Voice Recognition Using Machine Learning And Central Database To Enhance Security System

by

Mir Md. Taosif Nur
18101392
Sumaiya Sultana Dola
19101674
Apurba Kishore Banik
18101483
Tanzeem Akhter
18101408
Nafees Hossain
18101106

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Mir Md. Taosif Nur
18101392

Dola

Sumaiya Sultana Dola
19101674

Apurba

Apurba Kishore Banik
18101483

Tanzeem Akhter

Tanzeem Akhter
18101408

Nafees Hossain

Nafees Hossain
18101106

Approval

The thesis/project titled **"Voice Recognition Using Machine Learning And Central Database To Enhance Security System"** submitted by

1. Mir Md. Taosif Nur (18101392)
2. Sumaiya Sultana Dola (19101674)
3. Apurba Kishore Banik (18101483)
4. Tanzeem Akhter (18101408)
5. Nafees Hossain (18101106)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January, 2022.

Examining Committee:

Primary Supervisor:
(Member)

Ms. Jannatun Noor Mukta
Lecturer
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD

Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Voice recognition refers to the purpose of interpreting voice or identifying any individual voice. In modern days of technological advancement voice recognition has been playing an integral part behind many machine learning algorithms. Furthermore, speech recognition, alternatively referred to as voice recognition, can help us immensely in particular scenarios such as in building better access control system and security system. Voice detection and comparison is a challenging problem because the traditional methods of speech recognition are not on par with human capabilities. In modern machine learning methodologies there is a vast potential to overcome barriers of detecting human speech. The voice is a simple medium people use for everyday communication, so it can be used to improve security system by utilizing voice recognition identifying an individual. This article focuses on enhancing security system by deep learning based approach of voice recognition. Moreover, the article further elaborates about using available datasets from a central database which is used for voice detection and comparison. The focal point of this article is to apply the most suitable methodologies of machine learning and deep learning to detect any individual by the prosodic feature of speech from a given central database.

Acknowledgement

Firstly, all praise to the Great Allah for whom our thesis have been completed without any major interruption.
Secondly, to our advisor Ms. Jannatun Noor Mukta ma'am for her kind support and advice in our work. She helped us whenever we needed help.
And finally to our parents without their throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASR	Automatic Speech Recognition
$AUC(AUROC)$	Area Under The Receiver Operating Characteristics
$CART$	Clarification And Regrssion Trees
CNN	Convolutional Neural Network
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DT	Decision Tree
EER	Energy Efficiency Ratio
FFT	Fast Fourier Transform
GB	Gradient Boosting
GMM	Gaussian Mixture Model
GTD	Generalised Gama Distribution
HMM	Hidden Makarov Model
KNN	K-Nearest Neighbor
LMT	Logistic Model Tree
LPC	Linear Prediction Co-efficients
LR	Logistic Regression
MCE	Minimum Classification Error
$MDLF$	Multi-Dimensional Local Features
$MFCC$	Mel-frequency cepstral coefficients

ML	Machine Learning
MLP	Multilayer Perceptron
NB	Naive Bayes
NN	Neural Network
$PLDA$	Probabilistic Linear Discriminant Analysis
PPG	Phoneme Posteriorgram
$RBFN$	Radial Basis Function Network
RF	Random Forest
SVM	Support Vector Machine
UBM	Universal Background Model
VAD	Voice Activity Detection

Chapter 1

Introduction

b As of late, an ever increasing number of applications are being created to improve interaction among people and machines. Vocal communication is a vital medium for human interaction. Due to the headway of innovation, machines are gradually being more precise to comprehend voice of human. People have adapted to the machines by using machine language while it would be more advantageous if people can cooperate with machines in a more normal manner [10]. However, voice recognition is a troublesome issue to address on the grounds that the voice of an individual may change because of different reasons. Therefore, robust voice recognition system is important in light of the fact that it tends to be utilized in numerous fields, such as security, authentication/verification, surveillance, and so on [19]. Despite the fact that voice comparison is a difficult issue for analysts, deep learning approaches which are machine learning techniques, have the capacity to give a suitable answer for the issue [19]. Exploration in the area of more natural interaction between machines and humans narrowed down on creating two frameworks, which are speech and voice recognition that helps a machine to either comprehend and perceive words or distinguish who is talking, respectively [10].Additionally on the case of the text being something similar for enlistment and recognition, then it is known as text-dependent identification. On the other hand, text-independent system employs different text for training and testing [19]. Furthermore,speaker identification is the way toward deciding an obscure speaker's personality by cross-matching with their voice to the voices in the information base of enrolled speakers and Speaker verification can decide if an individual is what he or she claims to be on the basis of voice sample from the dataset [19]. Making the AI applications that provide speech and voice recognition functionalities would help towards improving quality of life.

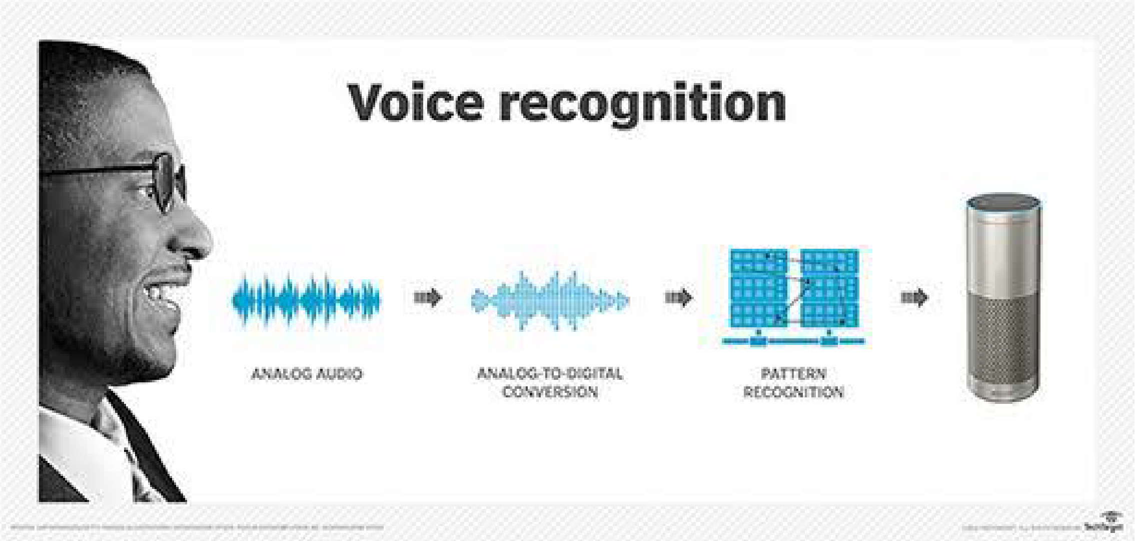


Figure 1.1: Voice Recognition.

At present, there is a rising pattern to make data, including individual data of people accessible through the web, which builds the danger of fraud and identify theft, causing security to be probably the most serious issue in modern science and technology [9]. This issue can be solved by utilizing biometric authentication like voice recognition which depends on human physical attributes. The primary benefit of voice recognition is that it tends to be utilized all the more effectively in an assortment of utilization such as controlling gadgets at home or online transactions [10]. This paper discusses and analyzes on the method of developing a voice recognition system using deep learning based approach that will enhance the security system. The voice recognition system will be using a central database to store voice sample of individuals and later using the sample for voice recognition and comparison. The paper will further analyze the key parameters for voice recognition which are important with the objective to use the voice as a biometric identification system to improve security.

1.1 Research Problem

Voice recognition is not a new thing in today's world. It is being implemented everywhere to ease our daily tasks. Moreover, many companies are coming up with new AI's every year that can recognize our speech and work accordingly. We can connect and do some of our most essential work for a day by only using our voice. Some of the home appliances, phones, smart home manager devices can understand our voice command. But the most important thing about voice recognition is that it has much more potential to be included in our important aspects of life such as security. To be exact, voice recognition can be used to verify a person with only using their voice. Machines can already learn, understand and store our voice that has been proven already, but the question lies in the matchmaking part of the process. If the matchmaking process is 100% efficient then it would be only a matter of some time for it to be start implementing in security sectors. As we already know voice recognition is a biometric feature that varies from person to person and it is already being implemented in our daily tasks. The implementation of voice recognition in verifying a person is not afar. Furthermore, if the accuracy is not up to the mark, then it might become dangerous because the chances of failing to verify the correct person will significantly rise. In the paper we will be taking a database of voices and use machine learning algorithms to check and verify the most efficient technique to find out a person by only using his voice. Many algorithms are already being used and sophisticated features like patterns and graphs of voices are being used for the verification of voice. There are so many things that needs to be tested, for example the estimation technique or the decision tree technique. We will be testing and exploring the techniques of Machine Learning because of it being more efficient than the traditional methods in the case of both time and accuracy, also search the most accurate technique so that the verification process is successful every time with enough information. Ultimately, if voice recognition is implemented in the security sectors, then we are the ones who are going to get more secure. As fingerprint scanner, iris scanner etc. is already a good biometric verification method. If we add voice recognition with these systems, which is better than the aforementioned method regarding many cases for example in collecting audio sample is easier than fingerprints in general scenarios, we are getting an extra layer of security on top.

1.2 Research Objectives

This research intends to focus on developing a security system to identify an individual as well as his national identity with the help of a central database and using some voice recognition techniques of deep machine learning. The purpose of this study is to:

(i) Understand how various machine learning algorithm works for extracting features from a person's voice.

(ii) Using a central database to get the most voice samples for matching.

(iii) Find out the most effective technique to get the most accurate result of detecting a person.

(iv) Evaluate the model.

(v) Similar to biometric system, implementing the voice matching for a better security in different large scale sectors.

(vi) Increasing the accuracy of gender predicting model.

Chapter 2

Literature Review

Speaker identification is the capability of a machine to identify an individual by interpreting voice of that individual. Speaker or voice recognition can be used for a variety things for example for voice commands, biometric security, voice search, voice to text, forensic science etc. In biometric security voice recognition is used for uniquely identify individuals using biological features. If we look at the history of voice recognition, the first speech recognition system was built in 1950s. At that time, it was designed around numbers instead of words. Slowly the ability of machine increased and from numbers we advanced to words and then moved to multiple voice in 1970s. During the 1980s the most popular method of voice recognition HMM (Hidden Markov Model) was seen [23]. At present the world has seen Siri, Cortana which are great examples of voice commands. Exploration on utilizing voice recognition for security, voice commands, forensic science and so on has been continuing for quite a while. On this paper, we focused on using voice recognition for security purposes and we went through some of those recent previous and recent research works to find motivation. Voice/Speech recognition has progressed from various perspectives in recent decades. The concept of deep learning in voice recognition can advance the research work in ways that the traditional methods of voice learning could not.

2.1 Classification

Akhilesh Halageri, Amrita Bidappa, Arjun C, Madan Mukund Sarathy and Shabana Sultana discussed utilizing deep learning in voice recognition and emphasized on speech signal patterns using neural network. They also examined about the issued surrounding the existing methods of ASR like HMM, GMM. The existing methods fall behind in case of cost, nonlinear manifold in data space, computing time and so on. Their idea involves processing the signal from input signal, extraction of feature, use deep learning models for the training data and learning environment and then the recognition of speech. For feature extraction, they utilized FFT/DFT and MFCC etc. Moreover, they pointed out the advantages of using deep learning over existing methods. The target of using deep learning is to make the models learn patterns instead of assuming, making it an automated system. Also, compared to HMM , GMM, deep learning models have better pattern recognition rate according to the authors.Although, in case of training procedure, MLP is hard to train, and the Delta Rule cannot be applied to MLP directly, they proposed a solution using backpropagation, a method of training artificial neural network [6].

Pramit Gupta, Somya Goel and Archana Purwar on the other hand, focuses on taking an approach for distinguishing between male and female by using stacked Machine Learning algorithm which employs acoustic features and comparing with existing algorithms. They proposed that the dataset be separated in 3 portions, out of which 2 portions are used in learning purposes and another one for the final test. They used CART (Clarification and regression trees) and SVM (Support Vector Machine) and neural models on 1 portion of the training data. However, various models make various predictions as they utilize different techniques. Even majority voting system sometimes provided incorrect result. So, for ensuring better prediction, they used ensemble learning techniques which takes numerous predictions gathered from the models as it's input and train another model to refine the predicted outcomes. The advantage of using dataset and neural network model is more accuracy in the voice recognition system [11].

In another research paper, Sunil Agarwal, Shruti A.K. and C. Rama Krishna inspect and explore the performance of 4 Machine Learning algorithms of voice recognition which are MLP, RBFN, C4.5 and BayesNet. Among these,

- **MLP (Multilayer Perceptron)** : Better for gender recognition.
- **RBFN (Radial basis network)** : Better performance in case of mass population size.
- **C4.5** : Better in case of consistency. Uses the conception of information entropy.
- **BayesNet (Bayesian network/belief network)** : Less sensitive to small data set size. More suitable for rapidly changing environment.

Their proposal is speech signal acquisition, feature extraction (LPC), Recognizer using Machine Learning algorithm and finally the outcome. Speaker recognition is mainly made of two steps namely feature extraction and recognizing speaker. They used COLEA, a MATLAB software for speech analysis and WEKA, an open source simulator for ML algorithm. While conducting their test, after comparing the results, it was noticed that in case of performance, RBFN has the most accuracy rate (97.3) and less misinterpretation rate (3.1) in case of 10 speakers and C4.5

performs better in case of 5 speakers. There are some problems for example in case of LPC determining formants from signals of speech is difficult. But, from overall comparison, we can notice that RBFN is suitable for recognizing, and also regarding the size of testers while MLP surpasses other models in case of gender recognition [1].

We looked into another comparison done in a research work by Nishtha H. Tandel, Harshadkumar B. Prajapati and Vipul K. Dabhi where they focused on a survey on Machine Learning techniques in voice recognition. As voice recognition is an unpredictable issue as the voice of an individual can change because of old age, throat illnesses or feelings, they focused on differences among different methods and the differences between traditional methods and deep learning methods as well. Speaker identification can be separated by 2 kinds, speaker identification as well as speaker verification, both which will be focused on in our research. Moreover, speaker recognition can be divided into Text-Dependent and Text-Independent field as well based on words or texts used in speech. In Text-Dependent same data is used for learning and testing while in Text-Independent different data is used and a dataset of different training sample of the same speaker is needed. They also mentioned four different approaches of forensic voice comparison which are Auditory, Spectrographic which is generally voiceprints or frequency spectrum, Acoustic in which we can see pitch, formants, fundamental frequency and HNR, and lastly Automatic Approach like MFCC, LPCC and so on. In any event 2 accounts of the speaker are required in every one of the methodologies referenced above. While describing Spectrographic approach, it was described that it is an approach based on image where speech records are transferred into speech images which are called spectrogram. Furthermore, the traditional methods for example, HMM , GMM and Vector Quantization was also mentioned. They showed in details the deep learning methods which requires the steps data acquisition, pre-processing, deep learning models and finally resemblance measure. VAD (Voice Activity Detection) is used to discover human presence or absence in speech. Speaker diarization is used to split or distinguish an input sample which is a wide area and separate subject itself and we will not explore that area. It is noted in their research that Siamese NN (Neural Network) is appropriate for voice comparison which is used for distinguishing between two input, whereas CNN is better in case of voice classification. Siamese NN has a good estimation rate in case of small training data. Our research requires a large set of data. Moreover, Siamese NN has to work with single sample that is, it does not require multiple samples of the same speaker [19].

In another work, Saritha Kinkiri, Wim J.C Melis and Simeon Keates describes how to identify voice, unique features have to be identified and afterwards compared with previously acquired data. They focused on the parameters that has to be identified in order to analyze voice sample. In their opinion, the key feature of using voice in biometric security lies in understanding the parameters of voice/speech. They show the steps: input signal, Fourier Transform which is for understanding frequency components, pattern recognition, training of data and testing, and finally the result. They brought up the fact that speech recognition is language independent which means accent independent as well. They focused more on the understanding parameters as voice is a longitudinal wave and it changes on the usage of vocal as well as other physical attributes. The voice of male and female can be distinguished by their number of vibrations per second for their vocal chords. Also, the chest can be considered a voice parameter as well. Additionally, they discussed biometric authentication. Verification is to check if someone really is who that individual claims to be whereas identification is to find an individual among a massive population by comparing the data of that individual with the data in the database, both of which is similar to our goal of research. Biometric technology is of 2 types, physical and behavioral. While zooming in on biometric voice recognition, the system 1st identifies unique features and then store them in database. For this, it is necessary to know which features or parameters are to be focused on and also the methods of measuring physical parameters as it can be challenging. They researched on how human ear works while hearing to use that same concept in case of machines. The existing methods mostly use syllables for speech recognition but there are other body parts which affects sound although some are less significant. Using only syllables can limit the accuracy rate of speech recognition. Moreover, they pointed out that as voice is a physical feature and cannot be changed, if it is compromised then the user will be in a tough spot [10].

The authors of another study discussed the idea of creating a smart doorbell system using Machine Learning algorithms while saving energy consumption using iris as well as voice recognition system. We will be focusing on the voice recognition part only. Their proposed steps for voice recognition was feature extraction, energy detection and normalization (VAD operation), GMM models for pattern matching operation, Tnorm, Znorm (normalization operation) and scoring. Audio sample is collected from the input block and then it is checked whether the sample is a human voice signal. If it is, the sample is extracted and based on the quality it is compared with the samples previously stored in the database which is done using GMM. However, the only problem is that the whole system is dependent on how well the user will cooperate and also the sample quality which affect the time required to conduct the process [8].

Meanwhile, Mr. Vijay K. Kale, Dr. Prapti D. Deshmukh and Mr. Hanumant R. Gite, in their paper, give significance on MFCC and LPC features for identifying speaker. Speaker recognition has 4 stages, sample analysis, feature extraction, modeling and testing. Feature extraction plays a significant role as it's job is to separate one speech from another. There are many types of features, for example,

- **Spectral feature** : Descriptors of short-term speech spectrum.
- **Dynamic feature** : Relevant to time evolution of spectral and other features.
- **Source feature** : Refers to the features of glottal voice source.
- **Super a segmental feature** : Span over several segments.
- **High-level feature** : Refer to symbolic type of information. For example, characteristic word usage.

MFCC (Mel Frequency Cepstral Coefficient) is the 1st method tested on their research. Its estimate of the human system reaction is superior than other methods because frequency bands are placed logarithmically.It is based on brief analysis, therefore an MFCC vector is calculated from each frame. . And LPC (Linear Predictive Coding) is a mathematical computational operation which is linear combination of several past samples. For the test done on MFCC feature extraction, KVKMMBR speech standard database was used and the tests were done in MATLAB software [8].

Moving on, Asti F. Fadlilah and Esmeralda C. Djamal discussed ways of bringing voice recognition in the daily activities by introducing voice instructions like smart doors, smart lights etc. They proposed to use MFCC for feature extraction and hierarchy SVM for voice recognition algorithm. In general, SVM can only distinguish from two classes. Hence, the process of recognizing more than two classes in stages is known as hierarchy SVM. They also used backpropagation algorithm for training data. Their work shows the comparison between SVM and hierarchy SVM and accuracy rate of recognizing voices of different persons as well [17].

In another study, a test was conducted on MFCC and LPC to compare and it was noted that MFCC was better at identifying speech signals than LPC [5].

In another research, Jayashree Padmanabhan and Melvin Jose Johnson Premkumar discussed about various Machine Learning techniques that can help to improve ASR (Automatic Speech Recognition), although they focused on the traditional methods. Furthermore, the advantages and disadvantages of using SVM (Support Vector Machine), Deep Learning based on ASR is also discussed in the paper. The authors stated that HMM (Hidden Markov Model) is the most dominant technique rather than using the Markov Model which was used in the 1970's when for the first time the concept of ASR came. HMM algorithm has a noticeable issue. The algorithm requires a large amoung of training data to avoid performance loss. Because the condition of testing and training might vary. To estimate the output density, GMM

(Gaussian Mixture Model) is used alongside HMM. This HMM/GMM systems are by far the most prominent learning approach in ASR stated by the authors. As there are more places to improve the researchers always thought about alternative estimation approaches that led them to use various ANN (Artificial Neural Networks) approaches. As time went by, the researchers found the SVM (Support Vector Machine) to be more helpful as it is generative and discriminative. These two things are complementary to each other and the excellent generalization capabilities of SVM improves the robustness of SVM. The HMM/GMM systems are going to get used more often as learning model for ASR as it improves the performance significantly [7].

Moreover, focusing on the work of Jong Won Shin, Joon-Hyuk Chang and Nam Soo Kim, we can know about Voice Activity Detection (VAD) and the conventional ways to improve this such as Laplacian and Gamma Distribution. Furthermore, they have also showed that Generalized Gamma Distribution (GTD) based approaches outperform these conventional techniques and can be used alongside Minimum Classification Error (MCE) and Support Vector Machine (SVM) to further enhance the performance of the VAD. The author discussed that all the data used for the test of the MCE technique was recorded from 4 male and 4 female speakers. After the test it was found that the proposed VAD with Machine Learning techniques such as MCE and SVM did the same as the conventional methods in clean channels. But the proposed method worked better in all the noisy channels. After that the authors confirmed that the proposed MCE method enhances the performance of the VAD. For the testing of the SVM, vehicular and street noises were also added with the test channels. Ultimately, from the result the authors claimed that the proposed method performed better or at least equal to the conventional methods. The suggested SVM approach was shown to have the greatest detection capability. But the large number of false alarm conditions still needed improvement. Finally, the authors explained that the method they proposed VAD with MCE and SVM based approach performed better than the existing and already very widely used approaches [3].

Then, looking at the works of Laura Verde, Giuseppe De Pietro and Giovanna Sannino we get to acquire knowledge about voice disorder. We have seen gradual rise of usage of phones in medical sectors and they discuss about health detections using mobile devices by utilizing Machine Learning techniques. In this examination, the precision in the separation of pathological from normal voices of the Machine Learning strategies was investigated to distinguish the most dependable one. From the experiments, RBF came on top with the best result [16].

2.2 Feature Matching

Moreover, By studying the work of Loh Mun Yee and Abdul Manan Ahmad, we may gain an understanding of various feature matching approaches as a type of comparative research. In their study, they analyzed the outcomes of DTW, GMM, and SVM for speaker recognition. The processes in their technique comprised data pre-processing, feature extraction, and pattern recognition determination. The variable MFCC was used to extract the features from voice samples. The whole project was conducted on Matlab 7.0. Among the 3 techniques performed for classifying speech pattern, SVM performed the worst due to several drawbacks while dealing with audio data. Implementing GMM yielded excellent results, while DTW did not lag far behind. In fact, DTW appeared to be catching up to GMM in terms of rise in speaker count [27].

Furthermore, the paper conducted by G.SUVARNA KUMAR, K.A.PRASAD RAJU , Dr.Mohan Rao CPVNJ , P.Satheesh, They used MATLAB to analyze the speakers voice in this study report to identify them. They classified the speech in some key features for example pitch, frequency, amplitude etc. using MFCC and for feature extraction they used MATLAB. Training and recognition are the two modes used in this thesis. As a result, the training mode will record the speakers voice and create a model, while the recognition model will attempt to recognize the speaker based on the training set. Another separation technique is RASTA, which smooth out the noise by providing a high-pass filter to every frequency's output .

Training Feature Extraction +Modeling	Testing Feature extraction.	Identifying best candidates.

Table 2.1: Bloch Diagram of the proposed approach.

Feature extraction using MFCC ->Statistical modeling using Gaussian mixture model. Then after printing the voice data of all the speakers using GMM then they checked which voice matched accurately within the data. GUI takes the speaker filename and then give the name of the speakers. To conclude The GMM can identify the speaker within a very short training data. Even it can easily detect the voices with extra noises. [26].

In addition, the study from A Holy Quran Reader/Reciter Identification System Using Support Vector Machine ,the authors distinguish the reader of the Holy Qurans based on several characteristics of the correlating sound wave. They used 15 well-known Quran readers and MFCC to extract features from the input sound wave. The MFCC is used to compare the accuracy of SVM and ANN, with SVM identifying 96.59 percent of his readers and ANN identifying 86.1 percent of his readers. The pronunciation of Arabic is distinct. And to recognize the reader, they used

Multi Directional Local Features (MDLF), which produced a more accurate result than other methods. They also reasoned that the phoneme length could be used to shorten the HMM chain, resulting in greater accuracy. The proposed approach was Acoustic Signals of the Ayat ->Feature extraction using MFCC->SVM classifier (Trained) -> Reader .MATLAB was used to create the recognition system. The speech was divided into sections using Hamming window, with each section containing 15ms frames. The 15 ms frames, according to the paper, can detect more accurately because the reciter has a steady rate of recitation. The ayah vector was then applied, followed by the reciter name beside each label. The feature Matrix was combined into a single folder, and the features were reorganized with the help of a randomized filter property. ANN was used to compare the files. However, SVM is more accurate than ANN, with a greater accuracy and lesser MSE. The results of the study can be used to identify the Holy Quran reader. It also demonstrated why SVM is more accurate than ANN. They plan to work with more readers in the future and use the Gaussian mixture to improve accuracy [25]

Additionally, the research conducted by R. Thiruvengatanadhan discusses about converting the speaker's speech to text in order to assist deaf people. Thus, for recognition, the text from the speech will be generated using SVM. They also used MFCC to extract features from each sentence and voice activity detection (VAD) to separate word segments of each sentence. The support vector machine produces more accurate results in recognition. The Training process performed: Continuous speech-> Detecting voice activity ->Isolated words->MFCC feature extraction-> SVM models-> Speech Recognition. Testing process was: Continuous speech-> Detecting voice activity ->Isolated words->MFCC feature extraction-> SVM models-> Speech Recognition. The proposed method of Extraction of MFCC from Audio Signal: Audio Signal ->Framing and Windowing -> FFT -> Mel Filter Bank-> Log->DCT->MFCC .Among the SVM Kernels the polynomial's speech recognition rate was 78% whereas the Gaussian and Sigmodal's rate was 95% and 81%. So Gaussian was showing more accurate results. The VAD was used to separate the words, and SVM was used to design each of the Speaker's; speeches, as well as to recognize the speech with the help of training data. In this experiment, however, Gaussian produced more accurate results.[15]

Moreover, observing the study of Orken Mamyrbayev, Aizat Kydyrbekova, Keylan Alimhan, Dina Oralbekova, Bagashar Zhumazhanov, Bulbul Nuranbayeva investigates the issue of voice recognition and builds a method for recognizing a single uttered phrase. When in an authorization session, the imposter initially displays a legitimate client's identification card. The imposter then uses the speaker recognition mechanism to acquire access by replaying the pre-recorded voice that was actually provided by the client. So, Voice authentication technologies that are im-

mune to such assaults are unquestionably needed. The DNN x-vector design can handle more data for training than an i-vector design, which becomes overloaded over a certain quantity of data. For speaker identification, data augmentation is used to boost the effectiveness of deep neural network (DNN) systems. The findings reveal that DNN x-vectors use guided learning to enhance data. MATLAB and ARDUINO will be implemented in this project. The voice recognition aspect is handled using MATLAB software, whereas the communication mechanism is handled by ARDUINO development tools. The person's energy features are obtained and stored as a reference template employing MFCC. Silence detection is used to retrieve the actual said speech, while filtration is used to disregard the rest. The majority of the signal's silence is eliminated during filtering. To accomplish this, we'll apply probability density function concept to eliminate the noise and signal quietness. The study offered approaches based on DNN plus I classifiers as well as a variable-length x-vector system that picks a model and conducts identification considering the length of test voice samples. This strategy produced superior results for both validating the individual and recognizing in an extensive database.[21]
The article written by David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, Sanjeev Khudanpur discusses how Because of supervised learning, the x-vector effectively utilizes data augmentation. As a consequence, the x-vectors performs very well on the assessment datasets. I-vectors are used in the majority of speech identification systems. The PLDA classifier has been used to evaluate i-vectors and identify whether they belong to the very same speaker or are different.The most common DNNs used in speaker identification are trained as acoustic models for automated speech recognition (ASR) and afterwards utilized to improve phonetic modeling with in i-vector network. The research uses data augmentation to the DNN training technique, resulting in a considerable enhancement for the x-vector structure by increasing the quantity and variety of current training data. When analyzing x-vectors to i-vector baselines on Speakers, it's discovered that while augmentation helps the PLDA classifier, it doesn't improve the i-vector extractor. The PLDA augmentation tends to benefit the x-vectors more than the standard systems. Data augmentation has shown to be a simple and successful approach for increasing DNN performance. In experiments performed it was observed that, on SRE16 Cantonese, the x-vector system outscored two traditional i-vector baselines i-vector (acoustic) and i-vector (BNF) considerably.The x-vector DNN may be trained using simply speaker identifiers, making it excellent for domains with minimal captured speech.[14]
Likewise, from another study we observe that the authors examine a speaker verification system based on DNN embedding (x-vectors) with an emphasis on resilience across a variety of data environments. The embedding's are retrieved and utilized

in a typical backend such as PLDA to perform speaker verification. In text independent voice recognition, such systems have lately been shown to be effective for both short and long speech lengths. The study conducts trials using training data, examining the effects of augmentation and the number of training voices independently and aims to give a thorough examination of systems that rely on x-vectors. The article analyzed five alternative gender-independent PLDA models: one only on clear dataset, while the others incorporated intentionally generated sounds and reverberation. For training UBM and i-vector conversion, the test employed PRISM training dataset specification without additional noise or resonance. The paper noticed superior performance from the i-vector baseline while training the PLDA without augmentation in the artificially manufactured noisy and resonance situation. Tests with retransmission data reveal that the embedding system regularly performs better or similar to the original data, and that multi-condition learning of PLDA has a favorable effect. It can be seen that just providing more hours of learning data enhanced performance, and also that the trend of the multi-condition PLDA training's impact to effectiveness is similarly constant. The examined embedding design is found to work well under a variety of situations.[13]
As well, the study conducted by Fuming Fang, Xin Wang, Junichi Yamagishi, Isao Echizen, Massimiliano Todisco, Nicholas Evans, Jean-Francois Bonastre approaches to speaker privacy protection is discussed in this work. The objective is to employ neural acoustic and waveform models to generate anonymous speech by extracting linguistic and speaker identifying information from each phrase. The paper concentrated on concealing speaker identification while preserving voice quality by losing a small portion of the linguistic elements. We employed a speaker-independent automated speech recognition (ASR) system based on a deep neural network (DNN) to collect linguistic information in form of a phoneme posteriorgram (PPG), and a pre-trained x-vector method to capture the voice identification. This work focuses on using neural acoustic and waveform models to construct an anonymized verbal waveform provided the PPG, anonymized x-vector. The suggested approach of anonymizing speech is successful in hiding speaker identification, with anonymized speech identical to the original speaker identity, according to experimental data. First from input signal, the designed methodology extracts an x-vector, a PPG, and the fundamental frequency. The x-vector is then anonymized using data obtained from x-vectors of external speakers. The outcome of the study shows that the suggested approach may successfully mask voice identification. When the softmax layer was evaluated to the other PPG situations, the softmax layer had a higher EER. Results also show that when x-vectors are averaged over a huge number of unknown speakers, the proposed system will be unable to accurately retrieve the input speech's actual linguistic characteristics. The PPG is employed as a symbol

of the linguistic information, and also the anonymous pseudo speaker is constructed by combining numerous speakers' x-vectors.[18]

Furthermore, the paper by authors Thayabaran Kathiresan, Arjun Verma and Volker Dellwo have talked about the underlying acoustic disparities between men and women when it comes to voice recognition (ASR). The system was based on a UBM with Gaussian mixtures and gender- independent variability matrix. For training, PLDA models with 200 latent identities were used and the dataset were the Voxceleb1 and Voxceleb2. After training the i-vector system, the recognition part was carried out separately for male and female. The results showed that the equal error rate was higher for the female speakers and that was almost 5.128% which was significantly higher than the male speakers which was only 3.937%. A dimension reduction technique was used to reduce the 400-dimensional i-vector to 2- dimensional features. As a result of the findings, it was discovered that male speakers are relatively fewer than the female speakers. Finally, the result was described in two possible ways. The first is that recognition performs better for male speakers because speech recognition technology was created primarily with masculine voices in mind. And in several scientific domains, this kind of prejudice in research knowledge is well acknowledged. The second is that its likely that the acoustics of male voices provide a broader range of indexical indications to identity, making them easier to detect.[24]

Additionally in the study by Noor Salwani Ibrahima, Dzati Athiar Ramli have explained how using a simple factor analysis, often known as i-vector analysis, a new low-dimensional speaker- and channel- dependent space is created. Furthermore, the authors have evaluated the impact of the parameter adjustment and kept track of the computation time. In this study, the tool variability model or i-vector technique, which was originally developed for speaker recognition, was utilized to test a frog identification system. Firstly, to eliminate the non-frog-sound component, basic energy-based voice activity detection (VAD) is used in the methodology part. Because the i-vectors modeling comprises both speaker and channel variability information in one space, channel compensation techniques in the total factor space are necessary to eliminate the nuisance effects. The system's identification result is determined by figuring out the matching score . The studies were carried out by the authors is on the Ubuntu operating system with the Spear toolkit. Spear is a state-of-the-art speaker recognition toolbox that is open source and customizable. Bob, a free signal processing and machine learning library, provides the foundation for this toolbox. The authors discovered that when the i-vector dimensionality increases, accuracy improves. Furthermore, the data revealed that smaller UBMs outperformed larger UBMs. Finally, the authors stated that, the effect of i-vector extractor parameters such as UBM size and i-vector dimensionality on frog identification detection accuracy was investigated and it was discovered that selecting UBM training data is the most important factor, followed by i-vector dimensionality. The authors also proposed that the effect of increasing i-vector dimensionality and larger UBM size be investigated.[12]

Besides, In the paper of authors Sinan E. Tandogan and Hüsrev Taha Sencar study the individuality of the human voice with respect to a widely used feature representation of speech utterances, namely, the i-vector model. In this study, the distribution of i-vectors is estimated. As a result, i-vectors are quantized while ensuring that both the quantized and original i-vectors are preserved.The authors examine the uniqueness of the i-vector representation of speech, which is commonly employed in speaker verification systems. And also give a brief review of methodologies, examine their theoretical underpinnings, and assess their usefulness to evaluating the distinguishability offered by the voice as a biometric modality. This study builds on previous research by carefully investigating the effects of i-vector discretization and determining how elements crucial to modeling between-speaker and within-speaker variability affect uniqueness estimations.Voice activity detection is applied to all utterances to remove silent parts and to eliminate those that are too short to extract a single i-vector. All of the remaining speech samples are subjected to measurements. The findings show that within-speaker variability has a greater impact on the accuracy of estimates. When speech samples are gathered in less controlled situations, estimates are found to be much lower. i-vectors generated from larger speech samples, on the other hand, allow for improved speaker modeling and result in higher uniqueness estimations. The authors stated that, speaker embeddings in general cannot isolate speaker variability from channel and session related fluctuations, which is one drawback of the uniqueness measure in the context of employing a feature representation of speech.[22]

In the study done by XinXing, J., Xu, S. it was addressed that speech recognition is a technology where the computer understands the word given through speech, rather than using a keyboard. Although current voice recognition has attained great recognition accuracy in principle and in the lab, many speech recognition techniques are implemented on computers. This paper mainly focuses on DSP for speech recognition and to improve the recognition rate a high performance DTW algorithm which is different from the traditional one is proposed for achieving better recognition result and real-time control. As DTW is a basic technique that is simple to implement and has no severe hardware resource requirements, it is frequently employed in speech recognition. The DTW algorithm computes the matching distance using the point matching method; the matching distances between the test voice and each reference speech template must be computed; when the volume of the reference template and the test voice increases, the recognition time increases dramatically.

The key technology of the DTW recognition system is to improve recognition speed without lowering recognition rate.[4]

Moreover, the authors Lindasalwa Muda, Mumtaj Begam, I. Elamvazuthi explained that the results of a voice recognition research employing the MFCC and DTW algorithms are presented in this publication. After acquiring voice recordings, the digitized speech samples are analyzed with MFCC to yield voice characteristics, according to the report.The speech characteristics coefficient may then be routed via DTW to discover the pattern that best fits the database plus input frame, lowering the error among them. If one time series can be ”warped” non-linearly by expanding or shrinking it along its time axis, this technique can be used to discover the ideal alignment between two time series. The primary idea behind DTW is to distinguish two dynamic patterns and identify the smallest distance among them to estimate their similarity.When a person repeats the same word, the length of the word changes with each repetition. Speech recognition is erroneous or rejected as a result of this. The DTW (Dynamic Time Warping) algorithm effectively solves such issues. The two voice recognition methods MFCC and DTW were addressed in this research, both of which are crucial in enhancing voice recognition performance. Based on the unique information contained in the speech signal, the approach was able to authenticate the specific speaker. The findings suggest that these strategies could be useful for voice recognition.[2]

Chapter 3

Methodology

The approach we took is to first acquire the voice samples and then extract the feature from the audio samples. For feature extraction we will use MFCC technique. After that two layers will be used for recognition. First of all, 9 classifier algorithms, namely Multilayer Perceptron, Radial Basis Function Network, Decision Tree, Random Forest, Gradient Boosting, Naive Bayes, K-nearest neighbor, Logistic Regression and Support Vector Machine will be used for categorizing the individuals from the samples as male and female as and the whichever algorithm will provide the best result will be accepted. After that, a sample voice sample will be taken and trained with the accepted classifier algorithm. With the predicted gender with the classifier algorithm, pattern recognition algorithm like GMM will be performed on the second layer for detecting the person and while using the pattern recognition model, if the result of classifier was male, then in the database, it will be crosschecked with only the male samples. If the result was female, then it will be crosschecked with the female samples.

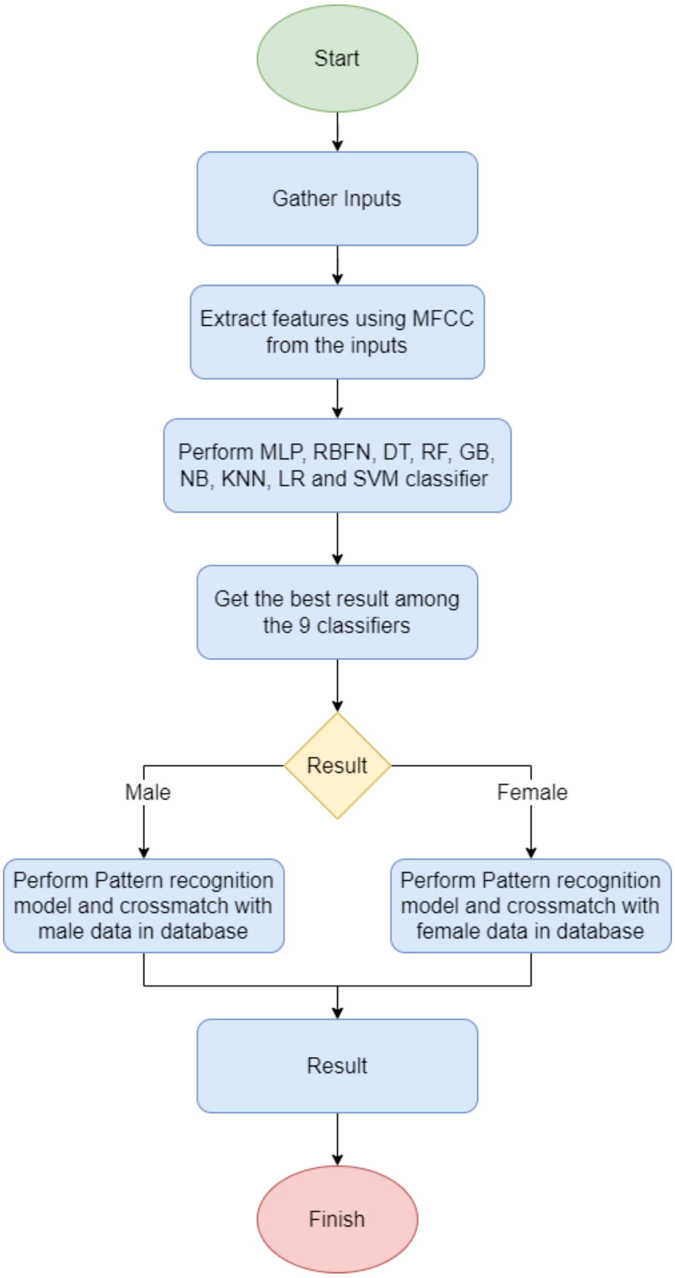

Start
Gather Inputs
Extract features using MFCC from the inputs
Perform MLP, RBFN, DT, RF, GB, NB, KNN, LR and SVM classifier
Get the best result among the 9 classifiers
Result
Male
Female
Perform Pattern recognition model and crossmatch with male data in database
Perform Pattern recognition model and crossmatch with female data in database
Result
Finish

Figure 3.1: The flow chart of the proposed voice detection model.

Our proposed voice detection process that is responsible for processing, classification, and producing predictions. There are four primary stages to it:

1. **Input data pre-processing**: Feature extraction is done in this stage. Data pre-processing is necessary as data cleaning and transformation are ways for removing outliers and standardizing data so that it can be utilized to develop a model more readily.

2. **Processing**: This stage is concerned with classifying input data into groups of gender using the 9 classifiers and building 2 different data frames.

3. **Detection**: In this stage GMM is applied on the two divided data frames

to detect the audio samples using the extracted features. It is run on the whole dataset containing both male and female as well and the results will be compared.

4. **Predictions**: This stage is for predicting score on the two divided data frames.

The input data is inserted into the pre-processing stage and then used by classifier models to build two data frames; both of which are applied on by GMM to detect the data samples. For achieving this, the preprocessed input data is split into two groups; one group is used for training and the other group is used for testing the accuracy of the model and stratifying data is implemented.

3.1 Input Data

An audio dataset is a bit different from other datasets. Because, in audio datasets we need to extract the features from the audio samples. There are many datasets available for researchers in the field of voice recognition. But we manually collected audio samples of voice from different individuals via social media platforms. The dataset contains more than 150 voice samples of Waveform Audio File Format from different ages of both male and female individuals.

Figure 3.2 describes how the dataset was made using the features extracted from audio samples that was collected from individuals. The dataset contains various features like spectral bandwidth, spectral centroid and most importantly MFCCs. This dataset was collected from individuals and it does not primarily contain the data needed to detect voice samples. As a result, it must be processed before it can be used to create the proposed voice detection model in this research.

3.2 Data Pre-Processing

Noise removal: The first step that we took was to reduce noise from noisy audio samples. Noise in audio refers to the residual low-level and disruptive sound in the background[20]. For removing noise, we used the software adobe audition 2021. Removing noise from noisy audio signal is necessary as it can disrupt the features of audio sample. In figure 3.3 and 3.4 we depicted the relation of noise and the original signal before and after noise removal.

Feature Extraction: For feature extraction we used Mel-Frequency Cepstral Coefficients (MFCC). It is the most popular choice for speech recognition or identification since 1970. Speech is basically the convolution of vocal tract frequency response with

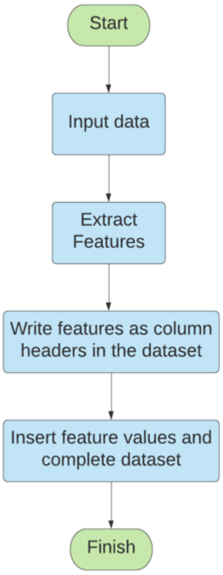

Start
Input data
Extract Features
Write features as column headers in the dataset
Insert feature values and complete dataset
Finish

Figure 3.2: The flow chart of the dataset creation process.

glottal pulse. The function of formalizing speech is,

$$log(X(t)) = log(E(t)) + log(H(t)) \tag{3.1}$$

here, log(H(t)) is the vocal tract frequency response which is necessary for separating components in formalizing speech. Figure 3.5 describes the working mechanism of MFCC.

We loaded a sample audio from our dataset and extracted the MFCC features along with the 2nd order of MFCC. The higher orders are calculated to show how the feature changes with each order. Figure 3.6, figure 3.7 and figure 3.8 shows the MFCC plots.

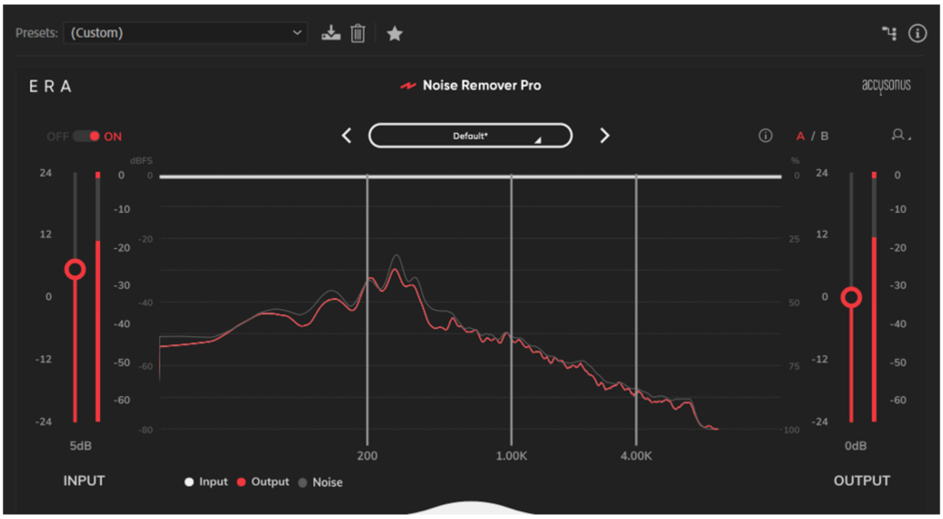


Figure 3.3: Before noise-removal.

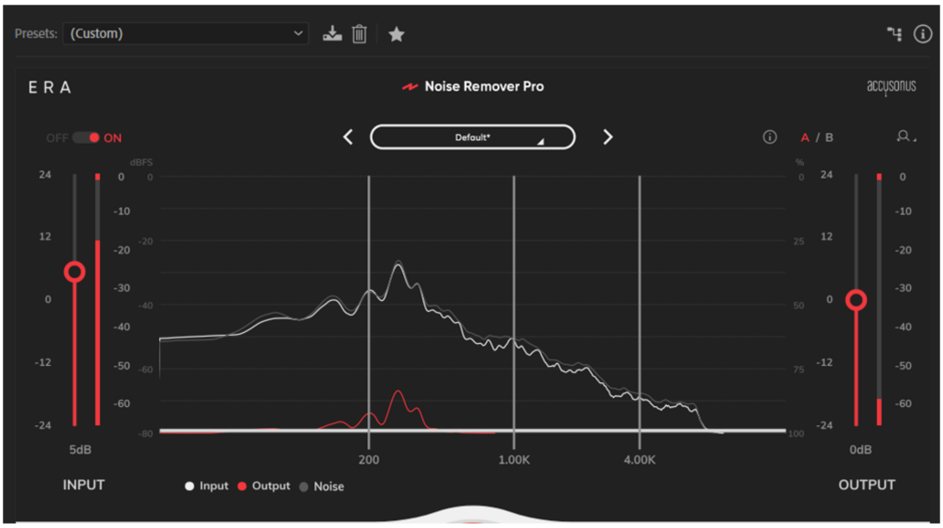
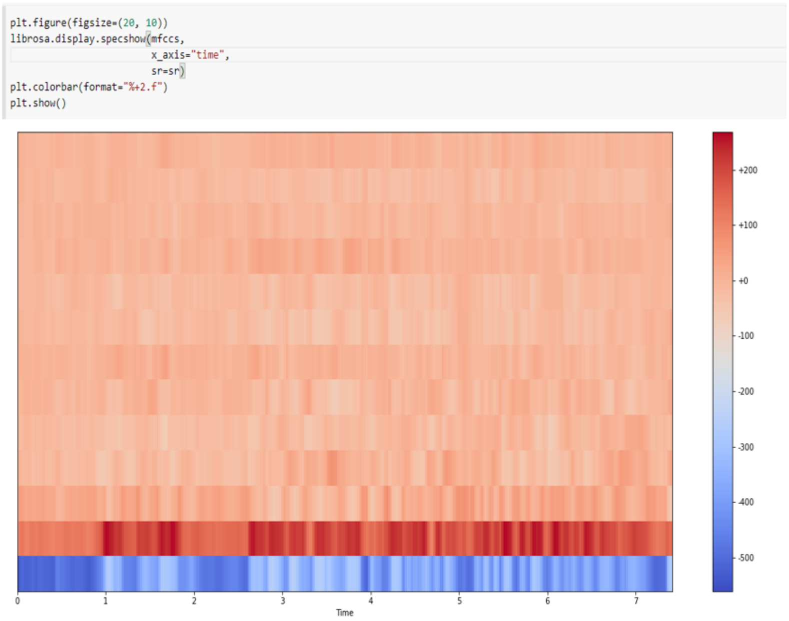


Figure 3.4: After noise-removal.



```
plt.figure(figsize=(20, 10))
librosa.display.specshow(mfccs,
                         x_axis="time",
                         sr=sr)
plt.colorbar(format="%+2.f")
plt.show()
```

Figure 3.6: MFCC extraction with time.

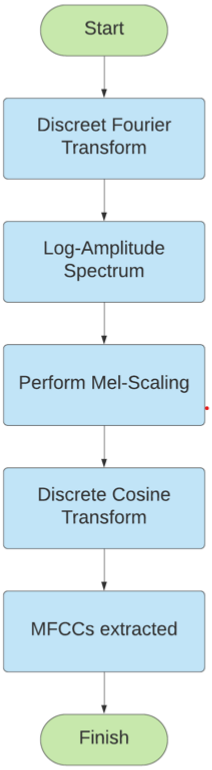
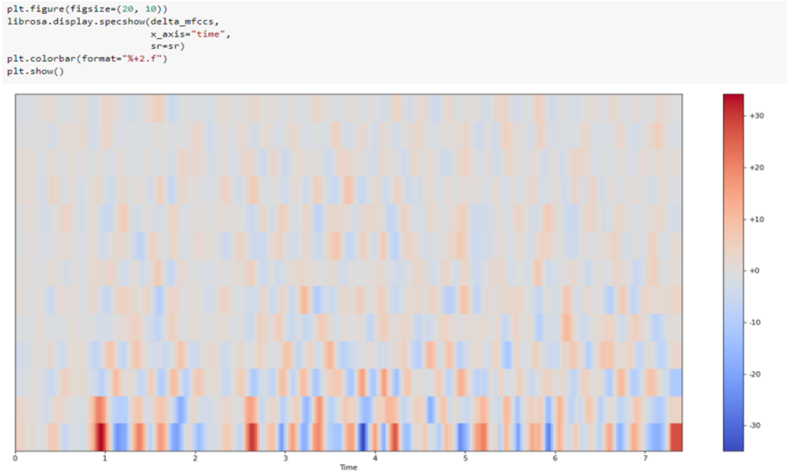
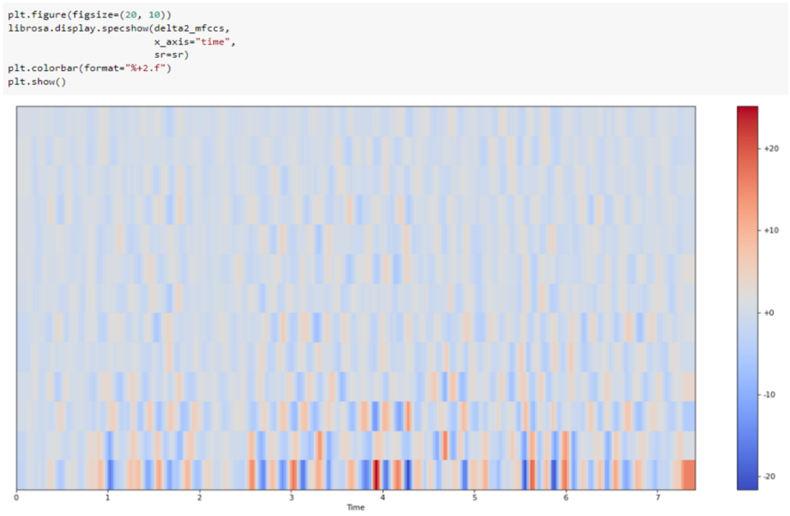


Figure 3.5: Working mechanism of MFCC.



```
plt.figure(figsize=(20, 10))
librosa.display.specshow(delta_mfccs,
                         x_axis="time",
                         sr=sr)
plt.colorbar(format="%+2.f")
plt.show()
```

Figure 3.7: delta MFCC extraction with time.



```
plt.figure(figsize=(20, 10))
librosa.display.specshow(delta2_mfccs,
                         x_axis="time",
                         sr=sr)
plt.colorbar(format="%+2.f")
plt.show()
```

Figure 3.8: delta2 MFCC extraction with time.

3.3 Classifiers and Pattern Recognition Methods

3.3.1 MLP

In the model, we have used MLP for gender identification over voice samples as gender identification using MLP has a very high recognition rate [17]. The Multi-layer Perceptron is a feed-forward artificial neural network that constructs output sets from a collection of inputs. Between the input and output layers of MLP, a directed graph is generated. It is a common deep learning algorithm that trains the network using a supervised learning technique called backpropagation [18]. In order to decrease error, training includes adjusting the model's parameters, or weights, biases and backpropagation adjusts those weights and biases relative to the error. Figure 6 depicts the representation of MLP process.

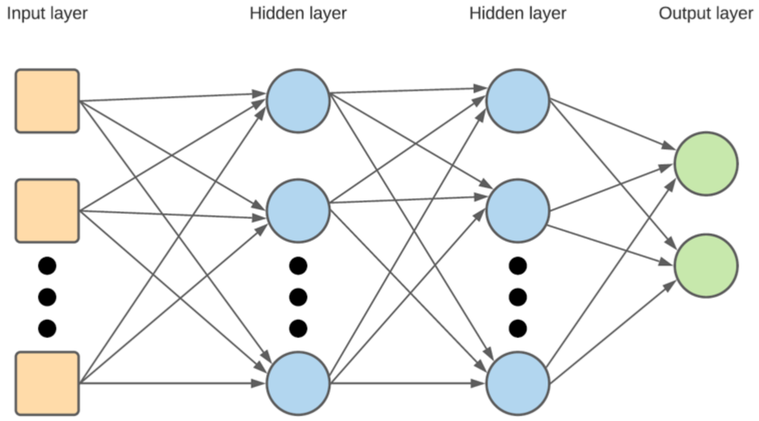


Figure 3.9: Representation of Multi-Layer Perceptron(MLP) model.

Computations in each output and hidden layer are given below:

$$O(x) = G(b(2) + W(2)h(x)) \tag{3.2}$$

$$H(x) = s(b(1) + W(1)x) \tag{3.3}$$

3.3.2 RBFN

Radial Basis Function Network is structurally similar to MLP. We used RBFN in our research for identifying speaker by analyzing features from speech samples. One of the biggest advantage of RBFN is that it is applicable in most dimensions because of low restrictions on how data are handled. This method generally uses the concept of Euclidean space Rn. In that case the function is:

$$s(x) = \sum_{j=1}^{m} \lambda_j \phi(||x - x_j||), x \epsilon R^n \tag{3.4}$$

Here, x_j are data points, ϕ is univariate, and the λ_j are scaler parameters[20]. Figure 7 shows the structure of RBFN.

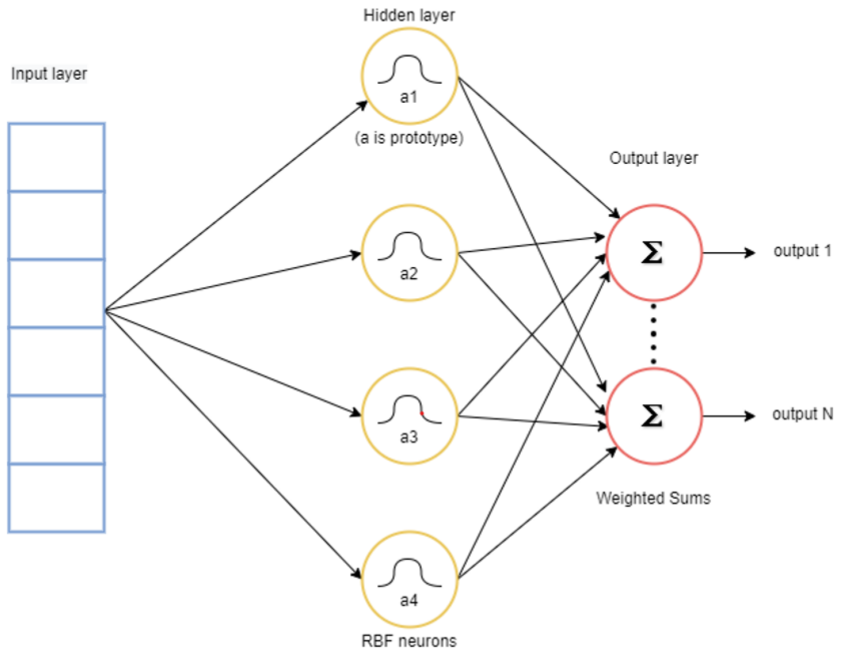


Figure 3.10: Representation of Radial Basis Function Network(RBFN) model.

3.3.3 Decision Tree

Decision trees are used as a tool for decision making that use the concept of a tree-like model of options and their possible results. The best way to demonstrate this algorithm model is, it is completely established on the basis of conditional control statements.

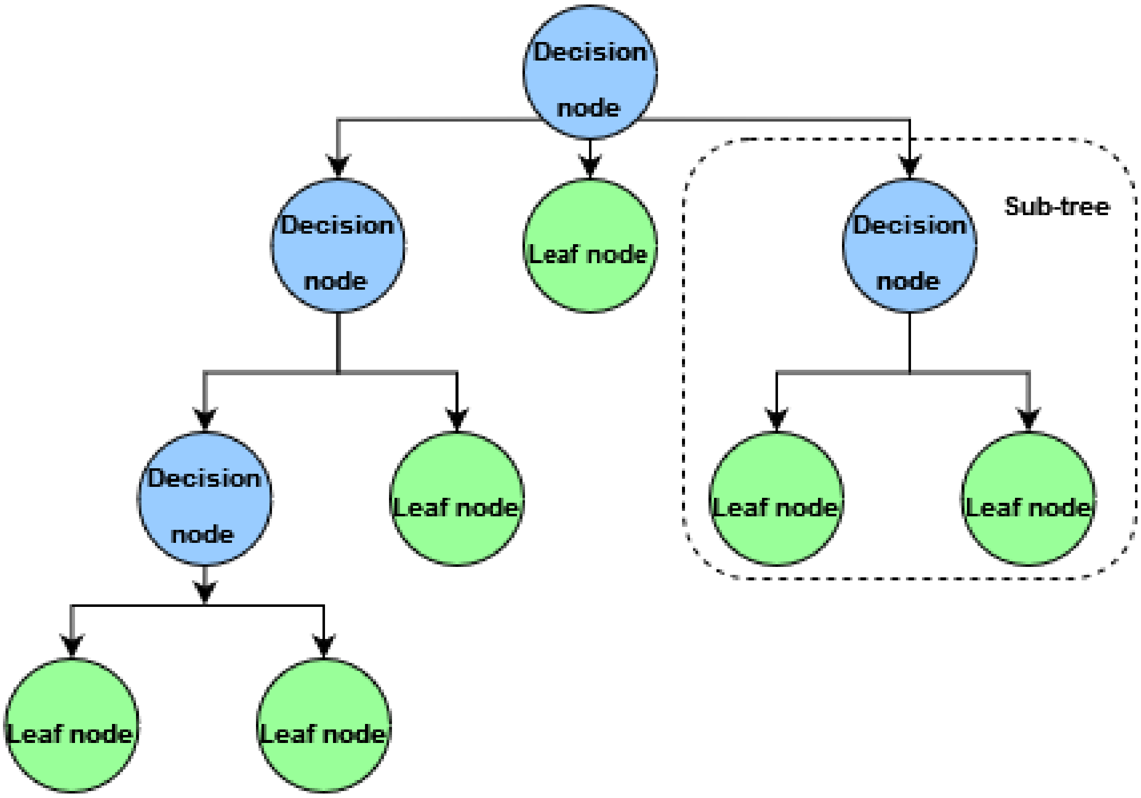


Figure 3.11: Decision Tree.

3.3.4 Gradient Boosting Classifier

Gradient boosting classifiers are a collection of machine learning algorithms that integrate a number of weak learning models to create a powerful predictive model which use decision tress. The implementation processes of gradient boosting are:

- First we have to fit the model
- Then we have to tune the model's parameters and Hyper parameters
- Next, we have to make predictions
- Finally, on the basis of those predictions we can interpret the results

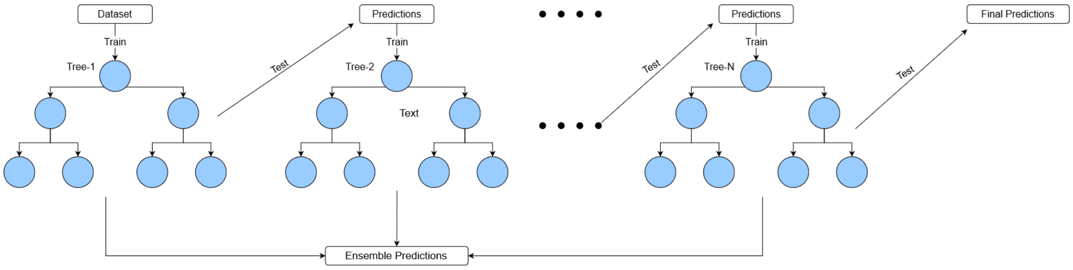


Figure 3.12: Gradient Boosting Classifier.

3.3.5 Random Forest

Random forests perform their work on training a large number of decision trees. The output of random forest is the class chosen by the majority of trees. Random decision forests address the problem of decision trees over-fitting their training set. As a result, they give better result than decision trees in most cases. However, they produce less accurate results than gradient enhanced trees.

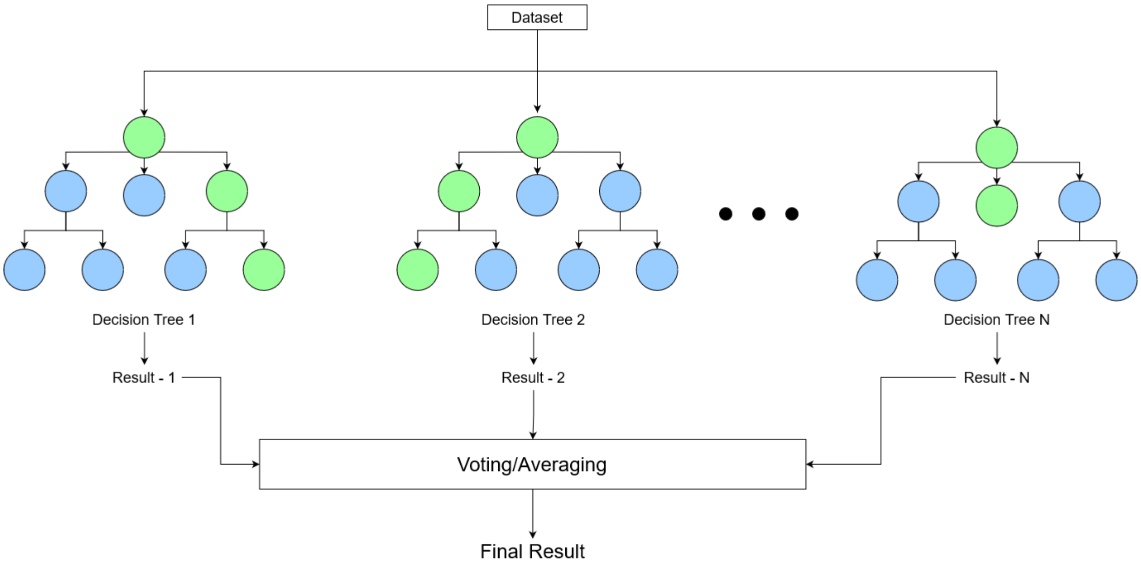


Figure 3.13: Random Forest.

3.3.6 Naive Bayes

It's a classification technique based on Bayes' Theorem and the predictor independence assumption. In simple terms, a Naive Bayes classifier assumes that chances of the presence of one feature in a class has no dependency on the chances of the presence of any other feature.

3.3.7 Logistic Regression

Logistic regression predicts the output and categorize them. Therefore, the result must be a categorical or discrete value. For example, Yes or No, 0 or 1, True or False etc. We fit an ”S” shaped logistic function which will be predicting our desired output as two maximum values from 0 to 1 in logistic regression.

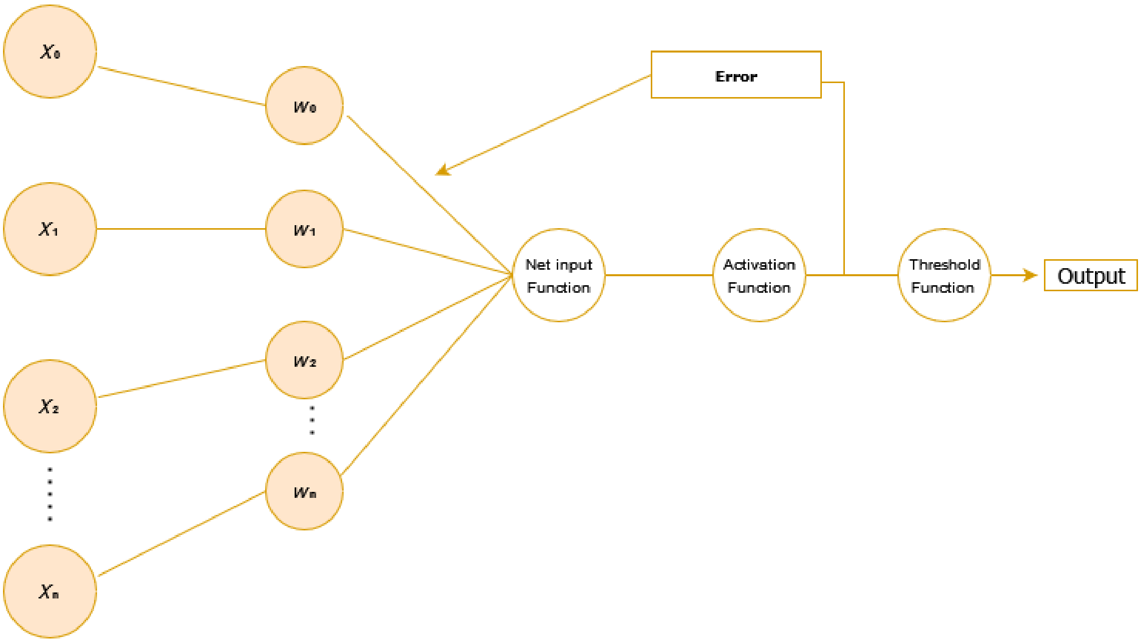


Figure 3.14: Logistic Regression Classifier.

3.3.8 Support-vector-machine Classifier

SVM is a classifier algorithm which is utilized in regression, classification, and outliner identification [26]. The fundamental goal of SVM is to split n-dimensional space into classes using a line or decision boundary so that data may be easily classified into the appropriate class. This decision boundary is also called the hyperplane [25].

Support vector machines have the advantage of being useful for high dimensions . Additionally, it is useful after the number of dimensions exceeds the number of features. Because the decision boundary is constructed from a subset of training points.

For instance, the SVM method can be used to categorize male and female voices. To begin, we must train our model with a variety of male and female voices so that our SVM may learn how to identify between male and female based on their characteristics. Thus, because the support vector establishes a decision border between these two sets of data (male and female) and selects extreme cases, it will observe the extreme cases of females and males.

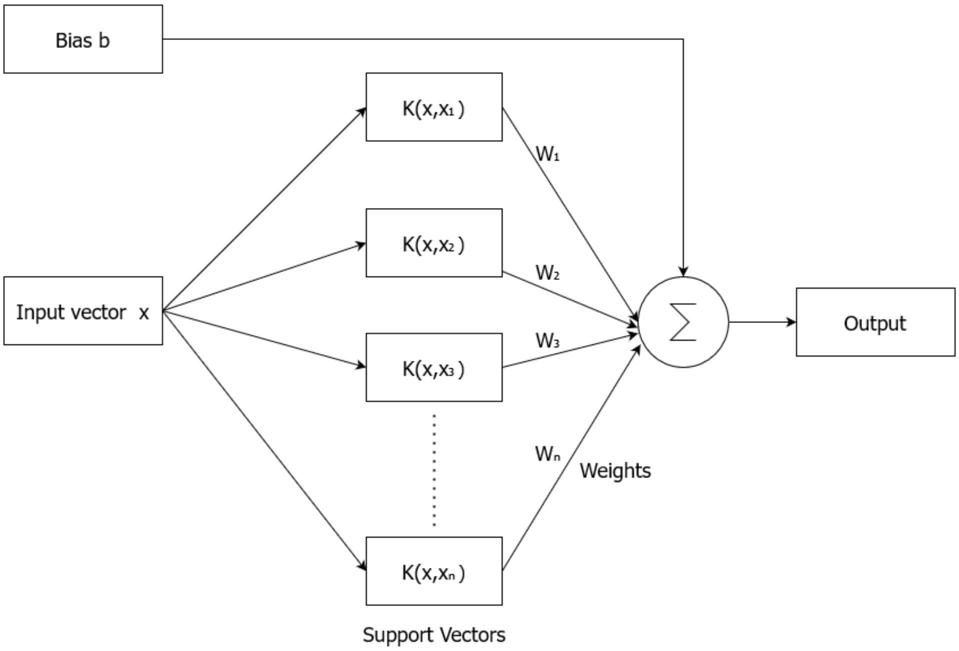


Figure 3.15: Support-Vector-Machine (SVM).

3.3.9 K-Nearest Neighbor

In K-Nearest Neighbor, the symbol 'K' denotes the nearest neighbors of a new unknown variable that must be predicted or categorized. For example, we sometimes share many common characteristics with our nearest peers. As a result, we tend to form friendships with those we share more common interests. The same approach is used in the KNN algorithm.

Its goal is to find all of a new unknown data point's nearest neighbors to determine what class it belongs to. It's a method based on distance. To classify the findings correctly, we must first identify the value of K, which must always be an odd number. Due to the fact that K is set to even, it is possible for a situation to emerge where the components of both groups are equal. It is preferable to choose an odd value for K since such a situation of equality between the two classes would never arise in this case. Given that one of the two groups would still be in the majority, an odd value for K is chosen.[35] The following table summarizes the effect of selecting a lesser or more significant K value on the model:

- **Increased K value** : When the value of k increases, the underfitting case arises. The model could not learn correctly from the training data in this situation.
- **Smaller K value** : When the value of k is smaller, overfitting conditions de-

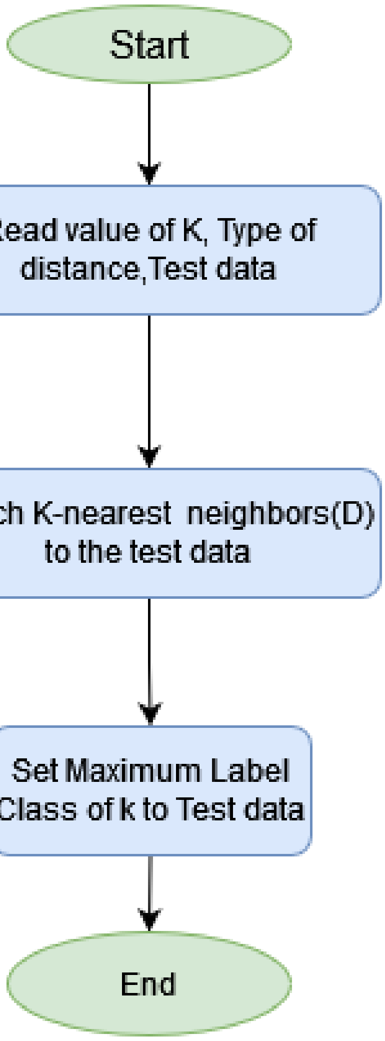

Start
Read value of K, Type of distance,Test data
Search K-nearest neighbors(D) to the test data
Set Maximum Label Class of k to Test data
End

Figure 3.16: K-Nearest Neighbor.

velop. The model will capture all of the training data, including noise. In this circumstance, this model will perform poorly on the test data.

3.3.10 Gaussian Mixture Model

Gaussian mixture models are effective clustering models for classifying audio based on their ability to distinguish between sounds. For example, differentiating between different instruments in a song or separating the speaker's voice from background noise when he speaks to his voice assistant. Gaussian mixture model is a probability distribution.

In the above figure, the clusters are highly defined because they are grouped into

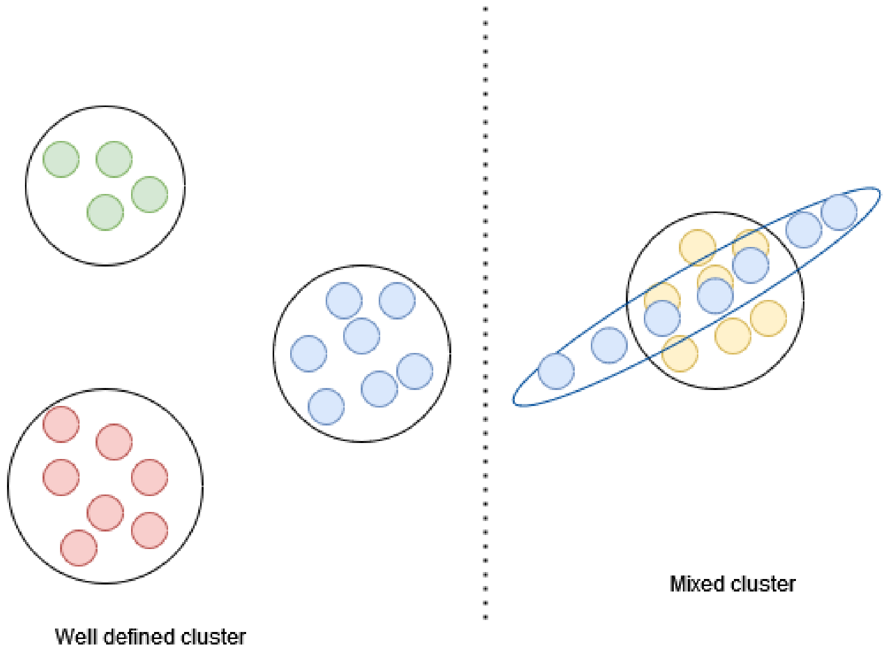


Figure 3.17: Gaussian Mixture Model.

three clusters. However in the figure above, two distinct cluster groups intersect. Thus, there are three clusters: one made of blue, one made of yellow, and another made of blue and yellow. So, GMM partitioned the dataset into distinct clusters, with the advantage that there is no requirement for specifying a relationship between the data points contained within the clusters. As this model is capable of classifying data-points into clusters and then applying learning, these clusters are created. By calculating the log likelihood of the voice vector and comparing it to previously gathered data, the Gaussian mixture model can recognize this speech. If the log probability of the voice vector matches the recorded data, the individual person can be identified [27]

Chapter 4

Implementation And Result

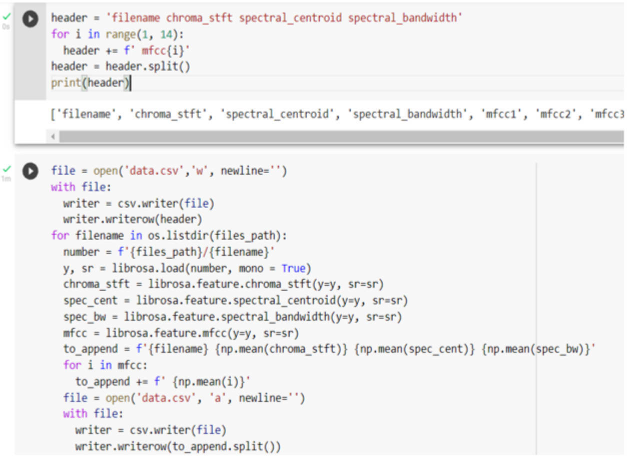
This section narrates the implementation of the proposed model of speaker identification based on gender classification. The 1st step is detecting speaker gender using the algorithm with the best result. The algorithm with the best result is selected after implementing multiple algorithms including MLP, DT, GBM, Naïve Bayes, KNN, LR and SVM. The 2nd step is to apply feature matching algorithms including GMM and DTW to identify individuals and compare among mixed gender dataset and dataset already separated by gender based on the result of the best algorithm in step 1. All the algorithms were implemented and tested in Google Colab. Firstly, the process of the 1st step contains four stages: creating the dataset, input data preprocessing, classification and testing. Secondly, speaker identification is performed on separate male and female databases. To begin with, the dataset is created with the extracted features of the data samples. After that, the input data are preprocessed for making the dataset suitable for training. The algorithms are trained then and at the end, tested.

4.1 Implementation

4.1.1 Creating Dataset

At first, an empty dataset (CSV file) with proper headers were created.Then, 13(default) MFCC features along with some other features like spectral centroid, spectral bandwidth from every audio sample were extracted and written in the dataset using python in Google Collab.
Furthermore, we used a Kaggle dataset containing previously derived MFCC features based on various emotions of more than 85 thousand males and females. Despite the lack of audio recordings, the MFCC features had already been retrieved, 58 per sample to be exact. The dataset was edited in a way that the features of the males and females were in the same database, features of males followed by features of

females.



```
header = 'filename chroma_stft spectral_centroid spectral_bandwidth'
for i in range(1, 14):
  header += f' mfcc{i}'
header = header.split()
print(header)
```

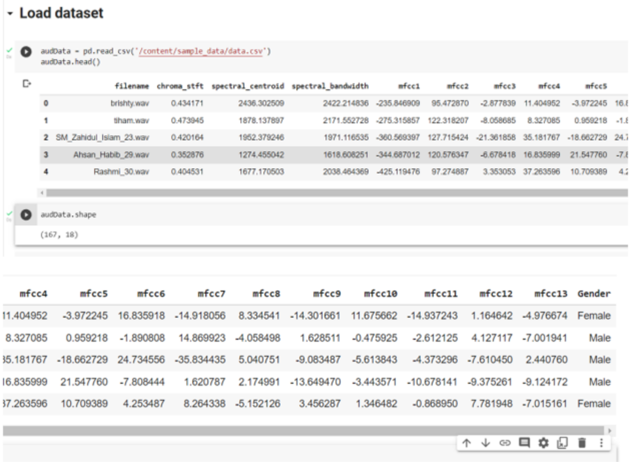
```
['filename', 'chroma_stft', 'spectral_centroid', 'spectral_bandwidth', 'mfcc1', 'mfcc2', 'mfcc3
```

```
file = open('data.csv','w', newline='')
with file:
  writer = csv.writer(file)
  writer.writerow(header)
for filename in os.listdir(files_path):
  number = f'{files_path}/{filename}'
  y, sr = librosa.load(number, mono = True)
  chroma_stft = librosa.feature.chroma_stft(y=y, sr=sr)
  spec_cent = librosa.feature.spectral_centroid(y=y, sr=sr)
  spec_bw = librosa.feature.spectral_bandwidth(y=y, sr=sr)
  mfcc = librosa.feature.mfcc(y=y, sr=sr)
  to_append = f'{filename} {np.mean(chroma_stft)} {np.mean(spec_cent)} {np.mean(spec_bw)}'
  for i in mfcc:
    to_append += f' {np.mean(i)}'
  file = open('data.csv', 'a', newline='')
  with file:
    writer = csv.writer(file)
    writer.writerow(to_append.split())
```

Figure 4.1: Creating dataset.

4.1.2 Input data Preprocessing

After the 1st data set is created with the manual data gathered from individuals, the CSV file can be downloaded. It will resemble the data shown in figure 4.2. The first column holds the filename of all the samples. 2nd, 3rd and 4th column holds the chroma, spectral centroid and spectral bandwidth and the rest of the columns show the MFCC values of each audio samples. 13 MFCCs were extracted from each of the sample data.



▾ Load dataset

```
audData = pd.read_csv('/content/sample_data/data.csv')
audData.head()
```

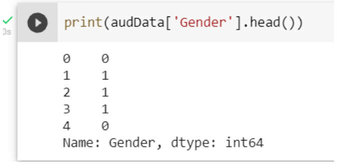
	filename	chroma_stft	spectral_centroid	spectral_bandwidth	mfcc1	mfcc2	mfcc3	mfcc4	mfcc5	mfcc6	mfcc7	mfcc8	mfcc9	mfcc10	mfcc11	mfcc12	mfcc13	Gender
0	brishty.wav	0.434171	2436.302509	2422.214836	-235.846909	95.472870	-2.877839	11.404952	-3.972245	16.835918	-14.918056	8.334541	-14.301661	11.675662	-14.937243	1.164642	-4.976674	Female
1	tiham.wav	0.473945	1878.137897	2171.552728	-275.315857	122.318207	-8.058685	8.327085	0.959218	-1.890808	14.869923	-4.058498	1.628511	-0.475925	-2.612125	4.127117	-7.001941	Male
2	SM_Zahidul_Islam_23.wav	0.420164	1952.379246	1971.116535	-360.569397	127.715424	-21.361858	35.181767	-18.662729	24.734556	-35.834435	5.040751	-9.083487	-5.613843	-4.373296	-7.610450	2.440760	Male
3	Ahsan_Habib_29.wav	0.352876	1274.455042	1618.608251	-344.687012	120.576347	-6.678418	16.835999	21.547760	-7.808444	1.620787	2.174991	-13.649470	-3.443571	-10.678141	-9.375261	-9.124172	Male
4	Rashmi_30.wav	0.404531	1677.170503	2038.464369	-425.119476	97.274887	3.353053	37.263596	10.709389	4.253487	8.264338	-5.152126	3.456287	1.346482	-0.868950	7.781948	-7.015161	Female

```
audData.shape
```

```
(167, 18)
```

Figure 4.2: A sample of raw data as obtained from the dataset.

Afterwards, it can be seen that the gender column is holding object type values. For increasing efficiency of training MLP, this column was encoded using label encoder. Figure 4.3 shows the Gender column after encoding where female is denoted as 0 and male is denoted as 1.

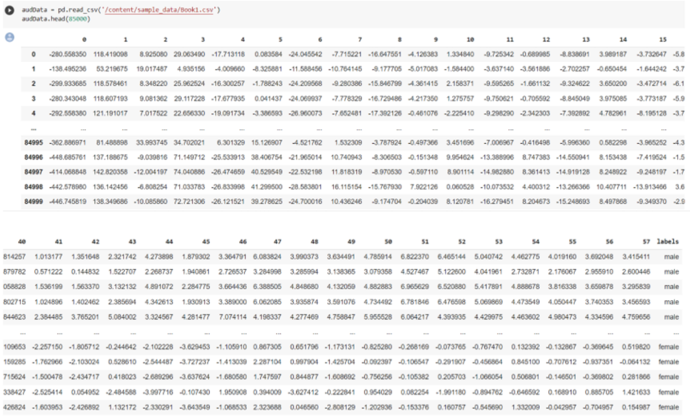


```
print(audData['Gender'].head())
```

```
0    0
1    1
2    1
3    1
4    0
Name: Gender, dtype: int64
```

Figure 4.3: Gender Column After Encoding.

Next, the sample database of MFCC features from over 85000 males and females was loaded into Google Colab. There was no need to start from scratch because the data was collected from Kaggle and then adjusted to include a new feature called ’Labels’ for gender. The database contains 85134 columns and 59 rows, 58 of them are MFCC features and one gender label. The database can be seen in figure 4.4.



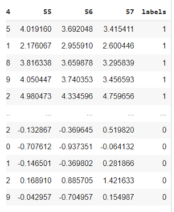
```
audData = pd.read_csv('/content/sample_data/Book1.csv')
audData.head(85000)
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
0	-280.558350	118.419098	8.925080	29.063490	-17.713118	0.083584	-24.045542	-7.715221	-16.647551	-4.126383	1.334840	-9.725342	-0.689985	-8.838691	3.989187	-3.732647	-5.8
1	-138.495236	53.219675	19.017487	4.935156	-4.009660	-8.325881	-11.588456	-10.764145	-9.177705	-5.017083	-1.584400	-3.637140	-3.561886	-2.702257	-0.650454	-1.644242	-3.7
2	-299.933685	118.578461	8.348220	25.962524	-16.300257	-1.788243	-24.209568	-9.280386	-15.846799	-4.361415	2.158371	-9.595265	-1.661132	-9.324622	3.650200	-3.472714	-6.1
3	-280.343048	118.607193	9.081362	29.117228	-17.677935	0.041437	-24.069937	-7.778329	-16.729486	-4.217350	1.275757	-9.750621	-0.705592	-8.845049	3.975085	-3.773187	-5.9
4	-292.558380	121.191017	7.017522	22.656330	-19.091734	-3.386593	-26.960073	-7.652481	-17.392126	-0.461076	-2.225410	-9.298290	-2.342303	-7.392892	4.782961	-8.195128	-3.7
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
84995	-362.886971	81.488898	33.993745	34.702021	6.301329	15.126907	-4.521762	1.532309	-3.787924	-0.497366	3.451696	-7.006967	-0.416498	-5.996360	0.582298	-3.965252	-4.3
84996	-448.685761	137.188675	-9.039816	71.149712	-25.533913	38.406754	-21.965014	10.740943	-8.306503	-0.151348	9.954624	-13.388996	8.747383	-14.550941	8.153438	-7.419524	-1.5
84997	-414.066848	142.820358	-12.004197	74.040886	-26.474659	40.529549	-22.532198	11.818319	-8.970530	-0.597110	8.901114	-14.982880	8.361413	-14.919128	8.248922	-9.248197	-1.7
84998	-442.578980	136.142456	-6.808254	71.033783	-26.833998	41.299500	-28.583801	16.115154	-15.767930	7.922126	0.060528	-10.073532	4.400312	-13.266366	10.407711	-13.913466	3.6
84999	-446.745819	138.349686	-10.085860	72.721306	-26.121521	39.278625	-24.700016	10.436246	-9.174704	-0.204039	8.120781	-16.279451	8.204673	-15.248693	8.497868	-9.349370	-2.9

	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	labels
814257	1.013177	1.351648	2.321742	4.273898	1.879302	3.364791	6.083824	3.990373	3.634491	4.785914	6.822370	6.465144	5.040742	4.462775	4.019160	3.692048	3.415411	male
879782	0.571222	0.144832	1.522707	2.268737	1.940861	2.726537	3.284998	3.285994	3.138365	3.079358	4.527467	5.122600	4.041961	2.732871	2.176067	2.955910	2.600446	male
058828	1.536199	1.563370	3.132132	4.891072	2.284775	3.664436	6.388505	4.848680	4.132059	4.882883	6.965629	6.520880	5.417891	4.888678	3.816338	3.659878	3.295839	male
802715	1.024896	1.402462	2.385694	4.342613	1.930913	3.389000	6.062085	3.935874	3.591076	4.734492	6.781846	6.476598	5.069869	4.473549	4.050447	3.740353	3.456593	male
844623	2.384485	3.765201	5.084002	3.324567	4.281477	7.074114	4.198337	4.277469	4.758847	5.955528	6.064217	4.393935	4.429975	4.463602	4.980473	4.334596	4.759656	male
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
109653	-2.257150	-1.805712	-0.244642	-2.102228	-3.629453	-1.105910	0.867305	0.651796	-1.173131	-0.825280	-0.268169	-0.073765	-0.767470	0.132392	-0.132867	-0.369645	0.519820	female
159285	-1.762966	-2.103024	0.528610	-2.544487	-3.727237	-1.413039	2.287104	0.997904	-1.425704	-0.092397	-0.106547	-0.291907	-0.456864	0.845100	-0.707612	-0.937351	-0.064132	female
715624	-1.500478	-2.434717	0.418023	-2.689296	-3.637624	-1.680580	1.747597	0.844877	-1.608692	-0.756256	-0.105382	0.205703	-1.066054	0.506801	-0.146501	-0.369802	0.281866	female
338427	-2.525414	0.054952	-2.484588	-3.997716	-0.107430	1.950908	0.394009	-3.627412	-0.222841	0.954029	0.082254	-1.991180	-0.894762	-0.646592	0.168910	0.885705	1.421633	female
426824	-1.603953	-2.426892	1.132172	-2.330291	-3.643549	-1.068533	2.323688	0.046560	-2.808129	-1.202936	-0.153376	0.160757	-0.545690	1.332009	-0.042957	-0.704957	0.154987	female

Figure 4.4: A sample of raw data as obtained from the Kaggle dataset.

Similarly, like the previous dataset, labels feature is holding object type values. So, this column was encoded using LabelEncoder. Figure 4.5 represent the dataset after encoding the ‘labels’ column. After encoding, ’male’ was assigned a value of 1 and ’female’ was assigned a value of 0.

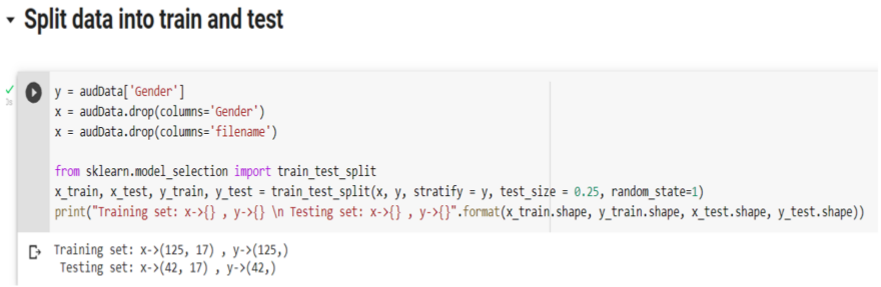


4	55	56	57	labels
5	4.019160	3.692048	3.415411	1
1	2.176067	2.955910	2.600446	1
8	3.816338	3.659878	3.295839	1
9	4.050447	3.740353	3.456593	1
2	4.980473	4.334596	4.759656	1
...	...	...	...	...
2	-0.132867	-0.369645	0.519820	0
0	-0.707612	-0.937351	-0.064132	0
1	-0.146501	-0.369802	0.281866	0
2	0.168910	0.885705	1.421633	0
9	-0.042957	-0.704957	0.154987	0

Figure 4.5: Labels column after encoding Kaglle dataset.

4.1.3 Classification using 9 classifiers

For training MLP algorithm, data was split into train and test data with ratio 0.25. Our target here is to detect correct gender of the speaker. And so, the Gender feature is our target label. For better accuracy rate scaling was implemented using minmax scaler from sklearn library. In the result section, the differences between the results obtained with and without scaling will be compared. The MLP algorithm was implemented on both data sets. Similarly, RBFN, Decision Tree, Random Forest, Gradient Boosting, Naive Bayes, KNN, Logistic Regression and Support Vector Machine was implemented on both data sets.

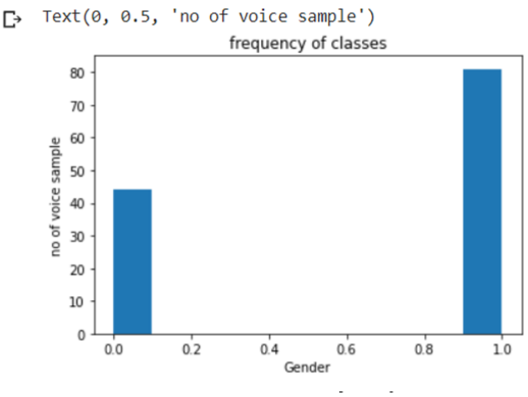


▾ Split data into train and test

```
y = audData['Gender']
x = audData.drop(columns='Gender')
x = audData.drop(columns='filename')

from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, stratify = y, test_size = 0.25, random_state=1)
print("Training set: x->{} , y->{} \n Testing set: x->{} , y->{}".format(x_train.shape, y_train.shape, x_test.shape, y_test.shape))
```

```
Training set: x->(125, 17) , y->(125,)
 Testing set: x->(42, 17) , y->(42,)
```

Figure 4.6: Splitting data into train and test data.



```
Text(0, 0.5, 'no of voice sample')
```

Figure 4.7: Target data plot.

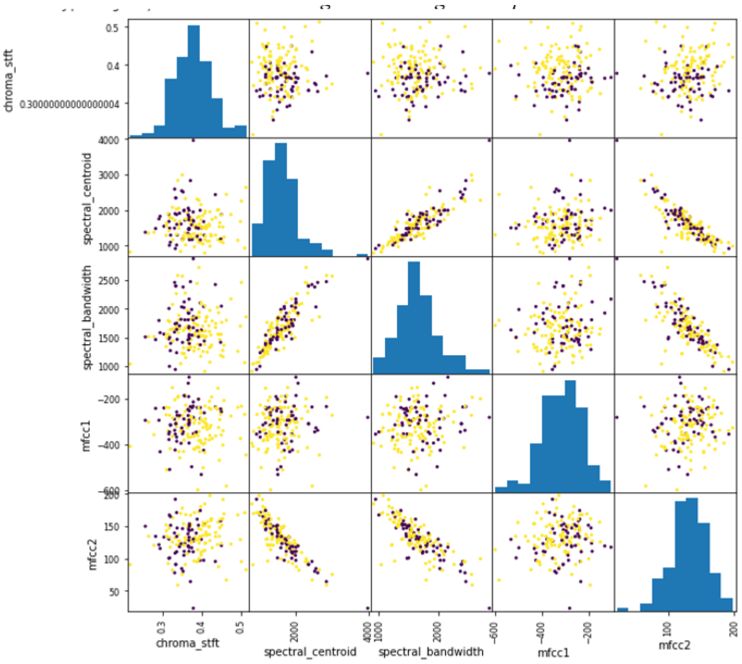
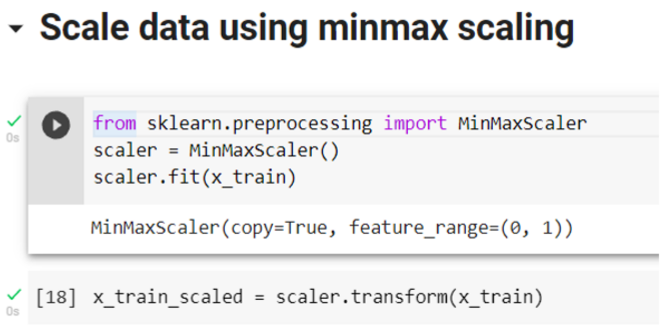


Figure 4.8: Scatter matrix.



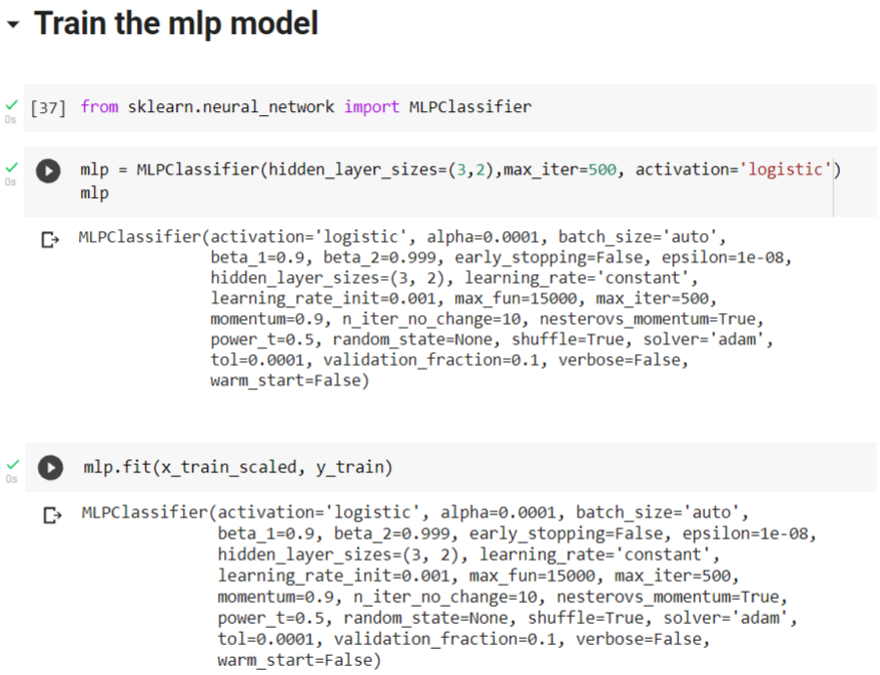
Scale data using minmax scaling

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
scaler.fit(x_train)
```

```
MinMaxScaler(copy=True, feature_range=(0, 1))
```

```
[18] x_train_scaled = scaler.transform(x_train)
```

Figure 4.9: MinMax scaling.



Train the mlp model

```
[37] from sklearn.neural_network import MLPClassifier
```

```
mlp = MLPClassifier(hidden_layer_sizes=(3,2),max_iter=500, activation='logistic')
mlp
```

```
MLPClassifier(activation='logistic', alpha=0.0001, batch_size='auto',
              beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(3, 2), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

```
mlp.fit(x_train_scaled, y_train)
```

```
MLPClassifier(activation='logistic', alpha=0.0001, batch_size='auto',
              beta_1=0.9, beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(3, 2), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Figure 4.10: Training MLP.

Similarly, same procedure was followed for the Kaggle database as can be seen in Figures 4.11, 4.12 and 4.13. MLP classifier was trained similar to the previous database.

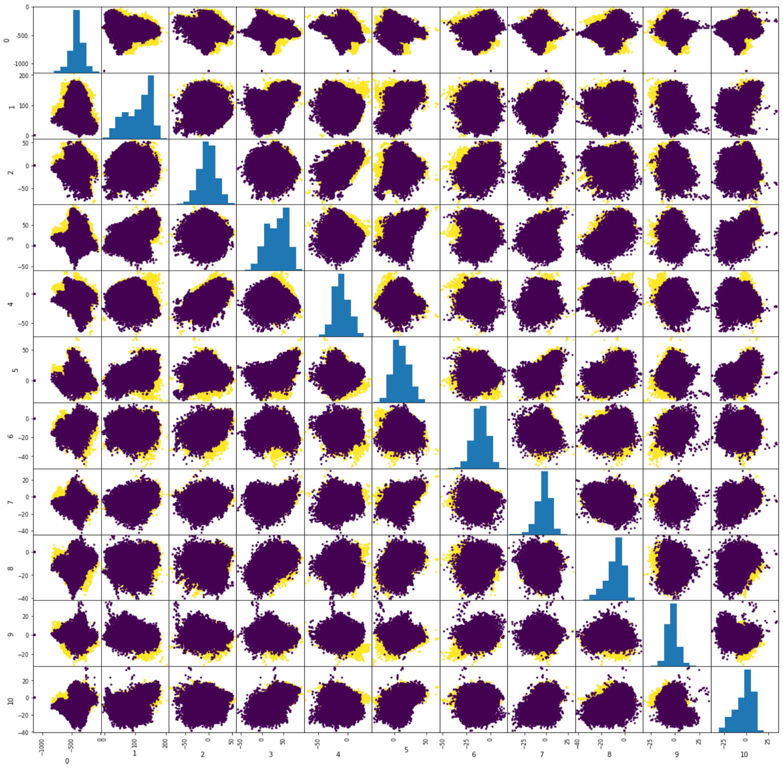


Figure 4.11: Scatter matrix for Kaggle dataset.

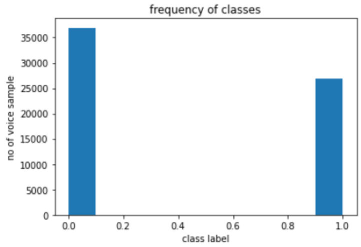
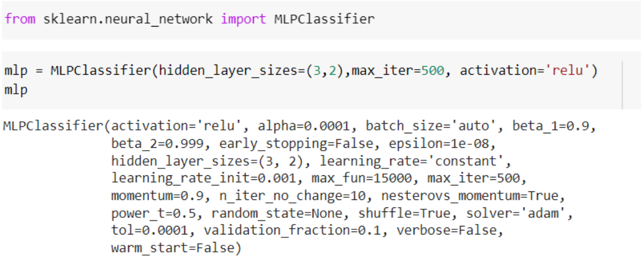


Figure 4.12: Target data plot for Kaggle dataset.



```
from sklearn.neural_network import MLPClassifier
```

```
mlp = MLPClassifier(hidden_layer_sizes=(3,2),max_iter=500, activation='relu')
mlp
```

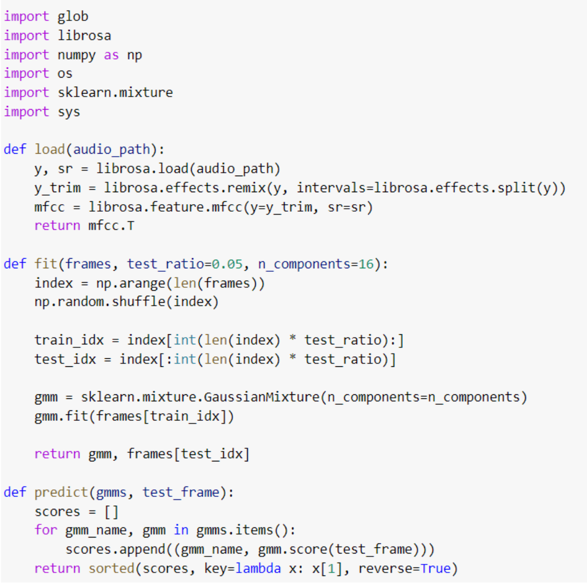
```
MLPClassifier(activation='relu', alpha=0.0001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=(3, 2), learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=None, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Figure 4.13: Training MLP for Kaggle dataset.

Similarly, both data sets were subjected to RBFN, Decision Tree, Random Forest, Gradient Boosting, Naive Bayes, KNN, Logistic Regression, and Support Vector Machine.

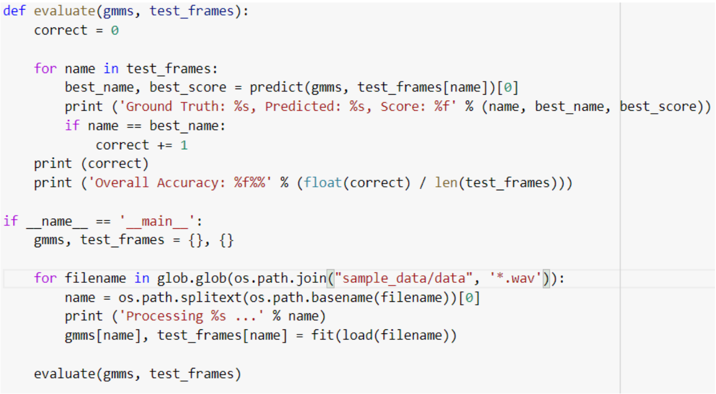
4.1.4 Speaker Identification Using GMM

Based on the result of Gender classification from the best result of multiple classifiers, data was assumed to be separated by gender, and pattern recognition algorithms GMM was used for speaker identification. The algorithm was used on both combined dataset-1 and the same dataset separated by gender. In the following 3 figures implementation of GMM is described.



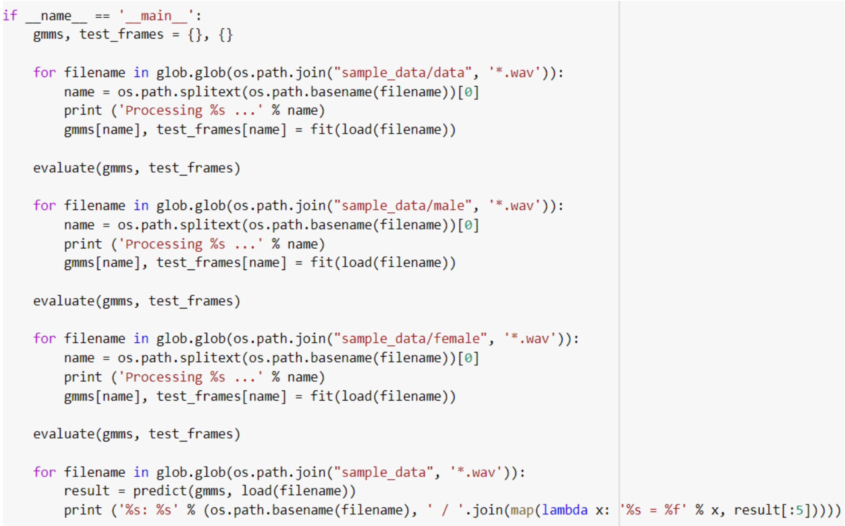
```python
import glob
import librosa
import numpy as np
import os
import sklearn.mixture
import sys

def load(audio_path):
    y, sr = librosa.load(audio_path)
    y_trim = librosa.effects.remix(y, intervals=librosa.effects.split(y))
    mfcc = librosa.feature.mfcc(y=y_trim, sr=sr)
    return mfcc.T

def fit(frames, test_ratio=0.05, n_components=16):
    index = np.arange(len(frames))
    np.random.shuffle(index)

    train_idx = index[int(len(index) * test_ratio):]
    test_idx = index[:int(len(index) * test_ratio)]

    gmm = sklearn.mixture.GaussianMixture(n_components=n_components)
    gmm.fit(frames[train_idx])

    return gmm, frames[test_idx]

def predict(gmms, test_frame):
    scores = []
    for gmm_name, gmm in gmms.items():
        scores.append((gmm_name, gmm.score(test_frame)))
    return sorted(scores, key=lambda x: x[1], reverse=True)
```

Figure 4.14: Implementation of GMM for Dataset1.



```python
def evaluate(gmms, test_frames):
    correct = 0

    for name in test_frames:
        best_name, best_score = predict(gmms, test_frames[name])[0]
        print ('Ground Truth: %s, Predicted: %s, Score: %f' % (name, best_name, best_score))
        if name == best_name:
            correct += 1
    print (correct)
    print ('Overall Accuracy: %f%%' % (float(correct) / len(test_frames)))

if __name__ == '__main__':
    gmms, test_frames = {}, {}

    for filename in glob.glob(os.path.join("sample_data/data", '*.wav')):
        name = os.path.splitext(os.path.basename(filename))[0]
        print ('Processing %s ...' % name)
        gmms[name], test_frames[name] = fit(load(filename))

    evaluate(gmms, test_frames)
```

Figure 4.15: Implementation of GMM for Dataset1.



```python
if __name__ == '__main__':
    gmms, test_frames = {}, {}

    for filename in glob.glob(os.path.join("sample_data/data", '*.wav')):
        name = os.path.splitext(os.path.basename(filename))[0]
        print ('Processing %s ...' % name)
        gmms[name], test_frames[name] = fit(load(filename))

    evaluate(gmms, test_frames)

    for filename in glob.glob(os.path.join("sample_data/male", '*.wav')):
        name = os.path.splitext(os.path.basename(filename))[0]
        print ('Processing %s ...' % name)
        gmms[name], test_frames[name] = fit(load(filename))

    evaluate(gmms, test_frames)

    for filename in glob.glob(os.path.join("sample_data/female", '*.wav')):
        name = os.path.splitext(os.path.basename(filename))[0]
        print ('Processing %s ...' % name)
        gmms[name], test_frames[name] = fit(load(filename))

    evaluate(gmms, test_frames)

    for filename in glob.glob(os.path.join("sample_data", '*.wav')):
        result = predict(gmms, load(filename))
        print ('%s: %s' % (os.path.basename(filename), ' / '.join(map(lambda x: '%s = %f' % x, result[:5]))))
```

Figure 4.16: Implementation of GMM for Dataset1.

4.2 Results

This section holds the result of the implementation of the classifier and identification algorithms. After running the model using Colab, results of classification of data are obtained. Frist of all, from results gained from dataset-1, it can be seen from the result that MLP performed somewhat satisfactory, while RBFN, SVM, Logistic Regression performed poorly, scoring a testing accuracy lower than 40 percent. KNN performed poorly as well. It can be said that RBFN is not suitable for comparatively smaller datasets. However, astounding performance was seen using Decision Tree, Random Forest, Gradient Boosting and Naïve Bayes Classifier. Each of the classifiers gave 100 percent accuracy on both training data and testing data. The dataset-1 consists of 167 individuals.
While considering in mind ways to increase accuracy, analysis of preprocessing was considered. Min-Max scaler was used to scale the data, but since both the datasets contain mainly audio feature, MFCC extracted from audio samples, abandoning scaling in the preprocessing could be beneficial. After running the same classifiers on unscaled data from dataset-1, the result was better than before as seen in table 4.1. RBFN could not produce much better result compared to the result obtained with scaled data, but Logistic Regression doubled its accuracy while KNN also improved. Although MLP did not perform any better in this case, SVM scored a staggering 100 percent accuracy with unscaled data while it was less than 50percent in case of scaled data.

Algorithms	**With Scaling(Accuracy)**		**Without Scaling(Accuracy)**	
	Training	**Testing**	**Training**	**Testing**
Multilayer Perceptron	0.65	0.64	0.65	0.64
Radial Basis Function	0.35	0.36	1.00	0.38
Decision Tree	1.00	1.00	1.00	1.00
Random Forest	1.00	1.00	1.00	1.00
Gradient Boosting	1.00	1.00	1.00	1.00
Naive Bayes	1.00	1.00	1.00	1.00
K-Nearest Neighbor	0.39	0.52	0.81	0.69
Logistic Regression	0.35	0.36	0.82	0.62
Support Vector Machine	0.35	0.36	1.00	1.00

Table 4.1: Analysis of Gender Recognition and Comparison Among 9 Classifiers on Dataset1.

In case of Kaggle dataset, 7 classifiers were used to detect gender among which all of the algorithms failed to impress in accuracy. However, after training them with unscaled data, MLP brought an accuracy greater than 90 percent, Decision Tree and Logistic Regression went over 80 percent, while Naive Bayes also improved. The noticeable performance however, was seen by Random Forest, Gradient Boosting and KNN, scoring 98 percent, 98 percent and 96 percent accuracy on testing data, respectively.

Algorithms	**With Scaling(Accuracy)**		**Without Scaling(Accuracy)**	
	Training	**Testing**	**Training**	**Testing**
Multilayer Perceptron	0.63	0.63	0.92	0.91
Decision Tree	0.52	0.52	0.80	0.80
Random Forest	1.00	0.67	1.00	0.98
Gradient Boosting	1.00	0.67	1.00	0.98
Naive Bayes	0.42	0.42	0.67	0.67
K-Nearest Neighbor	0.60	0.60	0.99	0.96
Logistic Regression	0.58	0.58	0.89	0.89

Table 4.2: Analysis of Gender Recognition and Comparison Among 7 Classifiers on Kaggle Dataset.

We can see the statistics of precision, recall, f1-score of the classifiers on dataset-1 in table 4.3. Table 4.4 represents the statistics with unscaled data.
Table 4.5 & 4.6 holds the statistics in case of Kaggle dataset.
Moreover, the confusion matrix is described in case of both datasets with scaled data as well as unscaled data in table 4.7 and 4.8.
In addition, the AUROC values are calculated and shown in the following 4 tables for both datasets and the ROC plot for the result with unscaled data for both datasets is shown on figure 4.8 and figure 4.9.
After that, GMM was performed on dataset-1 to perform speaker identification. The algorithm was applied on combined subset and subsets separated by gender. We can see the result on table 4.13. The algorithm was performed in case of 5 people, 10 people, 20 people, 30 people and finally on the whole dataset-1.
It can be seen that except in case of 10 people, the algorithm itself works better on separated datasets compared to combined dataset.

Algorithm	Gender	Precision	Recall	f1 Score	Support
MLP	Female	0.00	0.00	0.00	15
	Male	0.64	1.00	0.78	27
RBFN	Female	0.36	1.00	0.53	15
	Male	0.00	0.00	0.00	27
DT	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
RF	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
GB	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
NB	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
KNN	Female	0.43	1.00	0.60	15
	Male	1.00	0.26	0.41	27
LR	Female	0.36	1.00	0.53	15
	Male	0.00	0.00	0.00	27
SVM	Female	0.36	1.00	0.53	15
	Male	0.00	0.00	0.00	27

Table 4.3: Classification Report of Gender Prediction (With Scaling) For Dataset1.

Algorithm	Gender	Precision	Recall	f1 Score	Support
MLP	Female	0.00	0.00	0.00	15
	Male	0.64	1.00	0.78	27
RBFN	Female	0.37	1.00	0.54	15
	Male	1.00	0.04	0.07	27
DT	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
RF	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
GB	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
NB	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27
KNN	Female	0.57	0.53	0.55	15
	Male	0.75	0.78	0.76	27
LR	Female	0.46	0.40	0.43	15
	Male	0.69	0.74	0.71	27
SVM	Female	1.00	1.00	1.00	15
	Male	1.00	1.00	1.00	27

Table 4.4: Classification Report of Gender Prediction (Without Scaling) For Dataset1.

Algorithm	Gender	Precision	Recall	f1 Score	Support
MLP	Female	0.73	0.56	0.63	12306
	Male	0.54	0.72	0.62	8978
DT	Female	0.58	0.60	0.59	12306
	Male	0.43	0.41	0.42	8978
RF	Female	0.67	0.83	0.74	12306
	Male	0.66	0.44	0.52	8978
GB	Female	0.67	0.97	0.79	12306
	Male	0.90	0.34	0.49	8978
NB	Female	0.25	0.00	0.00	12306
	Male	0.42	1.00	0.59	8978
KNN	Female	0.59	0.99	0.74	12306
	Male	0.85	0.06	0.11	8978
LR	Female	0.58	1.00	0.73	12306
	Male	1.00	0.00	0.00	8978

Table 4.5: Classification Report of Gender Prediction (With Scaling) For Kaggle Dataset.

Algorithm	Gender	Precision	Recall	f1 Score	Support
MLP	Female	0.89	0.97	0.93	12306
	Male	0.95	0.84	0.89	8978
DT	Female	0.84	0.81	0.82	12306
	Male	0.75	0.79	0.77	8978
RF	Female	0.98	0.99	0.98	12306
	Male	0.98	0.97	0.98	8978
GB	Female	0.93	0.94	0.93	12306
	Male	0.91	0.90	0.91	8978
NB	Female	0.88	0.51	0.64	12306
	Male	0.57	0.90	0.70	8978
KNN	Female	0.97	0.95	0.96	12306
	Male	0.93	0.96	0.95	8978
LR	Female	0.91	0.91	0.91	12306
	Male	0.87	0.87	0.87	8978

Table 4.6: Classification Report of Gender Prediction (Without Scaling) For Kaggle Dataset.

Algorithm	With Scaling		Without Scaling	
	Confusion	Matrix	Confusion	Matrix
MLP	0	15	0	15
	0	27	0	27
RBFN	15	0	15	0
	27	0	26	1
DT	15	0	15	0
	0	27	0	27
RF	15	0	15	0
	0	27	0	27
GB	15	0	15	0
	0	27	0	27
NB	15	0	15	0
	0	27	0	27
KNN	15	0	8	7
	20	7	6	21
LR	15	0	6	9
	27	0	7	20
SVM	15	0	15	0
	27	0	0	27

Table 4.7: Confusion Matrix For Gender Prediction For Dataset1.

Algorithm	With Scaling		Without Scaling	
	Confusion	Matrix	Confusion	Matrix
MLP	6.8e+03	5.5e+03	1.2e+04	4.3e+02
	2.5e+03	6.5e+03	1.4e+03	7.5e+03
DT	7.4e+03	4.9e+03	1e+04	2.4e+03
	5.3e+03	3.7e+03	1.9e+03	7.1e+03
RF	1e+04	2.1e+03	1.2e+04	1.8e+02
	5.1e+03	3.9e+03	2.6e+02	8.7e+03
GB	1.2e+04	3.4e+02	1.2e+04	7.7e+02
	6e+03	3e+03	8.7e+02	8.1e+03
NB	1	1.2e+04	6.2e+03	6.1e+03
	3	9e+03	8.9e+02	8.1e+03
KNN	1.2e+04	94	1.2e+04	6.1e+02
	8.5e+03	5.2e+02	3.3e+02	8.6e+03
LR	1.2e+04	0	1.1e+04	1.1e+03
	9e+03	2	1.2e+03	7.8e+03

Table 4.8: Confusion Matrix For Gender Prediction For Kaggle Dataset.

Predictions	**AUROC Values**
Random Forest	1.00
Naive Bayes	1.00
Decision Tree	1.00
Gradient Boosting	1.00
K-Nearest Neighbors	0.63
Logistic Regression	0.619
Support Vector Machine	0.87
Random Prediction	**0.5**

Table 4.9: AUROC Values For The Classifiers For Dataset1 (With Scaling).

Predictions	**AUROC Values**
Random Forest	1.00
Naive Bayes	1.00
Decision Tree	1.00
Gradient Boosting	1.00
K-Nearest Neighbors	0.641
Logistic Regression	0.677
Support Vector Machine	1.00
Random Prediction	**0.5**

Table 4.10: AUROC Values For The Classifiers For Dataset1 (Without Scaling).

Predictions	**AUROC Values**
Random Forest	0.761
Naive Bayes	0.5
Decision Tree	0.556
Gradient Boosting	0.841
K-Nearest Neighbors	0.530
Logistic Regression	0.616
Random Prediction	**0.5**

Table 4.11: AUROC Values For The Classifiers For Kaggle Dataset (With Scaling).

Predictions	**AUROC Values**
Random Forest	0.998
Naive Bayes	0.813
Decision Tree	0.875
Gradient Boosting	0.978
K-Nearest Neighbors	0.989
Logistic Regression	0.959
Random Prediction	**0.5**

Table 4.12: AUROC Values For The Classifiers For Kaggle Dataset (Without Scaling).

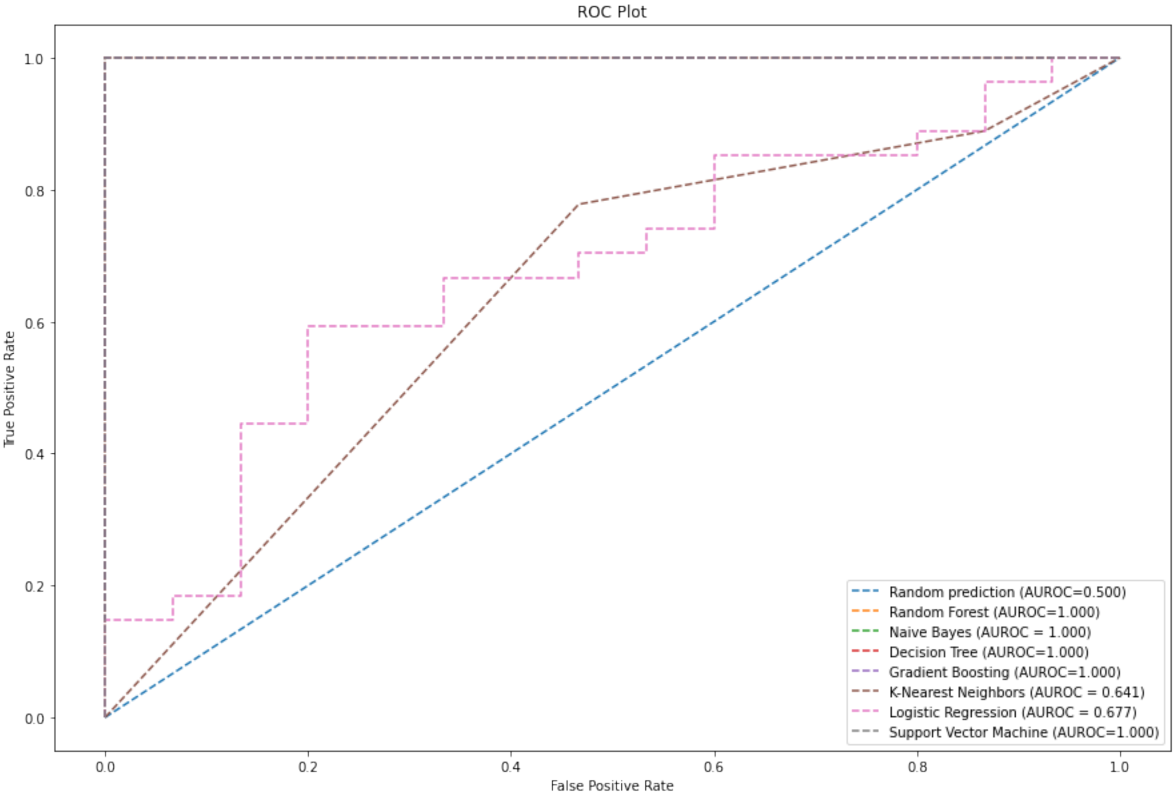


Figure 4.17: ROC Plot For Dataset1 (Without Scaling).

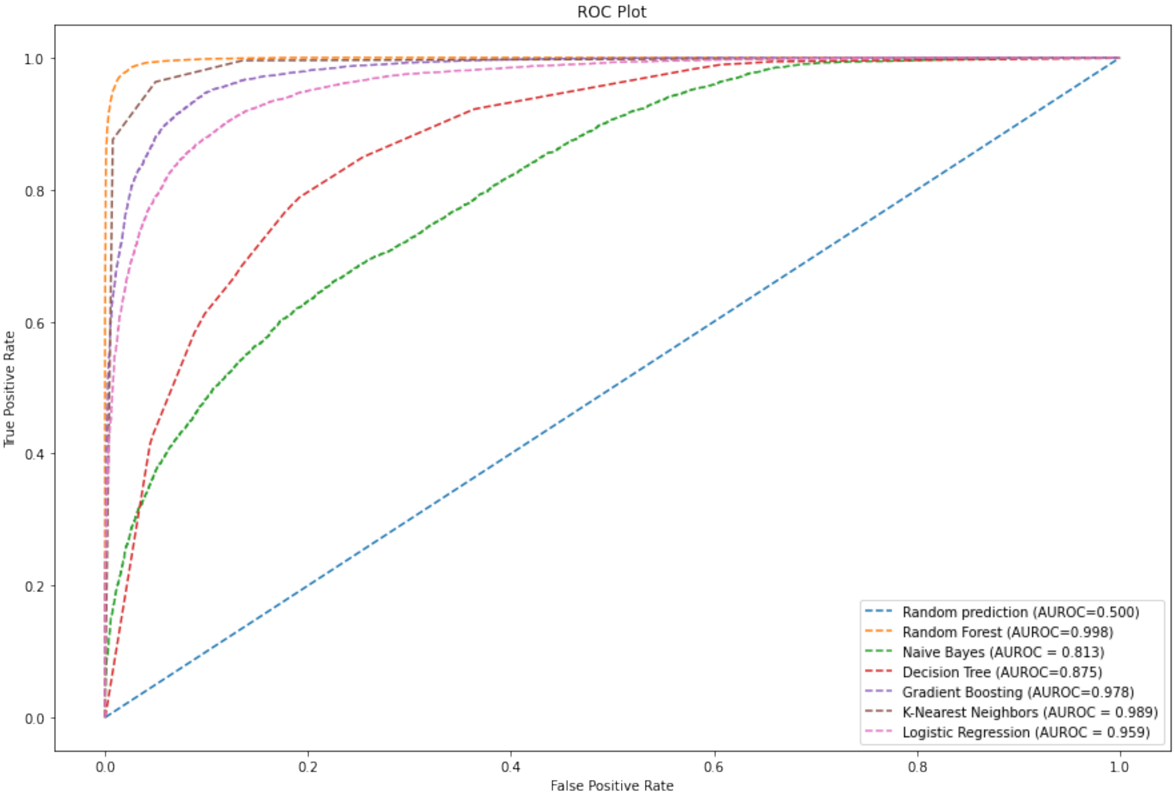


Figure 4.18: ROC Plot For Kaggle Dataset (Without Scaling).

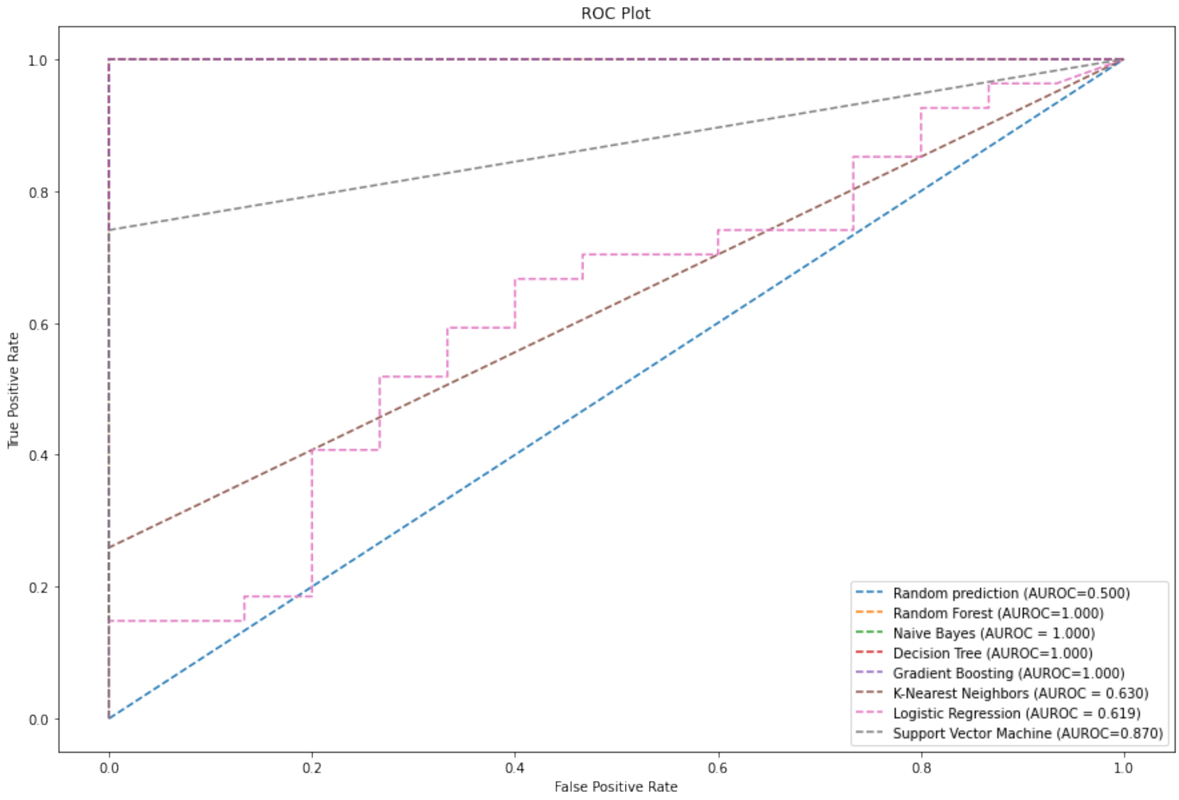


Figure 4.19: ROC Plot For Dataset1 (With Scaling).

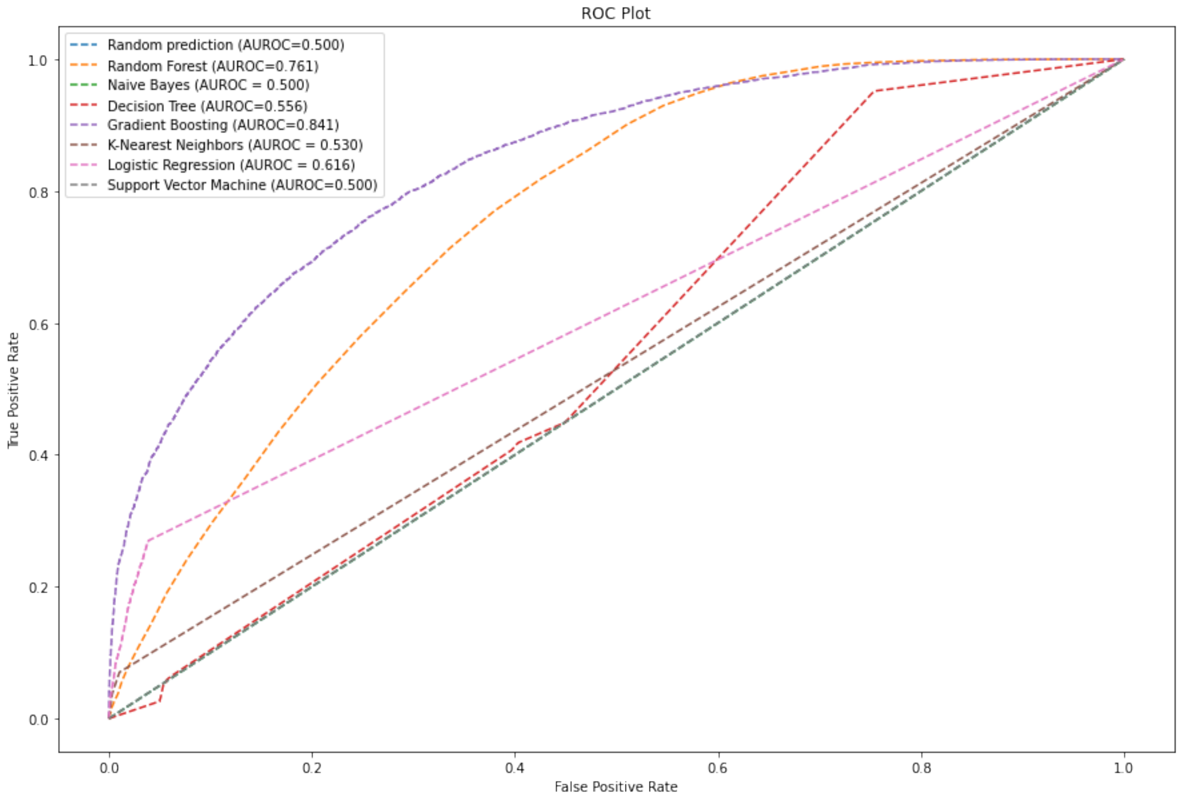


Figure 4.20: ROC Plot For Kaggle Dataset (With Scaling).

Number of People	Combined	Male	Female
5	40%	40%	60%
10	70%	40%	50%
20	38%	40%	47%
30	50%	56%	50%
167 Male and Female	30%	27%	36%

Table 4.13: Speaker identification using GMM for subsets of individuals from Dataset1.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

Person detection on a large scale using voice recognition from a pre-collected database is used for strengthening security system to minimize the risk of compromising the security system. Despite having multiple bio-metric security systems such as fingerprint, retina sensor and so on, voice recognition is prioritized here because of it's easier input availability when comparing with the other systems as it is easier to get the voice of someone for example during a crime scene than fingerprint or other components.

In this paper the goal was to detect gender as accurately as possible and based on the result implement speaker identification on database separated by gender. Some classifiers worked incredibly well in this case. There is shortage of research on using person detection using voice recognition using a central database on large scale scenarios for example, national security of a country or a company and this research aims to approach this area.

5.2 Future Work

Various tests and experiment have been left for the future for lack of resources. In this paper the accuracy of gender prediction was increased. Furthermore, We'd like to improve speaker identification accuracy by implementing stronger pattern matching algorithms. In addition, improving the function of running algorithms on larger datasets will be needed. There is also scope for structuring the model in such a way that in case of wrong gender prediction the model will search the database containing the other gender, which will be the right gender, and execute speaker identification again.

Bibliography

[1] S. Agrawal, A. Shruti, and C. R. Krishna, “Prosodic feature based text dependent speaker recognition using machine learning algorithms,” *International Journal of Engineering Science and Technology*, vol. 2, no. 10, pp. 5150–5157, 2010.

[2] L. Muda, M. Begam, and I. Elamvazuthi, “Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques,” *arXiv preprint arXiv:1003.4083*, 2010.

[3] J. W. Shin, J.-H. Chang, and N. S. Kim, “Voice activity detection based on statistical models and machine learning approaches,” *Computer Speech & Language*, vol. 24, no. 3, pp. 515–530, 2010.

[4] J. XinXing and S. Xu, “Speech recognition based on efficient dtw algorithm and its dsp implementation,” *Procedia Engineering*, vol. 29, pp. 832–836, 2012.

[5] A. M. Moselhy and A. A. Abdelnaiem, “Lpc and mfcc performance evaluation with artificial neural network for spoken language identification,” *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 6, no. 3, p. 55, 2013.

[6] A. Halageri, A. Bidappa, C. Arjun, M. M. Sarathy, and S. Sultana, “Speech recognition using deep learning,” *Int. J. Comput. Sci. Inf. Technol*, vol. 6, no. 3, pp. 3206–3209, 2015.

[7] J. Padmanabhan and M. J. Johnson Premkumar, “Machine learning in automatic speech recognition: A survey,” *IETE Technical Review*, vol. 32, no. 4, pp. 240–251, 2015.

[8] M. V. K. Kale, P. D. Deshmukh, and M. H. R. Gite, “Voice based biometric system feature extraction using mfcc and lpc technique,” *International Journal of Advanced Engineering Research and Science*, vol. 3, no. 5, p. 236 709, 2016.

[9] B. Soewito, F. L. Gaol, E. Simanjuntak, and F. E. Gunawan, “Smart mobile attendance system using voice recognition and fingerprint on smartphone,” in *2016 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, IEEE, 2016, pp. 175–180.

[10] S. Kinkiri, W. J. Melis, and S. Keates, "Machine learning for voice recognition," 2017.

[11] P. Gupta, S. Goel, and A. Purwar, "A stacked technique for gender recognition through voice," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, IEEE, 2018, pp. 1–3.

[12] N. S. Ibrahim and D. A. Ramli, "I-vector extraction for speaker recognition based on dimensionality reduction," *Procedia Computer Science*, vol. 126, pp. 1534–1540, 2018.

[13] O. Novotnỳ, O. Plchot, P. Matejka, L. Mosner, and O. Glembek, "On the use of x-vectors for robust speaker recognition.," in *Odyssey*, 2018, pp. 168–175.

[14] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2018, pp. 5329–5333.

[15] R. Thiruvengatanadhan, "Speech recognition using svm," *International Research Journal of Engineering and Technology (IRJET)*, vol. 5, no. 9, pp. 918–921, 2018.

[16] L. Verde, G. De Pietro, and G. Sannino, "Voice disorder identification by using machine learning techniques," *IEEE access*, vol. 6, pp. 16 246–16 255, 2018.

[17] A. F. Fadlilah and E. C. Djamal, "Speaker and speech recognition using hierarchy support vector machine and backpropagation," in *2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, 2019, pp. 404–409.

[18] F. Fang, X. Wang, J. Yamagishi, I. Echizen, M. Todisco, N. Evans, and J.-F. Bonastre, "Speaker anonymization using x-vector and neural waveform models," *arXiv preprint arXiv:1905.13561*, 2019.

[19] N. H. Tandel, H. B. Prajapati, and V. K. Dabhi, "Voice recognition and voice comparison using machine learning techniques: A survey," in *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, IEEE, 2020, pp. 459–465.

[20] *Noise*, Dec. 2021. [Online]. Available: https://en.wikipedia.org/wiki/Noise.

[21] M. Orken, K. Aizat, A. Keylan, O. Dina, Z. Bagashar, and N. Bulbul, "Development of security systems using dnn and i & x-vector classifiers," *Mamyrbayev, O., Kydyrbekova, A., Alimhan, K., Oralbekova, D., Zhumazhanov, B., & Nuranbayeva, B.(2021). Development of security systems using DNN and i*

& x-vector classifiers. Eastern-European Journal of Enterprise Technologies, vol. 4, no. 9, p. 112, 2021.

[22] S. E. Tandogan and H. T. Sencar, “Estimating uniqueness of i-vector-based representation of human voice,” *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 3054–3067, 2021.

[23] *A brief history of speech recognition.* [Online]. Available: https://sonix.ai/history-of-speech-recognition#:~:text=The%20first%20speech%20recognitio%20n%20systems,to%2016%20words%20in%20English..

[24] T. Kathiresan, A. Verma, and V. Dellwo, “Gender bias in voice recognition: An i-vector-based gender-specific automatic speaker recognition study,”

[25] *Machine learning.* [Online]. Available: https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm.

[26] *Scikit.* [Online]. Available: https://scikit-learn.org/stable/modules/svm.html.

[27] Tahira, *Speaker-identification-using-gmm-with-mfcc.* [Online]. Available: https://www.researchgate.net/publication/274963749_Speaker_Identification_Using_GMM_with_MFCC.