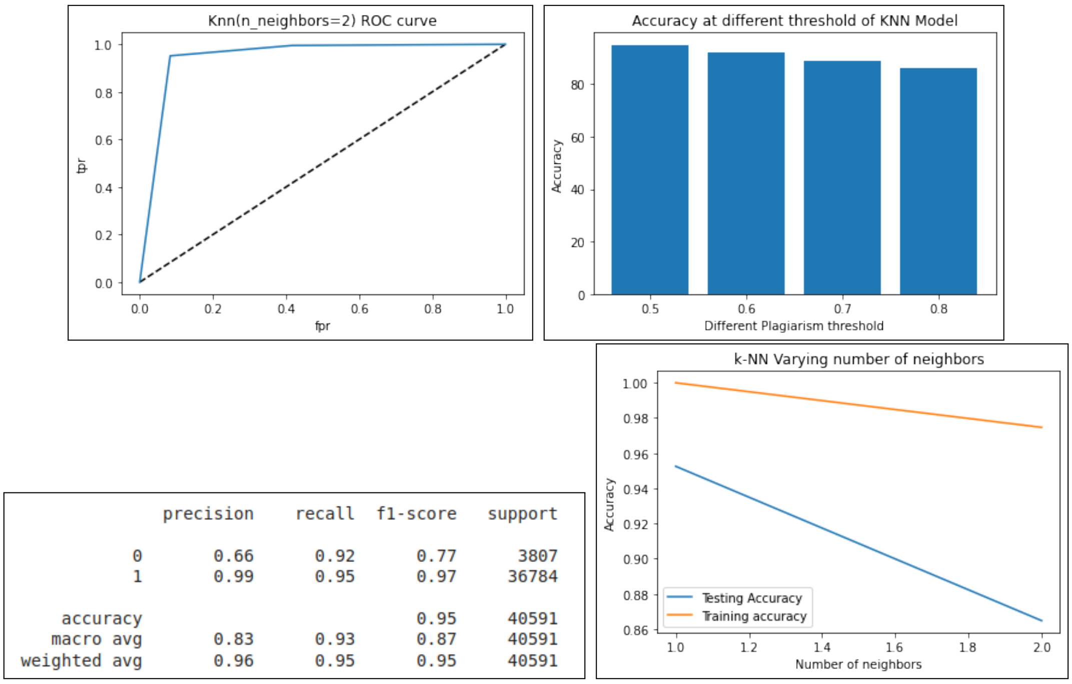
4.7 Implementation,Evaluation and Results of KNN

4.7.1 Implementation

K nearest neighbors is a straightforward method that saves all existing instances and categorizes new ones using a similarity metric. KNN has been utilized as a non-parametric approach in statistical estimates and pattern recognition since the early 1970s. KNN is carried out by dividing the data into training and testing datasets in a 4:1 ratio, i.e. 80 percent of train data and 20% of test data. KNN is implemented using the Sklearn library in python. The function used to implement is KNeighboursClassifier(). It is trained on the training data. The data that will be supplied to train will be the bag of words that are converted during the pre-processing stage and it will be classified based on the category where the similarity between two music, which is greater than 0.8 will be considered as plagiarised. To check the accuracy of the different plagiarism threshold levels where used (0.8,0.7,0.6,0.5) The vectors will be classified against the y variable that is duplicate in the data frame which has values 0 and 1. 0: not plagiarised and 1: plagiarised.

4.7.2 Evaluation and Results

KNN was trained for 4 different threshold values(0.8,0.7,0.6,0.5) on the dataset and show more accuracy for the threshold value 0.5 with an accuracy of 95%. which means when the threshold for identifying similarity in melodies between the music was higher it was able to classify whether the music is plagiarised or not. As the threshold values have increased the accuracy of the prediction reduced to 0.86 at 0.8 even though the trend in the reduction is not that significant. This means the model performed best at classification when the threshold was set as 0.8. the Trend in the accuracy is shown in the below plot. And the ROC plot is also shown in the Figure 14.



```
              precision    recall  f1-score   support

           0       0.66      0.92      0.77      3807
           1       0.99      0.95      0.97     36784

    accuracy                           0.95     40591
   macro avg       0.83      0.93      0.87     40591
weighted avg       0.96      0.95      0.95     40591
```

Figure 14: Results of KNN for k = 2

4.8 Comparison of the Developed models

From the above results, we can see the trend in the performance of each model for their respective threshold values below plot shows the highest accuracy of each model at their best plagiarism threshold value. Different colors represent the different threshold values. From the figure, we can see that the Random Forest model performs best with the threshold value range with 98% at 0.5 and 93% at 0.8. As shown in the Figure 15

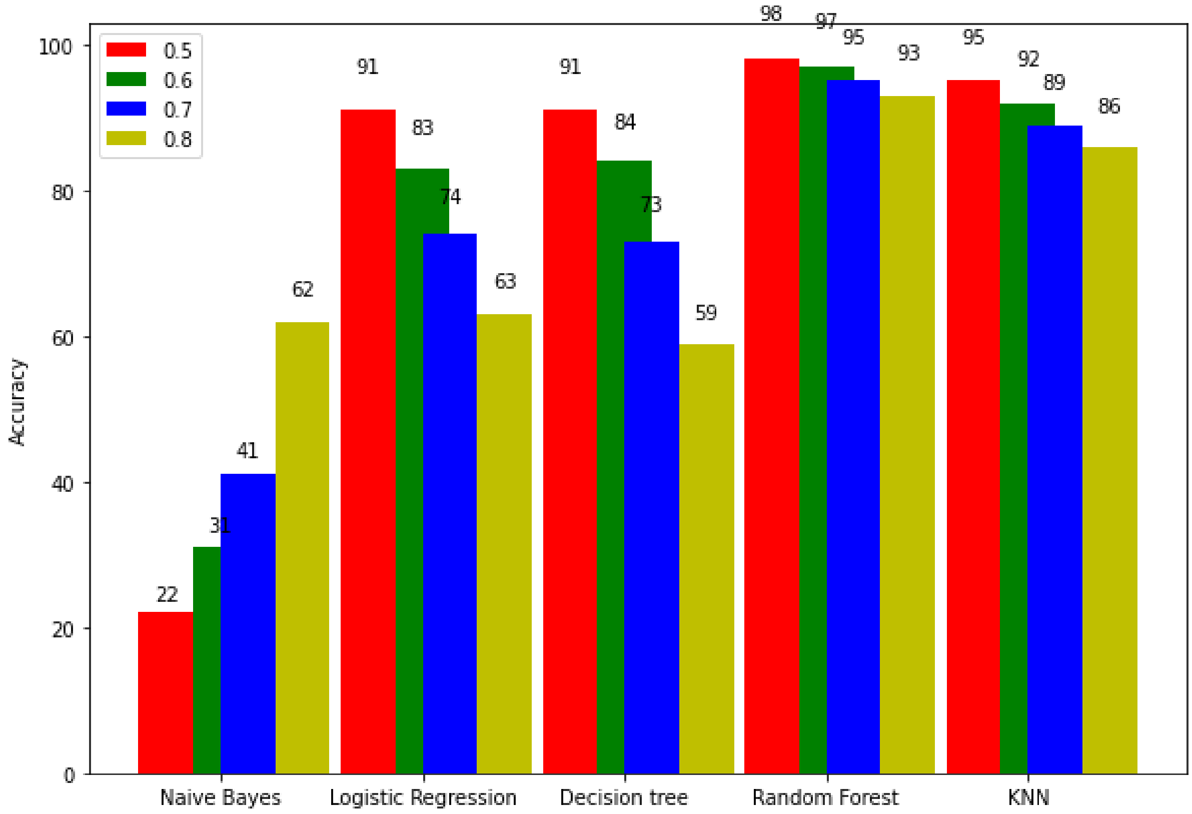


Figure 15: Comparison of the developed model based on their accuracy

4.9 Identification of the Plagiarism Threshold value in music

As shown in Figure 15 each model was trained for different plagiarism Threshold values(0.8,0.7,0.6,0.5) which was used to plot a bar chart to check whether which model at which plagiarism value performed best. Overall from the plot, we can see Random Forest classifier performed with the most accuracy, the rest of the models performed best at a plagiarism threshold value of 0.5 except Naive Bayes as Naive Bayes performs worst with a small dataset as mentioned in (Chen et al.; 2019). So we can say that for a dataset with chord progression which was analyzed by converting to text and, the text classification was performed, Random Forest performed with most accuracy at all threshold values. Since 3 out of 4 models performed best at a plagiarism threshold of 0.5 we can say that in this case, we can consider melodies that have got 0.5 and more similarity to other melodies can be considered as plagiarised.

5 Discussion,Conclusion and Future Work

The purpose of the research was to identify plagiarism in music using Machine Learning Algorithms and identify the boundary value that decides, at what threshold value plagiarism in music can be suspected .all the research objectives and sub-objectives were met. To solve the research question first, feature extraction was performed on the dataset that was used to extract the required data to be trained on the different models. Then the preprocessed data was fed to the models such as Gaussian Naive Bayes, Logistic Regression, Random Forest Classifier, Decision Tree, and KNN to know which model performed best. The results obtained were depicted in the form of a confusion matrix, Respective bar plots, and ROC plots. The models were trained at different plagiarism threshold values and to determine the boundary value and we concluded that 0.5 can be considered as a plagiarism threshold because 3 out of 4 models detected plagiarism threshold values at 0.5 itself. and there were able to predict and classify with the most accuracy. Among the models, the Random Forest classifier performed best at all the threshold values with an accuracy of 98% accuracy and Naive Bayes performed poorly at 0.5 value with an accuracy of 22% .

In the future, the main focus of the research will extend the idea to a multi-tracked midi file that can detect the melodies in the MIDI file with higher accuracy so that with higher accuracy more accurate threshold values can be computed. And this research can be extended to perform more data for higher accuracy.

Acknowledgement

I would like to thank my Research Supervisor Dr. Catherine Mulwa for the immense support and clear guidance that helped me to complete my research thesis with success and I would like to extend my gratitude to my parents for their immense blessings and support.

References

Charbuty, B. and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning, *Journal of Applied Science and Technology Trends* **2**(01): 20–28.

Chen, J., Dai, Z., Duan, J., Matzinger, H. and Popescu, I. (2019). Naive bayes with correlation factor for text classification problem, *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, IEEE, pp. 1051–1056.

Chen, Z., Zhou, L. J., Da Li, X., Zhang, J. N. and Huo, W. J. (2020). The lao text classification method based on knn, *Procedia Computer Science* **166**: 523–528.

Cuthbert, M. S., Ariza, C. and Friedland, L. (2011). Feature extraction and machine learning on symbolic music using the music21 toolkit., *Ismir*, pp. 387–392.

De Prisco, R., Malandrino, D., Zaccagnino, G. and Zaccagnino, R. (2017a). A computational intelligence text-based detection system of music plagiarism, *2017 4th International Conference on Systems and Informatics (ICSAI)*, IEEE, pp. 519–524.

De Prisco, R., Malandrino, D., Zaccagnino, G. and Zaccagnino, R. (2017b). Fuzzy vectorial-based similarity detection of music plagiarism, *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, IEEE, pp. 1–6.

De, S., Roy, I., Prabhakar, T., Suneja, K., Chaudhuri, S., Singh, R. and Raj, B. (2015). Plagiarism detection in polyphonic music using monaural signal separation, *arXiv preprint arXiv:1503.00022* .

Dittmar, C., Hildebrand, K. F., Gärtner, D., Winges, M., Müller, F. and Aichroth, P. (2012). Audio forensics meets music information retrieval—a toolbox for inspection of music plagiarism, *2012 Proceedings of the 20th European signal processing conference (EUSIPCO)*, IEEE, pp. 1249–1253.

González-Carvajal, S. and Garrido-Merchán, E. C. (2020). Comparing bert against traditional machine learning text classification, *arXiv preprint arXiv:2005.13012* .

Kadhim, A. I. (2019). Survey on supervised machine learning techniques for automatic text classification, *Artificial Intelligence Review* **52**(1): 273–292.

Lee, J., Park, S., Jo, S. and Yoo, C. D. (2011). Music plagiarism detection system, *Proceedings of the 26th International Technical Conference on Circuits/Systems, Computers and Communications*, pp. 828–830.

Li, B. and Han, L. (2013). Distance weighted cosine similarity measure for text classification, *International conference on intelligent data engineering and automated learning*, Springer, pp. 611–618.

Lu, C. and Dubnov, S. (n.d.). Chordgan: Symbolic music style transfer with chroma feature extraction.

Müllensiefen, D. and Frieler, K. (2006). Evaluating different approaches to measuring the similarity of melodies, *Data Science and Classification*, Springer, pp. 299–306.

Orkphol, K. and Yang, W. (2019). Word sense disambiguation using cosine similarity collaborates with word2vec and wordnet, *Future Internet* **11**(5): 114.

Razno, M. (2019). Machine learning text classification model with nlp approach, *Computational Linguistics and Intelligent Systems* **2**: 71–73.

Schuitemaker, N. (2020). *An analysis of melodic plagiarism recognition using musical similarity algorithms*, B.S. thesis.

Shah, K., Patel, H., Sanghvi, D. and Shah, M. (2020). A comparative analysis of logistic regression, random forest and knn models for the text classification, *Augmented Human Research* **5**(1): 1–16.

Velankar, M. and Kulkarni, P. (2018). Feature engineering and generation for music audio data, *IJETT* **5**(1): 10012–10018.

Wahdan, K. A., Hantoobi, S., Salloum, S. A. and Shaalan, K. (2020). A systematic review of text classification research based ondeep learning models in arabic language, *Int. J. Electr. Comput. Eng* **10**(6): 6629–6643.