VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangama", Belagavi, Karnataka, INDIA

A Project Report
On

Gender Identification by Voice

Submitted in partial fulfillment of the requirement for the award of the degree of

Bachelor of Engineering
in
Computer Science and Engineering

Submitted By

SANJAY TJ	**1GA15CS182**
SHIVAM UPADHYAY	**1GA16CS136**
TARUN M	**1GA16CS163**
YASHWANTHGOWDA HR	**1GA16CS184**

Under the Guidance of

Dr. BHAGYASHRI R HANJI
Assistant Professor
Department of CSE

Department of Computer Science and Engineering
(Accredited by NBA 2019-2022)
GLOBAL ACADEMY OF TECHNOLOGY
Rajarajeshwarinagar, Bengaluru - 560 098

2019 – 2020

GLOBAL ACADEMY OF TECHNOLOGY

Department of Computer Science and Engineering

CERTIFICATE

Certified that the Project Entitled **"Gender Identification by Voice"** carried out by **SANJAY TJ**, bearing **USN 1GA15CS182**, **SHIVAM UPADHYAY**, bearing **USN 1GA16CS136**, **TARUN M**, bearing **USN 1GA16CS163** and **YASHWANTHGOWDA HR**, bearing **USN 1GA16CS184**, bonafide students of Global Academy of Technology, are in partial fulfillment for the award of the **BACHELOR OF ENGINEERING** in Computer Science and Engineering from **Visvesvaraya Technological University, Belagavi** during the year 2019-2020. It is certified that all the corrections/suggestions indicated for Internal Assessment have been incorporated in the report submitted in the department library. The Project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said Degree.

____________________	____________________	____________________
Dr. Bhagyashri R Hanji Assistant Professor Dept. of CSE GAT, Bengaluru.	Dr. N Guruprasad Professor & HOD Dept. of CSE GAT, Bengaluru.	Dr. Rana Pratap Reddy Principal GAT, Bengaluru.

Name of the Examiners	Signature with date
1. ____________________	____________________
2. ____________________	____________________

GLOBAL ACADEMY OF TECHNOLOGY

Rajarajeshwarinagar, Bengaluru – 560 098

DECLARATION

We, **SANJAY TJ**, bearing **USN 1GA15CS182**, **SHIVAM UPADHYAY**, bearing **USN 1GA16CS136**, **TARUN M**, bearing **USN 1GA16CS163** and **YASHWANTHGOWDA HR**, bearing **USN 1GA16CS184**, students of Eighth Semester B.E, Department of Computer Science and Engineering, Global Academy of Technology, Rajarajeshwarinagar Bengaluru, declare that the Project Work entitled "**Gender Identification by Voice**" has been carried out by us and submitted in partial fulfillment of the course requirements for the award of degree in **Bachelor of Engineering** in **Computer Science and Engineering** from **Visvesvaraya Technological University, Belagavi** during the academic year **2019-2020**. The matter embodied in this report has not been submitted to any other university or institution for the award of any other degree.

1. SANJAY TJ	**1GA15CS182**
2. SHIVAM UPADHYAY	**1GA16CS136**
3. TARUN M	**1GA16CS163**
4. YASHWANTHGOWDA HR	**1GA16CS184**

Place: Bengaluru

Date: 27-05-2020

ABSTRACT

Machine learning is the science of getting computers to act without being explicitly programmed. Many researchers also think it is the best way to make progress towards human-level AI. Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs. Any technology user today has benefitted from machine learning.

This project is designed to predict the gender of a speaker from their voice. It has a variety of applications ranging from voice analytics to personalizing human machine interactions. We have built a learning system to identify the gender of a speaker using various machine learning algorithms. The gender bias in case of voice sample containing sounds like crying or yelling or any other emotions is the drawback of the existing system.

We have designed a dataset such that, it includes voices of either gender in all kind of emotional reactions. And built a learning model based on the dataset, and have trained it using various machine learning algorithms. As a result, we have created a web interface where users can record and upload their voice samples for prediction and have displayed the results in each of the learning models used. The overall prediction is based on the result which is most consistent with models.

From our project we have observed that Random forest decision tree and Support Vector Machine predict the results with best accuracy scores.

ACKNOWLEDGEMENT

The satisfaction and the euphoria that accompany the successful completion of any task would be incomplete without the mention of the people who made it possible. The constant guidance of these persons and encouragement provide, crowned our efforts with success and glory. Although it is not possible to thank all the members who helped for the completion of the project individually, we take this opportunity to express our gratitude to one and all.

We are grateful to management and our institute **GLOBAL ACADEMY OF TECHNOLOGY** with its very ideals and inspiration for having provided us with the facilities, which made this, project a success.

We express our sincere gratitude to **Dr. N. Rana Pratap Reddy,** Principal, Global Academy of Technology for the support and encouragement.

We wish to place on record, our grateful thanks to **Dr. N Guruprasad**, HOD, Department of CSE, Global Academy of Technology, for the constant encouragement provided to us.

We are indebted with a deep sense of gratitude for the constant inspiration, encouragement, timely guidance and valid suggestion given to us by our guide **Dr. Bhagyashri R Hanji**, Assistant Professor, Department of CSE, Global Academy of Technology.

We are thankful to all the staff members of the department for providing relevant information and helped in different capacities in carrying out this project.

Last, but not least, we owe our debts to our parents, friends and also those who directly or indirectly have helped us to make the project work a success.

SANJAY TJ **1GA15CS182**
SHIVAM UPADHYAY **1GA16CS136**
TARUN M **1GA16CS163**
YASHWANTHGOWDA HR **1GA16CS184**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

SRS	Software Requirement Specification
UML	Unified Modeling Language
DFD	Dataflow Diagram
AI	Artificial Intelligence
MP3	Moving Pictures Expert Group Audio Layer 3
SVM	Support Vector Machine
kNN	k-Nearest Neighbors
GMM	Gaussian Mixture Model
DA	Discriminant Analysis
CT	Classification Tree
MLP	Multi-Layer Perceptron
MFCC	Mel-Frequency Cepstral Co-efficient
LDA	Linear Discriminant Analysis
CART	Classification and Regression Trees
HTML	Hypertext Markup Language
CSV	Comma-Separated Values
WarbleR	A package designed to streamline analysis of animal acoustic signals in R

Chapter 1

INTRODUCTION

Machine learning is the science of getting computers to act without being explicitly programmed. Recently, Machine learning has given us self-driving cars, practical speech recognition, effective web search, and a vastly improved understanding of the human genome. Machine learning is so pervasive today that you probably use it dozens of times a day without knowing it. Many researchers also think it is the best way to make progress towards human-level AI.

Machine learning algorithms instead allow for computers to train on data inputs and use statistical analysis in order to output values that fall within a specific range. Because of this, machine learning facilitates computers in building models from sample data in order to automate decision-making processes based on data inputs.

Any technology user today has benefitted from machine learning. Facial recognition technology allows social media platforms to help users tag and share photos of friends. Recommendation engines, powered by machine learning, suggest what movies or television shows to watch next based on user preferences. Speech recognition is the ability of a machine or program to identify words and phrases in spoken language and convert them to a machine-readable format.

1.1 Definitions

Machine Learning: It is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Gender Identity: It is defined as a personal conception of oneself as male or female (or rarely others). This concept is intimately related to the concept of gender role, which is defined as the outward manifestations of personality that reflect the gender identity. Gender identity, in nearly all instances, is self-identified, as a result of a combination of inherent and extrinsic or environmental factors.

Audio Features: An abstract representation of pieces of digital music. Audio features are computed from the raw audio signal. Simple features are the number of zero crossings of the audio signal or its centroid. More sophisticated approaches, such as MP3-based features, rhythm Patterns, Rhythm Histograms, or statistical Spectrum Descriptors take into account, for instance, findings from psycho-acoustics.

Naïve Bayes Classifier: These are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naïve) independence assumptions between the features. They are among the simplest Bayesian network models.

Support Vector Machine (SVM): It is a machine learning algorithm that analyzes data for classification and regression analysis. SVM is a supervised learning method that looks at data and sorts it into one of two categories. An SVM outputs a map of the sorted data with the margins between the two as far apart as possible.

k-Nearest-Neighbor Classifier (kNN): It is an approach to data classification that estimates how likely a data point is to be a member of one group or the other depending on what group the data points nearest to it are in.

Random Forest Classifiers or Random Decision Forests: These are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

WarbleR: It is a package designed to streamline analysis of animal acoustic signals in R. This package allows users to collect open-access avian vocalizations data or input their own data into a workflow that facilitates spectrographic visualization and measurement of acoustic parameters. warbleR makes fundamental sound analysis tools from the R package seewave, as well as new tools not yet offered in the R environment, readily available for batch process analysis.

1.2 Project Report Outline

Chapter 1: Introduction

A brief introduction about the project, project related definitions that is machine learning, gender identification, audio features and various algorithms to be used in this project.

Chapter 2: Review of Literature

This chapter provides a view of the literature survey which include the system study, proposed work, problem statement, existing system, proposed system and scope of the project.

Chapter 3: System Requirement Specification

This chapter provides information on the various functional, non-functional requirements, hardware and software requirements to facilitate the system.

Chapter 4: System Design

This chapter provides a view of the proposed system which include system architecture, different levels of data flow diagrams, the various UML diagrams and the system modules.

Chapter 5: Implementation

This chapter provides the process of converting a new system design into an operational one. Implementation is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. It covers the steps for implementation, the issues in implementation and the various algorithms incorporated.

Chapter 6: Testing

This chapter gives the outline of all testing methods that are carried out to get a bug free system. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

Chapter 7: Results

This section describes the result of the project. The includes the snapshots of each module, the web interface designed for the system and finally the accuracies of each learning model.

Chapter 8: Conclusion

This chapter provides the conclusion, which is the sum up of the project, it is used to introduce some final comments at the project. A foregone conclusion is an outcome that seems certain.

Chapter 2

REVIEW OF LITERATURE

2.1 System Study

The authors [8] used a set of Neural Networks with Acoustic and Pitch related features. They investigated Gaussian Mixture Models (GMM), Multi-Layer Perceptron (MLP), and Decision Tree Classifiers. A combination of Piecewise Gaussian Modelling features and pitch-related features with a set of Neural Networks was shown to perform better than any individual classifier.

The author [9] utilized Yaafe to extract audio features out of sentence recordings. He trained on Naïve Bayes, Discriminant Analysis (DA), Support Vector Machine (SVM) with Linear kernel, k-Nearest Neighbor (kNN) and Classification Tree (CT) classifiers. The DA classifier is most performant in terms of test error rate and precision. The high bias problem implies the set of all available features he is considering does not capture enough gender specific characteristics of voice.

The authors [10] trained four classifiers: Logistic Regression Linear Regression Random Forest and AdaBoost available from Python's scikit-learn library, on f0 summary statistics, Mel frequency cepstral coefficients (MFCC) summary statistics, and using a combination f0 and MFCC statistics. Random forest yields the best model, achieving an accuracy of 85.0% when using all features,83.3% with only f0 features, and 84.1% when using the f0 + energy + voice quality feature set.

The authors [11] used a Support Vector Machine (SVM) based gender identification is applied on discriminative weight training. The performances of gender classification system have been evaluated on the conditions of clean speech, with gender classification accuracy of at most 98% and remains 95% for most noisy speech.

The authors [12] trained model using Support Vector Machine (SVM), Decision Trees, Gradient Tree Boosting, Random forests and accuracy is calculated. Some unexpected behavior is in the peaks at very low frequencies (<50 Hz). This can be due to the presence of noise in the audio recordings. The Random Forest gave the best accuracy followed by the Gradient Boosting SVM and Decision Tree.

The authors [13] used five different algorithms. They are Linear Discriminant Analysis (LDA), k-Nearest Neighbor (kNN), Classification and Regression Trees (CART), Random Forest and Support Vector Machine (SVM) on basis of eight different metrics. The result shows that SVM algorithm performs better in classification and with reduced error rate.

2.2 Proposed Work

2.2.1 Problem Statement

Identifying the gender of a speaker from voice using various machine learning algorithms and overcome the gender bias in case of voice sample containing sounds like crying or yelling which is the drawback of the existing system.

2.2.2 Existing System

- Algorithms used for classifying:
  - Linear Discriminate Analysis
  - K-Nearest Neighbour
  - Classification and Regression Trees
  - Random Forest
  - Support Vector Machine
  - Naïve Bayes
  - Gradient Tree Boosting and Mel-frequency cepstral coefficients summary statistics to provide the output.

2.2.3 Proposed System

- To create a dataset such that, it includes voices of either gender in all kind of emotional reactions.
- To create a learning model based on the dataset, and train it using various machine learning algorithms.
- To create a web interface where user can upload their voice samples.
- As a result, to display the prediction with the results in each of the learning models used.

2.3 Scope of the project

Gender identification by voice is useful in speech-based recognition systems which employ gender-dependent models. Gender differentiation help to improve automatic emotion recognition from speech. Classifying speaker’s gender is an important task in the context of multimedia indexing.

Gender identification can improve the prediction of other speaker traits such as age and emotion, either by jointly modelling gender with age (or emotion) or in a pipelined manner. Automatic gender detection also useful in some cases of a mobile healthcare system i.e., there are some pathologies, such as vocal fold cyst. For detecting feeling like male sad, female anger, etc. Differentiating audios and videos using tags. Spontaneous salutations. Helping personal assistants to answer questions with gender-specific results etc.

Chapter 3

SYSTEM REQUIREMENTS SPECIFICATION

3.1 Functional Requirements

The functional requirements for a system describe what the system should do. These requirements depend on the type of software being developed. The general approach taken by the organization when writing requirements. The functional system requirements describe the system function in detail, its inputs and outputs, exceptions and so on.

The Functional requirements are as follows:

- Develop a robust system which can predict the gender of a recorded voice sample.
- Train the model against enough variety in dataset to overcome gender bias.
- Perform data pre-processing on the dataset to increase the accuracy of the prediction.
- Provide web interface features for the users to get indicate the accuracies on various models.

3.2 Non-Functional Requirements

Non-functional requirements, as the name suggests, are requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability, response time and store occupancy. Alternatively, they may define constraints on the system such as capabilities of I/O devices and the data representations used in system interfaces.

The non-functional requirements are as follows:

- Make the system in such a way that after uploading an audio file, it takes minimum amount of time to processes and display the result.
- Record the voice sample from the microphone in a confined space and as quite as possible to obtain best prediction.

- The algorithm should never fail in any of the test cases.

3.3 Hardware Requirements

Table 3.3 Hardware requirements

Processor	Intel i5 7^{th} gen or above
RAM	8 GB
Hard Disk	256 GB
Peripherals	Monitor, Keyboard, Mouse, Microphone, Speaker

3.4 Software Requirements

Table 3.4 Software requirements

Operating System	Windows 10 (64bit)
Development Software	Anaconda Jupyter Notebook, R Studio, Visual Studio and Wamp Server
Web Browser	anything which supports HTML 5
Programming Environment	Python version 3.8.2 R version 3.6.3 Flask version 1.1.2

Chapter 4

SYSTEM DESIGN

4.1 Design Overview

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could see it as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture, and systems engineering. System design is one of the most important phases of software development process. The purpose of the design is to plan the solution of a problem specified by the requirement documentation.

In other words, the first step in the solution to the problem is the design of the project. The design of the system is perhaps the most critical factor affecting the quality of the software. The objective of the design phase is to produce overall design of the software. It aims to figure out the modules that should be in the system to fulfill all the system requirements in an efficient manner.

The design will contain the specification of all these modules, their interaction with other modules and the desired output from each module. The output of the design process is a description of the software architecture.

The design phase is followed by two sub phases:

- High Level Design
- Detailed Level Design

4.2 System Architecture

System architecture include the representation, connection and arrangement of components that are used in the project. The same is shown in the Fig 4.2.

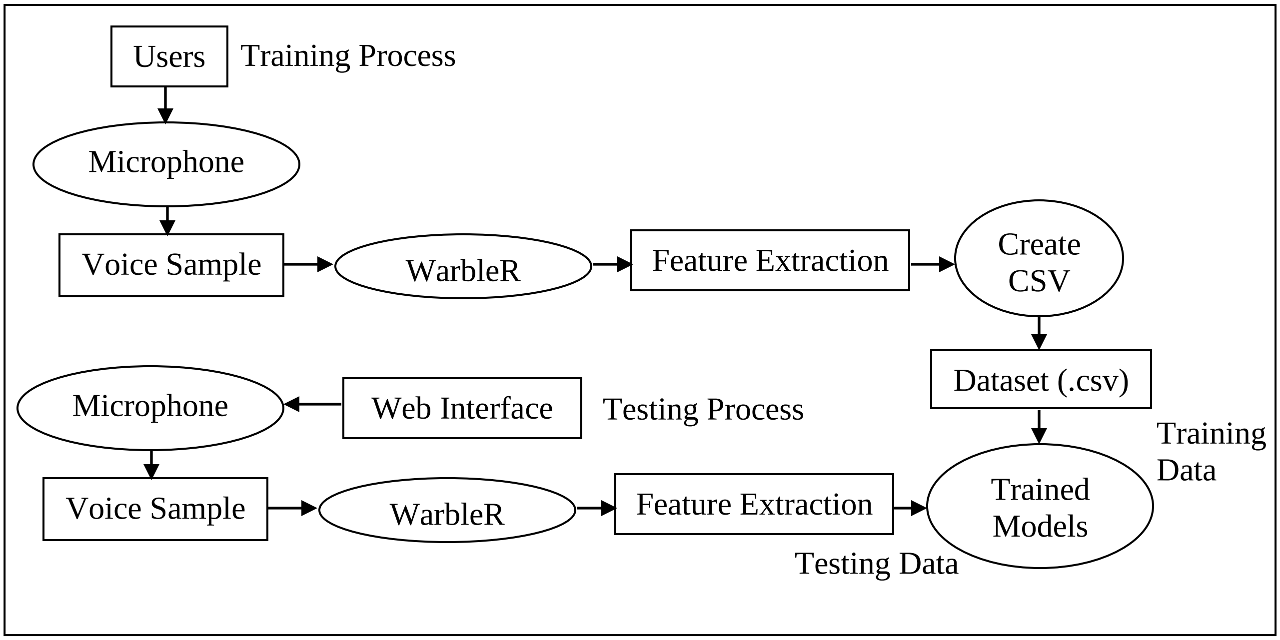


Fig 4.2 System Architecture

4.3 Data Flow Diagrams

A DFD is a logical model of the system. The model does not depend on the hardware software and data-structures of file organization. It tends to be easy for a given non-technical users to understand and thus serves as an excellent communication tool. A DFD is a logical model of the system. The model does not depend on the hardware software and data-structures of file organization. It tends to be easy for a given non-technical users to understand and thus serves as an excellent communication tool.

4.3.1 Data Flow Diagram - Level 0

In level 0 of data flow diagram, the voice features are extracted using R programming library WarbleR and are stored in the dataset (.csv). The same is shown in the Fig 4.3.1.

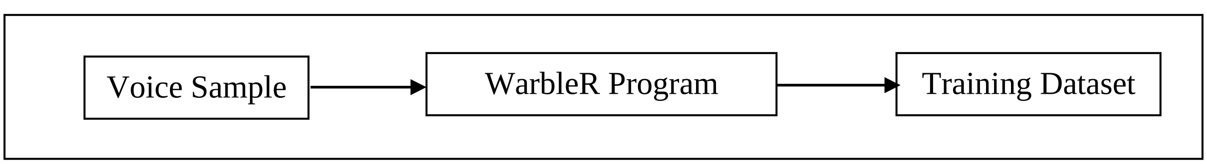


Fig 4.3.1 DFD-Level 0

4.3.2 Data Flow Diagram - Level 1

In level 1 of data flow diagram, the dataset is trained to classify the testing voice sample either male or female using machine learning algorithms. The same is shown in the Fig 4.3.2.

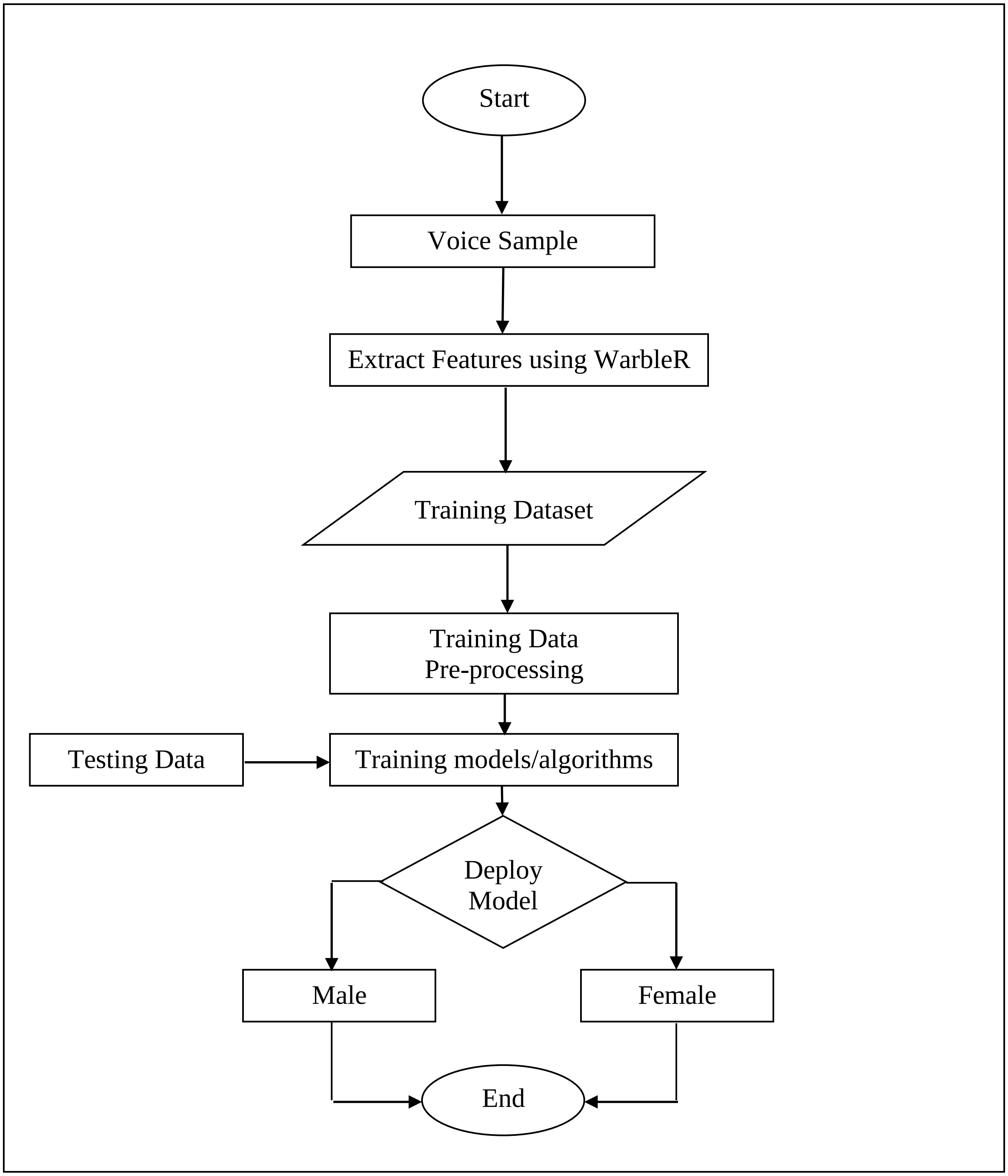


Fig 4.3.2 DFD-Level 1

4.3.3 Data Flow Diagram - Level 2

In level 2 of data flow diagram, the dataset is trained using Naïve Bayes classifier, Support Vector Machine, k-Nearest Neighbors classifier and Random Forest classifier to predict the given sample as either male or female. The same is shown in the Fig 4.3.3.

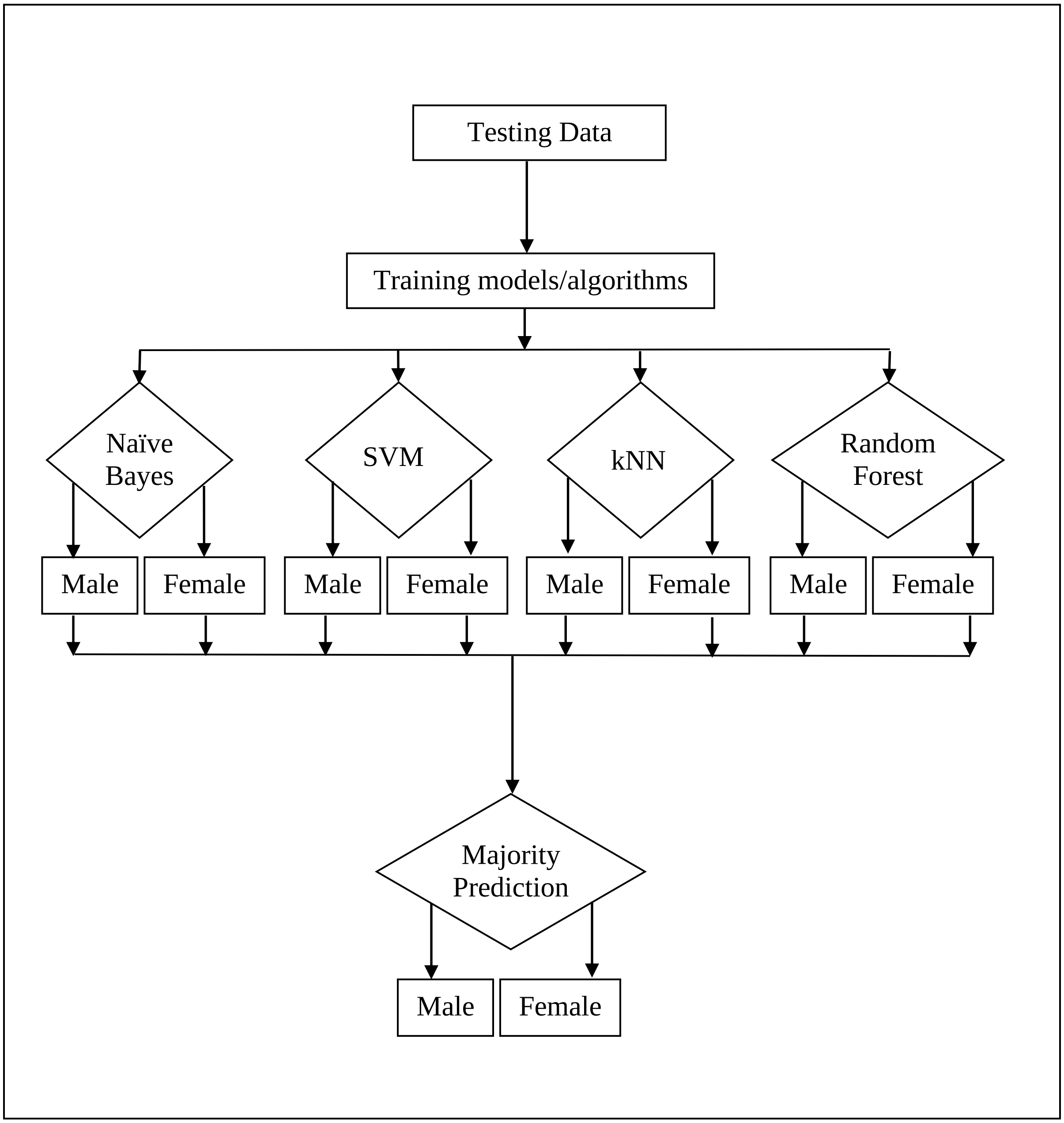


Fig 4.3.3 DFD-Level 2

4.4 Use Case Diagrams

A use case diagram at its simplest is a representation of a user's interface with the system that shows the relationship between the user and different use cases in which the user is involved. A use case diagram can identify different types of users of a system and the different use cases and will often be accomplished by other types of diagram as well. It has been said that "use case diagrams the blue print for your system". The same is shown in the Fig 4.4.

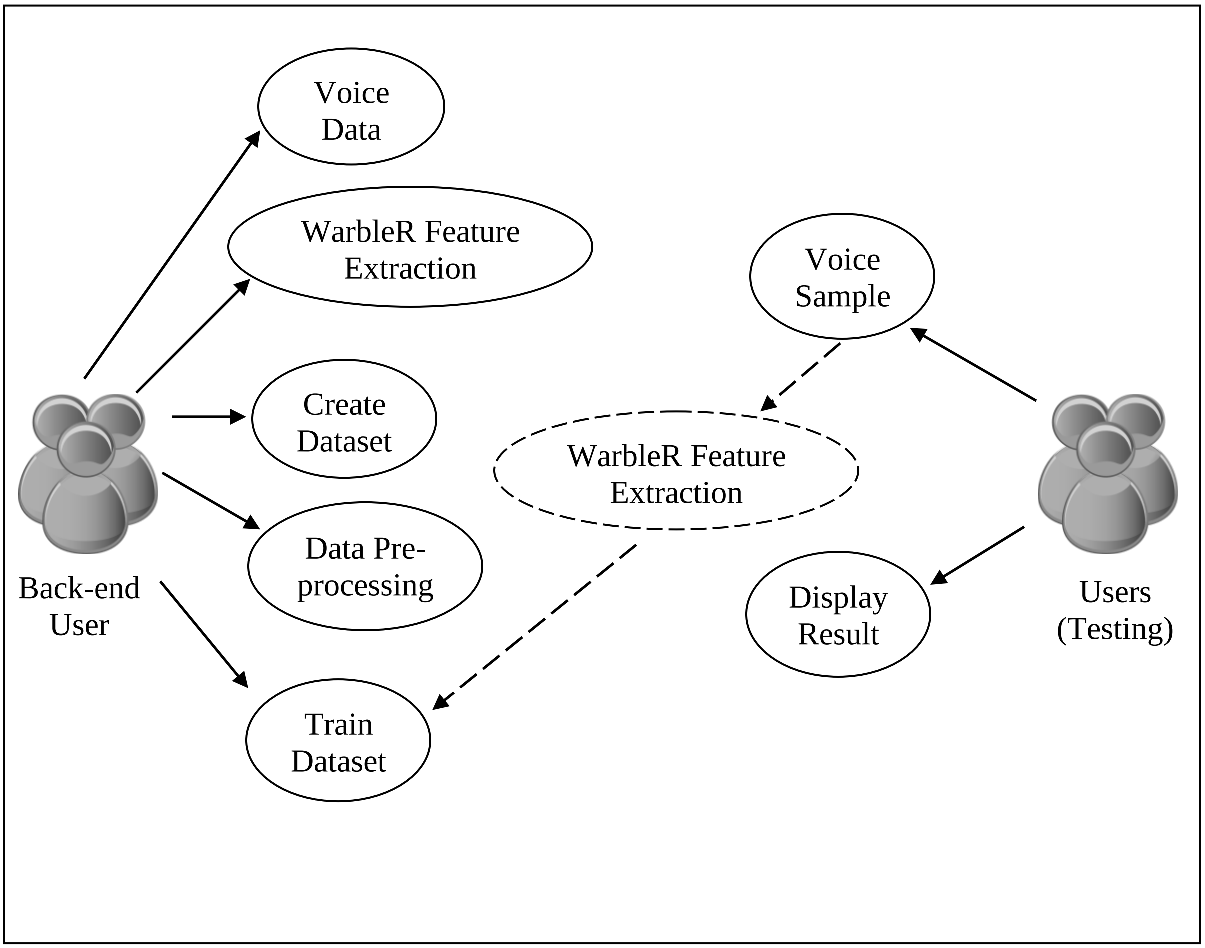


Fig 4.4 Use case diagram

4.5 Class Diagrams

Class diagram is a type of static structure diagram that describes the structure of a system and the relationship of the objects. The same is shown in the Fig 4.5

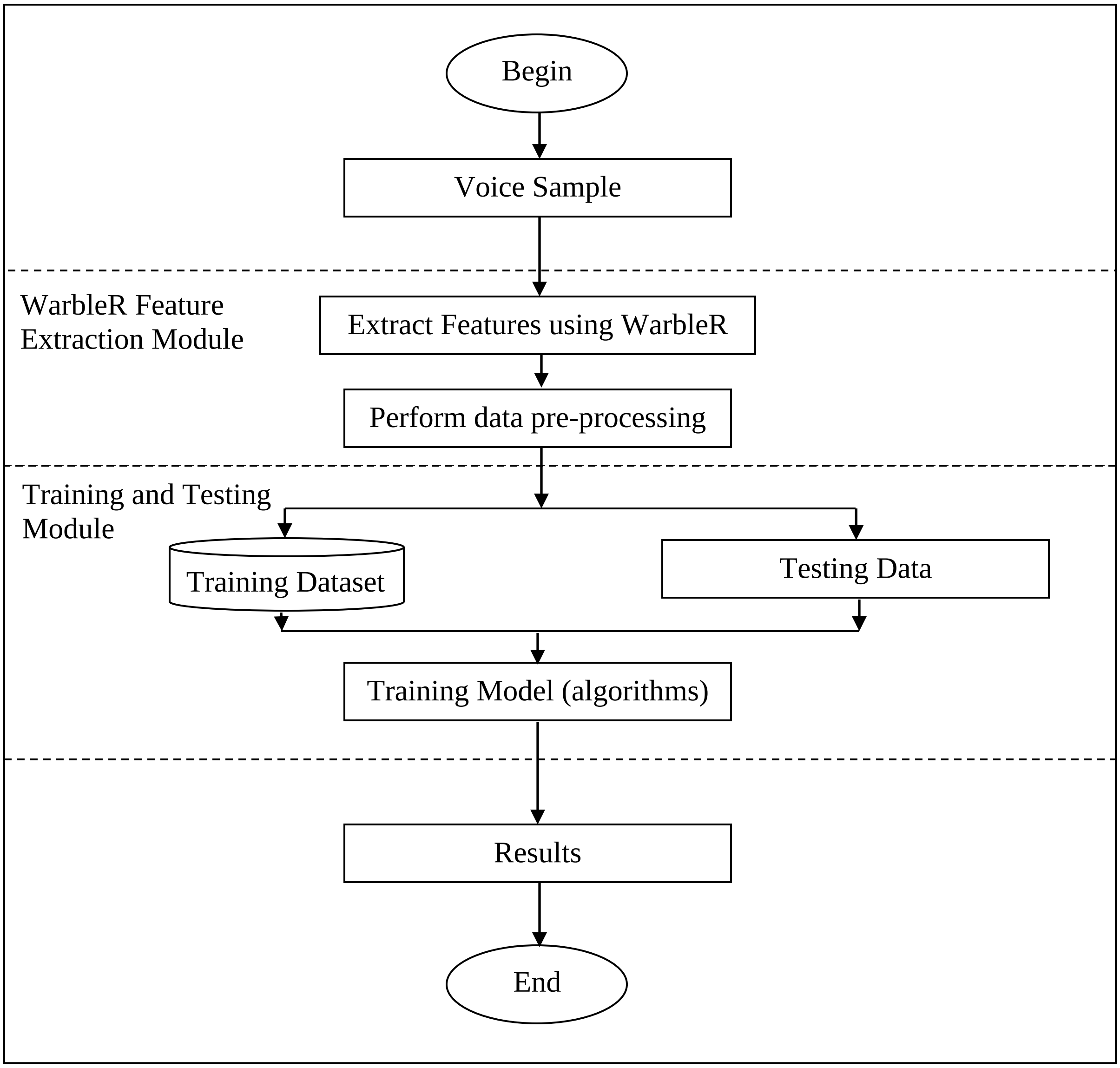


Fig 4.5 Class diagram

4.6 Sequence Diagrams

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of the message exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with the use case realization in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios. The same is shown in the Fig 4.6

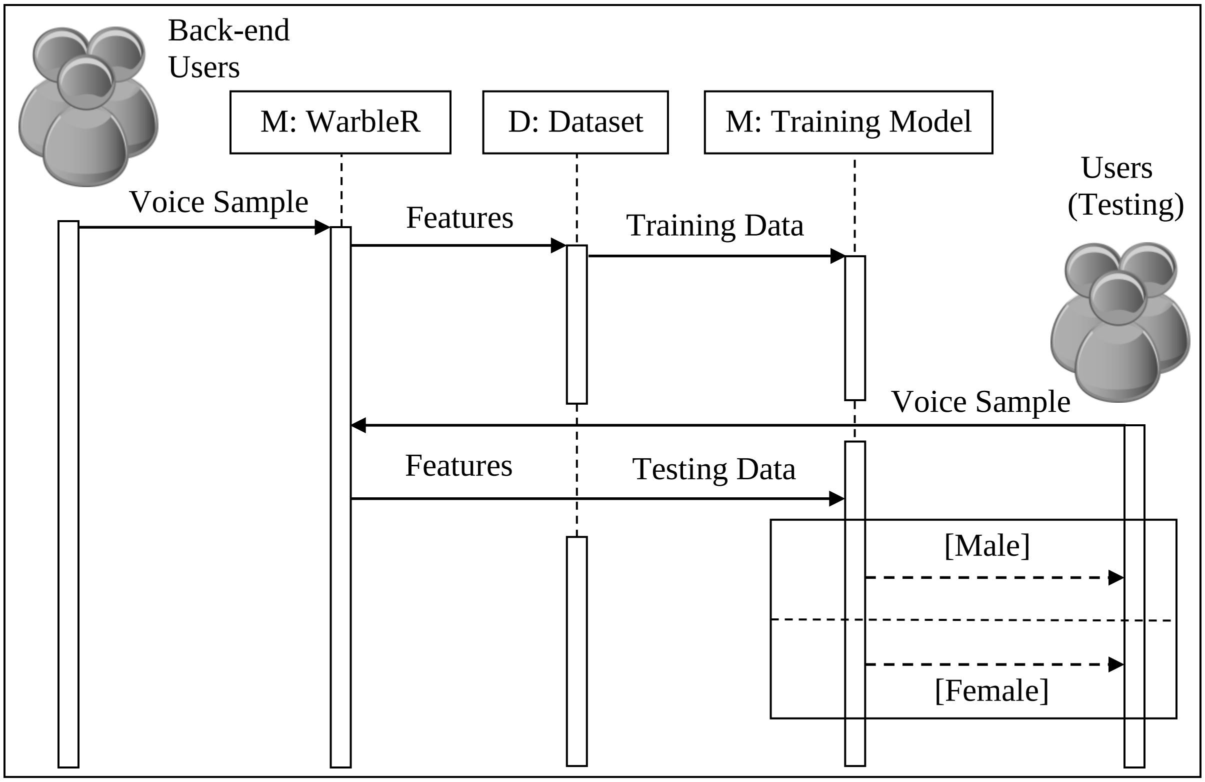


Fig 4.6 Sequence diagram

4.7 Activity Diagrams

Activity diagram is a representation of workflows of step wise activities and actions with support for choice, interaction and competency. Activity diagrams are intended to model both computational and organizational process (workflows), as well as the data flows intercepting with relating activities. The same is shown in the Fig 4.7

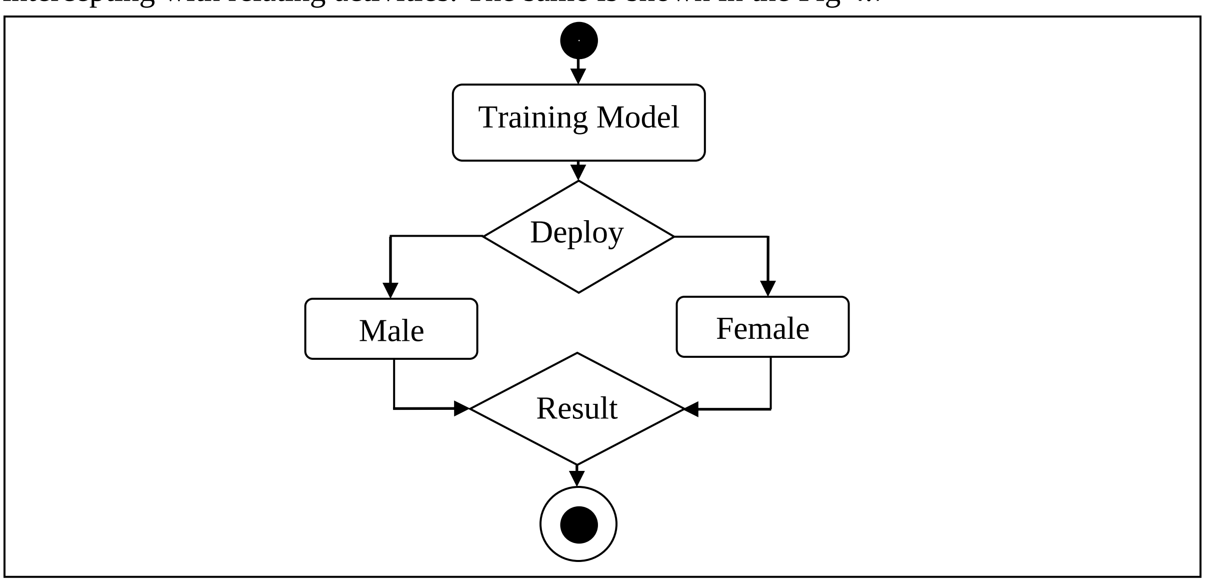


Fig 4.7 Activity diagram

4.8 Modules of the Project

4.8.1 Module 1

Module Name: Training Module

Functionality: The audio features extracted using the WarbleR program is feed to the training model in the form of CSV file. Pre-processing is done on the input dataset. Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. After this, the dataset is trained using Naïve Bayes classifier, Support Vector Machine, k-Nearest Neighbors classifier and Random Forest classifier.

Input: Training Voice Dataset (.csv).

Output: The system is trained.

Language used: Python version 3.8.2.

4.8.2 Module 2

Module Name: WarbleR Feature Extraction Module

Functionality: R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis. warbleR is a package designed to streamline analysis of acoustic signals in R. This package allows users to collect open-access avian vocalizations data or input their own data into a workflow that facilitates spectrographic visualization and measurement of acoustic parameters. warbleR makes fundamental sound analysis tools from the R package seewave, as well as new tools not yet offered in the R environment, readily available for batch process analysis.

Input: Audio File (.wav).

Output: Dataset in form of CSV.

Language used: R version 3.6.3.

4.8.3 Module 3

Module Name: Testing Module

Functionality: After the training is done, the system is ready for predictions. The user now records a new voice sample, this voice sample is uploaded through the web interface and feed to the WarbleR program to extract the audio features necessary. These features are now feed to Naïve Bayes classifier, Support Vector Machine, k-Nearest Neighbors classifier and Random Forest decision trees to make prediction or classify the given sample as either male or female. Now, for the final output the predictions of the most consistent algorithms are taken into consideration.

Input: Sample Audio File(.wav) or voice recorded via web interface.

Output: Prediction either male or female on the web interface.

Language used: Python version 3.8.2, R version 3.6.3, HTML5, CSS3, Flask version 1.1.2.

Chapter 5

IMPLEMENTATION

Implementation is the process of converting a new system design into an operational one. It is the key stage in achieving a successful new system. It must therefore be carefully planned and controlled. The implementation of a system is done after the development effort is completed.

5.1 Steps for Implementation

5.1.1 Installation of Software utilities:

1. Install Python version 3.8.2 from https://www.python.org/downloads/
2. Install R version 3.6.3 from https://cran.r-project.org/bin/windows/base/
3. After installing R, set environment variable for R in your windows machine.
4. Install Wamp 64-bit version from https://www.apachefriends.org/download.html
5. Install Brackets text editor for Web Interface development.
6. Install Anaconda Navigator for learning models development using python.

5.1.2 Sample Data Used:

The dataset designed include the voice samples downloaded from Vox Forge repository http://www.repository.voxforge1.org/downloads/SpeechCorpus/Trunk/Audio/. As stated in our problem statement, we included voice samples with various human emotional voices of either gender from https://voice123.com/, whose audio features being extracted in the Training Phase.

5.1.2 Debugging Phase:

Debugging refers to identifying, analyzing and removing errors. This process begins after the software fails to execute properly and concludes by solving the problem and successfully testing the software. we experienced errors while integrating the python and R scripts in the backend which we have overcome by incorporating flask framework. Flask is a micro web framework written in Python, it does not require specific tools or libraries. Flask was used to integrate the whole project, the web interface, learning model in python scripts and feature extraction model in R scripts.

5.1.4 Implementation steps:

1. Machine Learning models were coded in python programming language using Anaconda Navigator. The machine learning algorithms used for this project are Naïve Bayes classifier, k-Nearest Neighbor classifier, Random Forest Classifier and Support Vector Machines.

2. The Feature Extraction program was coded in R programming language. The different libraries used to compliment WarbleR for feature extraction are tuneR, seewave, caTools and xgboost. This R program returns the 20 audio features in a CSV file.

3. The 20 different audio features used for training the machine learning models are: mean frequency, standard deviation of frequency, median frequency, first quartile frequency, third quartile frequency, interquartile frequency range, skewness, kurtosis, spectral entropy, spectral flatness, mode frequency, frequency centroid, mean fundamental frequency, min. fundamental frequency, max. fundamental frequency, mean dominant frequency, min. dominant frequency, max. dominant frequency, range of dominant frequency and modulation index.

4. The web interface for the working of the system was developed in HTML5 for structuring the web pages and CSS3 for styling the web pages using Brackets text editor and implement through Wamp server.

5. The machine learning python scripts, feature extraction R program, the web interface developed to depict the working of the system all were integrated using Flask web framework. Given below are the steps followed for integrating the whole project using Flask web framework.
   - Download Python latest version
   - While installing Select Custom Installation and Choose ADD_PATH.

- Install above packages. Command to install packages: pip install (package_name)
- Ex: pip install pandas → If python is installed for ALL USERS (root directory)
  Else (AppData directory): python -m pip install pandas
- Run MySQL server or Just run WAMP server and No need create any database. It will create automatically.
- Extract and Paste above FlaskProject folder in ROOT or Desktop folder.
- Open cmd prompt and Navigate to FlaskProject folder and type: python app.py
- It will execute and it gives url: http://127.0.0.1:5000/ or http://localhost:5000/
- Access the Project Web Interface at http://localhost:5000/

6. The final working of the system can be accessed in the web interface. A user can either record his/her voice from the recorder in the web interface or upload a voice sample to the system with a button click. The test voice sample is processed in the back end. First, the audio file is fed as input to the R program and a CSV file is generated as the output with the audio features. And, next this CSV file is fed as input to the machine learning models in the python scripts, after processing the output is returned, which is to be displayed in the web interface. The output here is, the overall prediction, a table containing the prediction in different learning models and certain audio features.

5.2 Implementation Issues

The implementation phase of software development is concerned with translating design specifications into source code. The primary goal of implementation is to write source code and internal documentation so that conformance of the code to its specifications can be easily verified and so that debugging testing and modification are eased. This goal can be achieved by making the source code as clear and straightforward as possible. Simplicity clarity and elegance are the hallmarks of good programs and these characteristics have been implemented in each program module.

The goals of implementation are as follows.

- Minimize the memory required.
- Maximize output readability.
- Maximize source text readability.
- Minimize the number of source statements.
- Minimize development time.

5.3 Algorithms Used

5.3.1 Naïve Bayes Classifier

It is a classification technique based on Bayes' Theorem with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a feature in a class is unrelated to the presence of any other feature. Using Bayes theorem, P(A|B) = (P(B|A) x P(A)) / P(B), we can find the probability of A happening, given that B has occurred. Here, B is the evidence and A is the hypothesis. The assumption made here is that the predictors/features are independent. That is presence of one feature does not affect the other. Hence it is called naive.

Pseudocode:

Input: Training dataset **T**, **F** = (f_1, f_2, f_3, …, f_n) //value of the predictor variable in testing dataset.

Output: A class of testing dataset.

Steps:

1. Read the training dataset **T**;
2. Calculate the mean and standard deviation of the predictor variables in each class;
3. Repeat,

   Calculate the probability of f_i using the gauss density equation in each class;

   Until the probability of all predictor variables (f_1, f_2, f_3, …, f_n) has been calculated.
4. Calculate the likelihood for each class.
5. Get the greatest likelihood.

5.3.2 k-Nearest Neighbors Classifier

A k-nearest-neighbor classifier is a data classification algorithm that attempts to determine what group a data point is in by looking at the data points around it. An algorithm, looking at one point on a grid, trying to determine if a point is in group A or B, looks at the states of the points that are near it. The range is arbitrarily determined, but the point is to take a sample of the data. If most of the points are in group A, then it is likely that the data point in question will be A rather than B, and vice versa.

Pseudocode:

Classify (**X**, **Y**, x) // **X**: training data, **Y**: class labels of **X**, x: unknown sample

for i=1 **to** m **do**

Compute distance d (X_i, x)

end for

Compute set *I* containing indices for the k smallest distances d($\mathbf{X}_i$, x).

return majority label for {$\mathbf{Y}_i$ where i ∈ I}

5.3.3 Random Forest Classifier

Random Forest Classifier is an ensemble learning algorithm. The basic premise of the algorithm is that building a small decision-tree with few features is a computationally cheap process. If we can build many small, weak decision trees in parallel, we can then combine the trees to form a single, strong learner by averaging or taking the majority vote. In practice, random forests are often found to be the most accurate learning algorithms to date.

Pseudocode:

Precondition: A training set **S** := (x_1, y_1), ...,(x_n, y_n), features **F**, and number of trees in forest **B**.

1. **function** RandomForest(**S** , **F**)
2. $H \leftarrow \emptyset$
3. **for** i ∈ 1, ..., B **do**
4. $\mathbf{S}^{(i)} \leftarrow$ A bootstrap sample from **S**
5. $h_i \leftarrow$ RandomizedTreeLearn($S^{(i)}$, F)
6. $H \leftarrow H \cup \{ h_i \}$
7. **end for**

8. **return** *H*

9. **end function**

10. **function** RandomizedTreeLearn(**S**, **F**)

11. At each node:

12. f ← small subset of **F**

13. Split on best feature in f

14. **return** the learned tree

15. **end function**

5.3.4 Support Vector Machine

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.

Pseudocode:

Define number of features+1 as **F** and SVs+1 as **SV**

for each SV

    for each feature of the **SV**

        Read streamed data

        Convert it to float

        Store into *array_SVs [SV][F]*

    end for

end for

Read streamed data

Convert it to float

Store into *array_ay [0]*(b value)

for each SV

    Read streamed data

```
        Convert it to float
        Store into array_ay [SV]
end for
for each feature
        Read streamed data
        Convert it to float
        Store into array_test [F]
end for
for each feature
        Clear array_AC [F]
end for
for each SV
        for each feature of the SV
                array_AC [F] += array_AC [F] * array_test [F]
        end for
end for
for each feature
        Distance_value += array_AC [F] * array_test [F]
end for
Distance_value -= b
if (Distance_value >= th) then
        return 1
else
        return -1
end if
```

Chapter 6

TESTING

This chapter gives the outline of all testing methods that are carried out to get a bug free system. Quality can be achieved by testing the product using different techniques at different phases of the project development. The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components sub-assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 Test Environment

Testing is an integral part of software development. Testing process certifies whether the product that is developed compiles with the standards that it was designed to. Testing process involves building of test cases against which the product has to be tested. The testing environment requires a localhost server. The localhost server software used for this project is Wamp server, which includes the Apache Tomcat server and MySQL server 8.0. The project is programmed in python and R programming language; hence the environment should include python 3.8 and R 3.6 program execution environments.

6.2 Unit Testing of Modules

6.2.1 Training Module

Table 6.1 Test Cases for Training Module

Cases	Test Data	Expected Results	Observed Results	Remarks
Case 1	Upload Dataset (.csv)	Uploaded Successfully	Uploaded Successfully	Pass
Case 2	Train Dataset (.csv)	Learning Models Accuracy Graph	Learning Models Accuracy Graph	Pass

6.2.2 WarbleR Feature Extraction Module

Table 6.2 Test Cases for Feature Extraction Module

Cases	Test Data	Expected Results	Observed Results	Remarks
Case 1	Input an audio file (.wav)	20 features extracted in a CSV file (testData.csv)	20 features extracted in a CSV file (testData.csv)	Pass

6.2.3 Testing Module

Table 6.3 Test Cases for Testing Module

Cases	Test Data	Expected Results	Observed Results	Remarks
Case 1	Upload Audio File (.wav/.mp3)	Uploaded Successfully	Uploaded Successfully	Pass
Case 2	Record User Audio using web Recorder	Uploaded Successfully	Uploaded Successfully	Pass
Case 3	Play Uploaded Audio	Audio played along with voice frequency, waveform, and spectrum graph	Audio played along with voice sampling frequency, waveform, and spectrum graph	Pass
Case 4	Test an uploaded Audio File	Final prediction along with predictions of individual models and audio waveform	Final prediction along with predictions of individual models and audio waveform	Pass
Case 5	Testing of a male voice	Final prediction: Male	Final prediction: Male	Pass
Case 6	Testing of a female voice	Final prediction: Female	Final prediction: Female	Pass
Case 7	Delete uploaded Audio File	Audio file Deleted	Audio file Deleted	Pass

6.3 Integration Testing of Modules

6.3.1 Module 1: Training Module

A dataset in form of a CSV file is uploaded from the “Train” page in the web interface. As soon as the user uploads the dataset, in the background the dataset is passed as an input to a python method “calculate” in “traintest.py”. Then, the method returns the training and testing accuracies of each learning model used, and this result is depicted in from a graph on the “Train” page.

6.3.2 Module 2: WarbleR Feature Extraction Module

When a user clicks on “Test” button on the “Test & Results” page in the web interface, in the background the audio file is passed as a parameter to “processFolder” function of “audio.R” file, which in-turn calls the “specan3” function which creates a CSV file “testData.csv” and writes the audio features into the CSV files.

6.3.3 Module 3: Testing Module

A user can upload an audio file in wav/mp3 format directly or can record the voice using the recorder in the “Upload” page. This audio file can now be tested in the “Test & Result” page. When a user clicks on “Test” button, in the background the audio file is processed in “audio.R” and the resulting CSV file is fed to “calculate_voice” method in “test_seperate.py”. This method processes the features of the audio file in all of the learning models, and the resulting predictions is displayed on the “Test & Result” page along with the audio waveform.

6.4 System Testing

Ultimately, software is included with other system components and the set of system validation and integration tests are performed. System testing is a series of different tests whose main aim is to fully exercise the computer-based system. Although each test has a different role all work should verify that all system elements are properly integrated and formed allocated functions.

Table 6.4: Test Cases for Input-Output

Name of the Test	**System Testing**
Item being tested	Overall functioning of GUI with all functions properly linked.
Sample Input	Recorded and Uploaded Audio file.
Expected Output	All the modules working as expected.
Actual Output	Application reacts to user inputs in expected manner.
Remarks	Successful

6.5 Functional Testing

Functional testing is used to ensure the visual clarity of the system, flexibility of the system, user friendliness of the system. The various components which are to be tested are:

- Relative layout
- Various Links and Buttons

Chapter 7

RESULTS

This section describes the screens of the "**Gender Identification by Voice**". The snapshots are shown below for each module.

Snapshot 1: Home Page of the web interface

This snapshot as in Fig 7.1, shows the "Home" page of the web interface developed for the project. From this page, a user can navigate to "Upload", "Train" or "Test & Train" page.

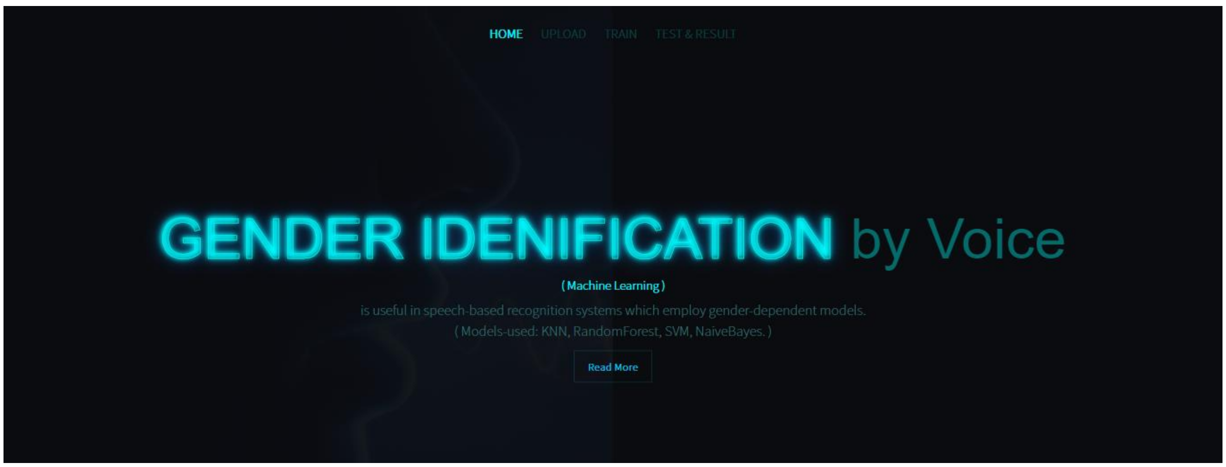


Fig 7.1 Snapshot of the Home Page

Snapshot 2: System Details

This snapshot as in Fig 7.2, shows the details and working of the system along with execution steps.

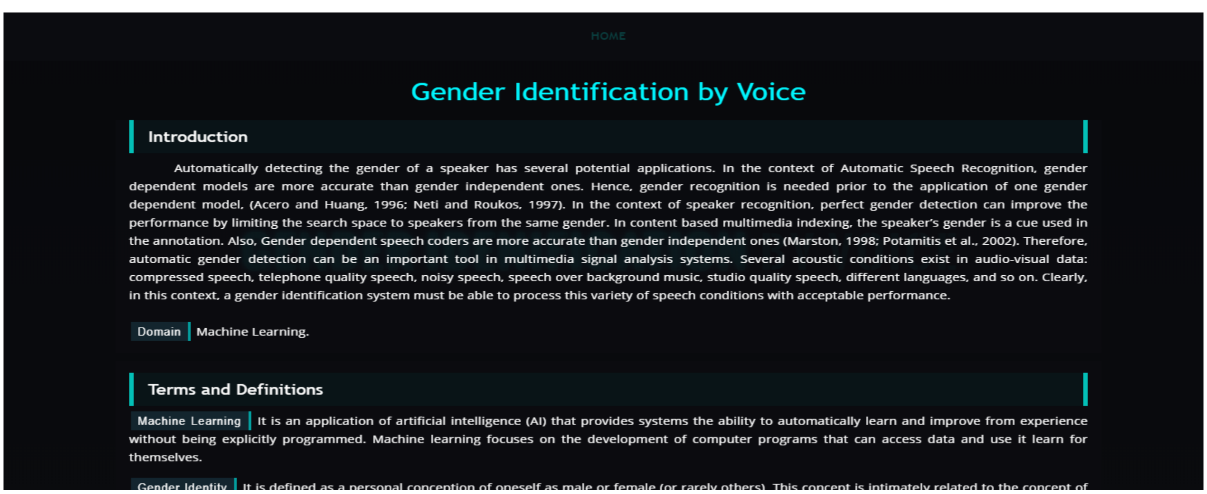


Fig 7.2 Snapshot of the System Details

Snapshot 3: Audio Upload Page, View-1

This snapshot as in Fig 7.3, shows the "Upload" page, where a user can upload a wav/mp3 audio file directly from his/her system.

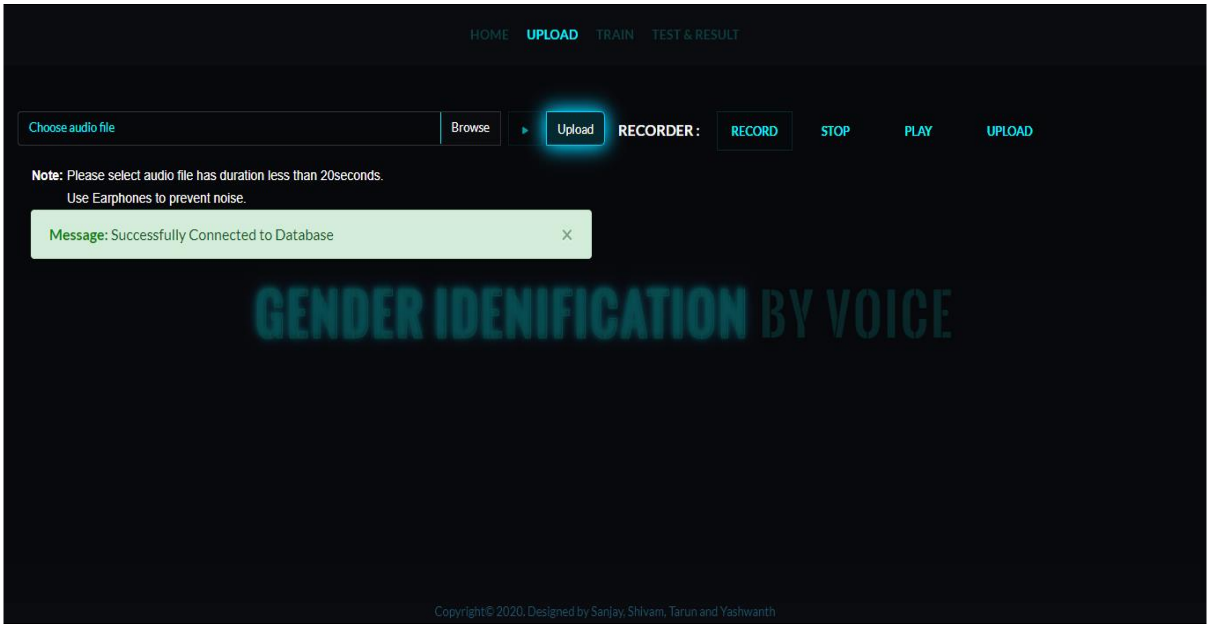


Fig 7.3 Snapshot of the Upload Page View-1

Snapshot 4: Audio Upload Page, View-2

This snapshot as in Fig 7.4, shows the "Upload" page, where a user can record their voice using the recorder provided on the page and can play or upload the file.

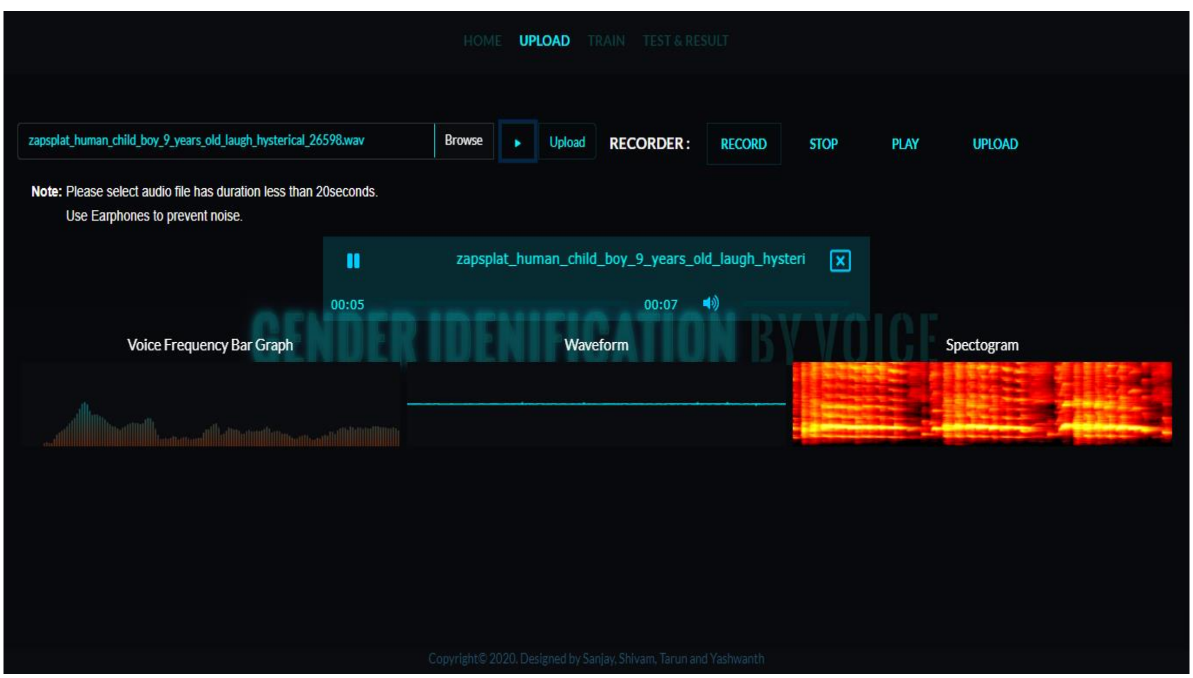


Fig 7.4 Snapshot of the Upload Page View-2

Snapshot 5: System Training Page

This snapshot as in Fig 7.5, shows the "Train" page, where a user can upload the voice training dataset in form of a CSV file and as a result the accuracies in each learning model is depicted in a graph.

Fig 7.5 Snapshot of the Train Page

Snapshot 6: Testing and Result Page, View-1

This snapshot as in Fig 7.6, shows the "Test & Result" page, where a user can test the already uploaded audio file gender prediction. As a result, the final prediction along with predictions of each learning model is displayed. The audio upload in this case is a Male's voice and hence the system predicts it as "Male" accurately.

Fig 7.6 Snapshot of the Test & Result Page View-1

Snapshot 7: Testing and Result Page, View-2

This snapshot as in Fig 7.7, shows the “Test & Result” page, where a user can test the already uploaded audio file gender prediction. As a result, the final prediction along with predictions of each learning model is displayed. The audio upload in this case is a Female’s voice and hence the system predicts it as “Female” accurately.

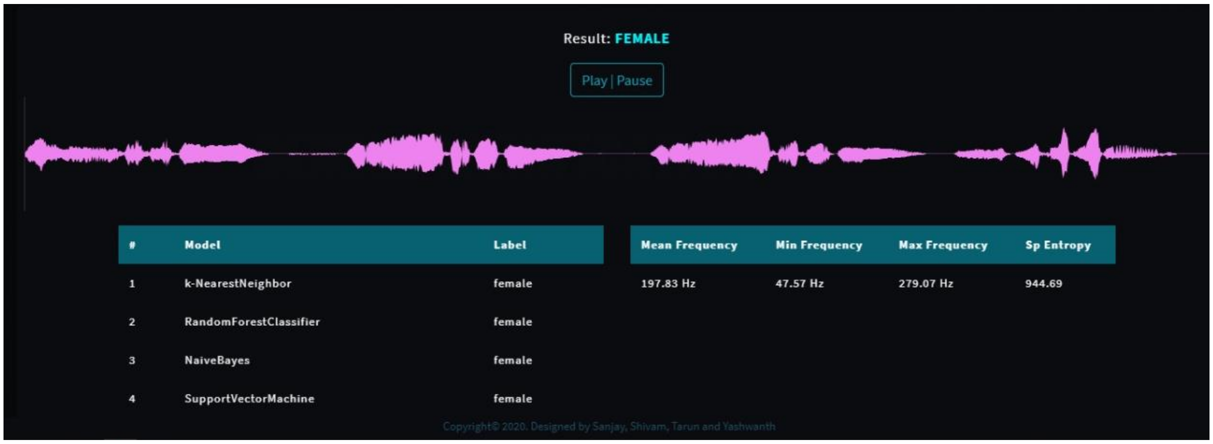


Fig 7.7 Snapshot of the Test & Result Page View-2

Snapshot 8: Background Processing

This snapshot as in Fig 7.8, shows the background processing in windows command prompt as the system is being run on the web interface. Initially the system is started by executing “app.py” file in this prompt.

Fig 7.8 Snapshot of the Background Processing

Chapter 8

CONCLUSION

Recognizing the gender of human voice has been considered one of the challenging tasks because of its importance in various applications. The contributions are threefold including studying the extracted features by examining the correlation between each other, building classification models using different ML techniques from distinct families, and evaluating the natural feature selection techniques in finding the optimal subset of relevant features on classification performance.

Our experiments involve applying standard machine learning techniques such as Naïve Bayes, k-NN, Random Forest and Support Vector Machine to the voice-based gender identification problem. We can observe that Random Forest Decision Trees provides best accuracy which is up to 99% as the testing accuracy. In addition, we also conclude that general-purpose audio features may not be able to capture enough gender-specific characteristics of voice.

8.1 Major contributions

Automatically detecting the gender of a speaker has several potential applications. The applications of gender detection system have increased significantly due to the recent developments in speech/speaker recognition, human-computer interaction, and biometric security systems including authentication to access data, surveillance, and security.

Moreover, a gender detection system can be used for automatic transfer of a phone call of a male/female to the relevant person or department. In a mobile healthcare system, gender detection can play a significant role. There are some vocal folds pathologies, which are biased to a gender; for example, vocal folds cyst can be seen particularly in female patients. If there is a mechanism to automatically detect the gender of the patient, it is easier for a healthcare professional to prescribe the appropriate treatment.

8.2 Future Enhancements

It would be interesting to introduce high order audio features to our models. In the future, more experiments can be conducted to use various feature categories, ML techniques, and other natural feature selection techniques. Furthermore, the proposed techniques can be examined on different datasets, since here only standard voice samples were used.

BIBLIOGRAPHY

[1] Tom M. Mitchell, Machine Learning, India Edition 2013, McGraw Hill Education.

[2] Randy Connolly, Ricardo Hoar, "Fundamentals of Web Development", 1st Edition, Pearson Education India. (ISBN:978-9332575271)

[3] Charles R. Severance, "Python for Everybody: Exploring Data Using Python 3", 1st Edition, CreateSpace Independent Publishing Platform, 2016.

[4] https://www.rdocumentation.org/packages/warbleR/

[5] https://palletsprojects.com/p/flask/

[6] https://www.python.org/

[7] https://www.apachefriends.org/index.html

[8] Hadi Harb and Liming Chen, "Voice-Based Gender Identification in Multimedia Applications" J. Intell. Inf. Syst. 24. 179-198. 10.1007/s10844-005-0322-8, 2005.

[9] Kunyu Chen, "Gender Identification by Voice", Stanford University, 2014.

[10] Sarah Ita Levitan, Taniya Mishra & Srinivas Bangalore, "Automatic identification of gender from speech", 84-88. 10.21437/SpeechProsody, 2016.

[11] T. Jayasankar, K. Vinothkumar and Arputha Vijayaselvi, "Automatic Gender Identification in Speech Recognition by Genetic Algorithm", Applied Mathematics & Information Sciences 11, No. 3, 907-913, 2017.

[12] Remna R. Nair, Bhagya Vijayan, "Voice based Gender Recognition", International Research Journal of Engineering and Technology (IRJET), Vol. 6, No. 5, May, 2019.

[13] A Raahul, R Sapthagiri, K Pankaj and V Vijayarajan, "Voice based gender classification using machine learning", IOP Conf. Series: Materials Science and Engineering 263, 2017.

JOURNAL PUBLICATION

Our project was published in **MAT Journals** as “Voice Grounded Gender Identification”, Journal of Web Development and Web Designing, Vol 5, Issue 2, July 2020. And the following are the certificates that were issued to our group.

MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled Voice Grounded Gender Identification Submitted by Sanjay T. J has been published in Journal of Web Development and Web Designing Volume 5 Issue 2 Year 2020

Arvind Sharma
Managing Director, MAT Journals
(A Unit of ARV InfomediaPvt. Ltd.)

INDEXING:DOI||Cosmos Impact Factor|| J-Gate||Figshare||Google Scholar||Cite Factor||Zenodo||Scientific Indexing Services||Root Indexing

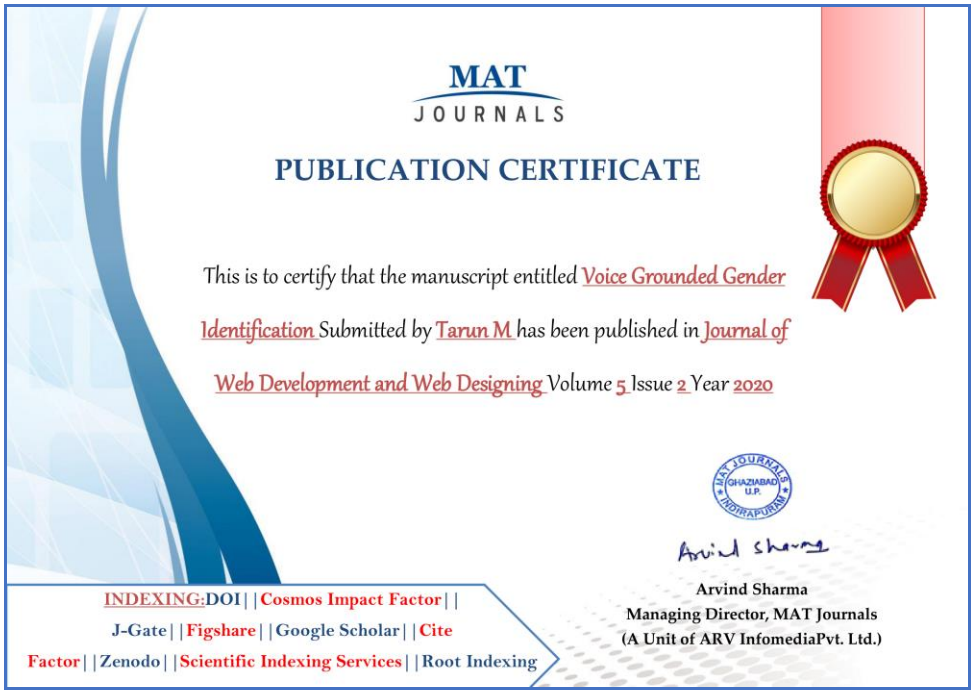
MAT JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled Voice Grounded Gender Identification Submitted by Shivam Upadhyay has been published in Journal of Web Development and Web Designing Volume 5 Issue 2 Year 2020

Arvind Sharma
Managing Director, MAT Journals
(A Unit of ARV InfomediaPvt. Ltd.)

INDEXING:DOI||Cosmos Impact Factor|| J-Gate||Figshare||Google Scholar||Cite Factor||Zenodo||Scientific Indexing Services||Root Indexing



MAT
JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled Voice Grounded Gender Identification Submitted by Tarun M has been published in Journal of Web Development and Web Designing Volume 5 Issue 2 Year 2020

Arvind Sharma
Managing Director, MAT Journals
(A Unit of ARV InfomediaPvt. Ltd.)

INDEXING:DOI||Cosmos Impact Factor|| J-Gate||Figshare||Google Scholar||Cite Factor||Zenodo||Scientific Indexing Services||Root Indexing

MAT
JOURNALS

PUBLICATION CERTIFICATE

This is to certify that the manuscript entitled Voice Grounded Gender Identification Submitted by Yashwanthgowda H. R has been published in Journal of Web Development and Web Designing Volume 5 Issue 2 Year 2020

Arvind Sharma
Managing Director, MAT Journals
(A Unit of ARV InfomediaPvt. Ltd.)

INDEXING:DOI||Cosmos Impact Factor|| J-Gate||Figshare||Google Scholar||Cite Factor||Zenodo||Scientific Indexing Services||Root Indexing