**CHAPTER - 1**

**INTRODUCTION**

1.1 DOMAIN DESCRIPTION:

1.1.1 MACHINE LEARNING:

Machine Learning is an application of artificial intelligence (AI) that provides system 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 to learn for themselves.

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, impr­oves with experience E." - Tom M. Mitchell, Carnegie Mellon University.

**1.1.2 APPROACHES:**

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

**SUPERVISED MACHINE LEARNING:**

Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal.

**UN-SUPERVISED LEARNING:**

Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labelled, classified or categorized.

**1.1.3 REAL LIFE EXAMPLE FOR MACHINE LEARNING:**

## 1. SPEECH RECOGNITION:

Machine learning can translate speech into text. Certain software applications can convert live voice and recorded speech into a text file. The speech can be segmented by intensities on time-frequency bands as well.  
Real-world examples of speech recognition:

1. Voice search
2. Voice dialing
3. Appliance control

**2. MEDICAL DIAGNOSIS:**

Machine learning can help with the diagnosis of diseases. Many physicians use chatbots with speech recognition capabilities to discern patterns in symptoms.  
Real-world examples for medical diagnosis:

1. Assisting in formulating a diagnosis or recommends a treatment option.
2. Oncology and pathology use machine learning to recognize cancerous tissue.
3. Analyze bodily fluids.

**3. STATISTICAL ARBITRAGE:**

Arbitrage is an automated trading strategy that’s used in finance to manage a large volume of securities. The strategy uses a trading algorithm to analyse a set of securities using economic variables and correlations.  
Real-world examples of statistical arbitrage:

1. Algorithmic trading which analyses a market microstructure.
2. Analyze large data sets.
3. Identify real-time arbitrage opportunities.

1.2 ABOUT PROJECT:

Biometric authentication is simply the process of verifying a person’s identity using their unique characteristics and then allowing them to log into a service, application, or device, etc.

Biometric [authentication](https://www.aware.com/authentication-verification/), unlike passwords or token-based authentication, uses unique biological characteristics to verify an individual’s identity. It’s harder to spoof and generally more convenient for users since they don’t have to remember passwords or carry a physical token that can easily be lost or stolen. The [authenticator](https://www.aware.com/mobile-biometric-authentication/) is part of the individual.

To understand it better, biometrics is the general name for any type of body characteristic or identifier like a facial pattern, fingerprint, voice, iris pattern. Biometric identification verifies that "you are you" based on a chosen characteristic or a body part. Biometric authentication goes one step further and uses that biometric identifier to compare whether it matches the person's biometric information already stored in a database, allowing them to access a given environment or perform certain actions.

It’s almost normal now to secure your phone with a fingerprint. But how comfortable are you offering a passcode using your voice?

### 1.2.1 VOICE AUTHENTICATION:

Voice authentication is a type of security authentication that relies on a person's unique voice patterns for identification in order to gain access. This type of authentication requires a device that can capture a person's voice very accurately and software that can recognize the patterns of the voice and compare it to already-recognized patterns.

Voice authentication is also known as voice biometrics, voice ID or speaker recognition.

To ensure correctness of the biometric authentication system, a voice enrolment is necessary. In this phase, a person's unique biometric voice patterns are processed, stored as a voiceprint, and secured in a database. This information will later allow validation and identification of an individual by the system.

## 1.2.2 TYPES OF VOICE AUTHENTICATION:

There are two common forms of voice authentication engines, namely: text-independent and text-dependent.

**TEXT INDEPENDENT:**

Here, the system does not store any pre-recorded audio to compare it with the input. It is a method of voice authentication without any prior speech content inputted into the biometric system. It is much more convenient since it offers free speech.

**TEXT-DEPENDENT:**

For text-dependent recognition, a previously given paraphrase that is stored in the system has to be repeated, there is a constraint in speech content here.

### 1.2.3 ENROLLMENT AND VERIFICATION:

A voice biometric sample must be captured and enrolled using a microphone to create a reference template to compare against samples for future authentication attempts. Unique vocal qualities are then analysed; such as:

1. Duration
2. Intensity
3. Dynamics
4. Pitch
5. Amplitude

### 1.2.4 HOW DOES VOICE AUTHENTICATION WORK?

Voice biometric recognition starts with a voice input from the person whose voice identity needs to be stored in the system. This print is then kept for authentication purposes.

The voiceprint is made with the use of software that splits the statement into various frequencies. More behavioural attributes are identified at this stage that works as a team to mutually make-up the voiceprint.

Since voiceprints behave just like fingerprints, each one of them is peculiar and connected to a sole person. The prints are stored in databases to identify later and acknowledge users.

For example, during the procedure, a text-independent system will focus on matching unfamiliar speeches with the previously stored voiceprint of a person.

There are over 70 body parts that contribute to how a person speaks. Voice recognition technology works by digitizing a person's speech to create a template. This is also known as a stored model voice print. The technology reduces each spoken word to segments composed of dominant frequencies called formants. Each formant has several tones that collectively identify a speaker's unique voice print.

**1.3 OBJECTIVE:**

Voice authentication makes logins as easy as speaking. This may sound futuristic, but it’s the security that’s saving businesses from breaches every day. Voice authentication is a technology that uses a person’s voice to identify and validate their identity.

This type of biometric solution is very effective and fast at processing voice patterns. Implementation of such systems is also straightforward.

The training and testing is done by using the Gaussian mixture model of machine learning algorithm.

**CHAPTER - 2**

**LITERATURE SURVEY**

**2.1 THEORETICAL BACKGROUND:**

This is project based on voice authentication. By using voice the user will login. By using Voice biometrics is the primary interest within this paper. Voice biometric is a form of inherence-based authentication factor in that it, like fingerprint, iris etc., is supposed to be unique to the user. Similar to other inherence based authentication factors, the user uses their own voice to authenticate themselves. To do this, users will often use some form of microphone to record their voice, which is often verified in real time against their voice print that is on file.

While the technology for voice biometric authentication has been around for years now, only in recent years has it seen huge developments, primarily by companies such as Nuance, regarding it being applied commercially and being considered a secure way for users to authenticate themselves.

With the growth of smart technologies in many different sectors such as hospitals, financial sectors, the military, aviation there is an even greater need to determine the authenticity of a genuine user. Authentication can be defined as the process of verifying an identity claim using the users’ knowledge (e.g., secret questions, passwords, PINs), their possessions (e.g., ID cards, mobile phones, tokens), their location, or their biometrics (e.g., biometrics, fingerprints signatures) all of which can be referred to as different authentication factors (Fu 2015). The classification of user authentication factors can be seen in, the taxonomy we created, which classifies authentication factors into four main categories, Knowledge-based, Biometric (or inheritance)-based, Ownership-based, and Location-based authentication factors.

##### EXISTING SYSTEM DRAWBACKS:

Existing methods for authentication. Authentication can be defined as the process of identification and verification claim using the users’ knowledge (e.g., secret questions, passwords, PINs), their possessions (e.g., ID cards, mobile phones, tokens), their location, or their biometrics (e.g., biometrics, fingerprints signatures) all of which can be referred to as different authentication factors.

**FINGER PRINT BIOMETRIC:**

The fingerprint-based biometric is the oldest and most conventional method due to the varied fingers print of human beings. All the biometric systems authenticate and validate an individual’s identity based on stored data. Fingerprint recognition is a type of physical biometrics. For this authentication method, a fingerprint scanner is used to authenticate data.

**FACIAL BIOMETRIC SYSTEM:**

The facial structure most common technique in day-to-day life to remember any other person. In this present generation, there are excellent quality cameras that can capture facial expressions from very distance also and can also be used for safety and security purposes. But this biometric system is also evident for many frauds, scam attack and false recognition due to aging, injury and identical twins. Therefore, this technique is having very limited applications.

**IRIS BIOMETRIC SYSTEM:**

The iris is the colored circular segment at the front of the eye that contains the pupil at its center. The iris controls the size of the pupil to adjust the amount of light entering the eye. Iris recognition technology uses the unique patterns of the colored tissue that form the iris. These patterns are captured by a camera operating in near infra-red wavelengths.

##### 2.3 PROPOSED SYSTEM FEATURES:

The GMM approach gives better result if the speech signal is clean, and the performance is highly degraded for a noisy speech data showed that the usage of a universal background model in combination with missing data recognition yields substantial improvements in recognition performance, especially in the presence of highly non-stationary background noise at low signal-to-noise ratio. Suggested a robust feature extraction technique based on filter bank. The future work is to find a robust feature extraction technique using MCFF transform for a noisy data so that the performance of an automatic speech recognition would be improved.

**2.4 ADVANTAGES OF PROPOSED SYSTEM:**

1. GMM is introduced to train the voiceprint feature model of voice segment data effectively. When number of training samples is 5, the success rate of system authentication is between 89% and 96% and time required to authenticate a voice sample is between 210ms~320ms, which demonstrate the high accuracy and good real-time.
2. The advantage of Mixture models is that they do not require which subpopulation a data point belongs to. It allows the model to learn the subpopulations automatically.
3. GMM is a lot more flexible in terms of cluster covariance.

**2.5 FEASIBILITY STUDY:**

As the name implies, a feasibility analysis is used to determine the viability of an idea, such as ensuring a project is legally and technically feasible as well as economically justifiable. It tells us whether a project is worth the investment—in some cases, a project may not be doable. There can be many reasons for this, including requiring too many resources, which not only prevents those resources from performing other tasks but also may cost more than an organization would earn back by taking on a project that isn’t profitable.

### 2.5.1 Operational Feasibility:

The number of people working on this project are 3 to 4. These persons should have knowledge on the technologies from the domain of Artificial Intelligence (A.I.), those are understanding of Machine Learning (M.L.) and its types. Working of Natural Language Processing (N.L.P.).

### 2.5.2 Technical Feasibility:

Technical feasibility is the study which assesses the details of how you intend to deliver a product or service to customers. Think materials, labor, transportation, where your business will be located, and the technology that will be necessary to bring all this together. It’s the logistical or tactical plan of how your business will produce, store, deliver and track its products or services.

#### 2.5.2.1 Survey of Technology:

For our project we have chosen the Artificial Intelligence aka A.I. Technology as we found that by using this technology, we can complete our project and get out desired output for the users.

#### 2.5.2.2 Feasibility of Technology:

For our project from Machine learning aka M.L., we have chosen Unsupervised Machine Learning task to train our data on the GloVe, i.e., Global Vectors for Word.

Representation Dataset. Later, training on this dataset, we’ll then, give our inputs to the model and it’ll display the top N Sentences.

### 2.5.3 Economic Feasibility:

Our project is economically supportive as its required small or medium amount of resources which will cost up-to medium amount for those resources.

**CHAPTER -3**

**SYSTEM ANALYSIS**

It is a process of collecting and interpreting facts, identifying the problems, and decomposition of a system into its components.

System analysis is conducted for the purpose of studying a system or its parts in order to identify its objectives. It is a problem-solving technique that improves the system and ensures that all the components of the system work efficiently to accomplish their purpose

**3.1 SPECIFICATION:**

### 3.1.1 Functional Specification:

1. The following are the functional requirements of our project.
2. A training dataset has to be created on which training is performed.
3. A testing dataset has to be created on which testing is performed

**3.1.2 Non-Functional Requirements:**

1. **Maintainability**: Maintainability is used to make future maintenance easier, meet new requirements.
2. **Robustness**: Robustness is the quality of being able to withstand stress, pressures or changes in procedure or circumstances.
3. **Reliability:** Reliability is an ability of a person or system to perform and maintain its functions in circumstances.
4. **Size:** The size of a particular application play a major role, if the size is less then efficiency will be high.
5. **Speed:** If the speed is high then it is good. Since the number of lines in our code is less, hence the speed is high.

**3.2 HARDWARE REQUIREMENTS:**

The selection of hardware is very important in the existence and proper working of any software. In the selection of hardware, the size and capacity requirements are also important.

|  |  |
| --- | --- |
| **PROCESSOR** | **INTEL CORE (I3/I5/I7)** |
| **RAM** | **4.00GB** |
| **HARD DISK** | **500GB** |

**3.3 SOFTWARE REQUIREMENTS:**

One of the most difficult tasks is that, the selection of the software, once system requirement is known that is determining whether a particular software package fits the requirements.

|  |  |
| --- | --- |
| **TECHNOLOGY** | **PYTHON 3.8.32** |
| **IDE** | **JUPYTER NOTEBOOK (ANACONDA)** |
| **OPERATING SYSTEM** | **WINDOWS 7/8/10** |
| **BROWSER** | **GOOGLE CHROME (51.2 AND ABOVE)** |

##### 3.4 MODULE DESCRIPTION:

##### In our project we have used the different modules for identifying and verifying we are using the user login and for data analysis, pre-processing, visualization, etc.

##### Data analysis and Pre-processing

##### Model Training and Testing

##### Feature Extraction

##### Accuracy measures

##### Visualization

1. **DATA ANALYSIS AND PRE-PROCESSING:**
2. Data Analysis and data preprocessing is a process of preparing raw data and making it suitable for a machine learning model.
3. It is a first and crucial step while creating a machine learning model.
4. When creating a machine learning project, it is not always the case that we come across clean and formatted data.

For data analysis we are using the data base to store the data. For that sqlite3 module is used.

**SQLite:**

SQLite is a C library that provides a lightweight disk-based database that doesn’t require a separate server process and allows accessing the database using a nonstandard variant of the SQL query language. Some applications can use SQLite for internal data storage. It’s also possible to prototype an application using SQLite and then port the code to a larger database such as PostgreSQL or Oracle.

Firstly, we will define our libraries like numPy, tkinter, sklearn, in the code. NumPy is used for matrix multiplications, Pandas is used for data manipulation, tkinter for visualization of data and sklearn is for Machine learning Algorithms.

For pre-processing sklearn library is used. In the sklearn library there was a module called pre-processing. That module is used for data cleaning.

**SKLEARN:**

Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction via a consistence interface in Python

For pre-processing the sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers, transformers, and normalizers on a dataset containing marginal outliers is highlighted in [Compare the effect of different scalers on data with outliers](https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py).

1. **MODEL TRAINING AND TESTING:**

For model training and testing there is a module called Gaussian mixture model. From

Sklearn library we can import the Gaussian mixture model.

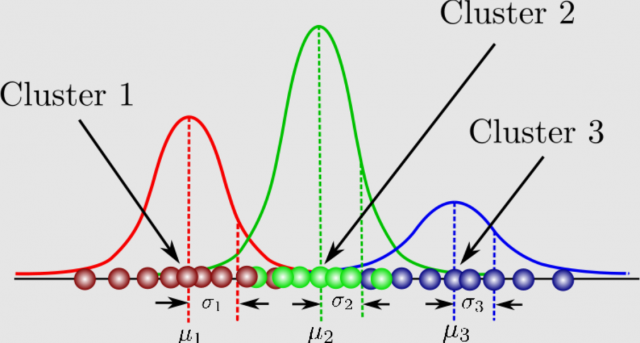
To model a system which has two distinct states (like speech and noise) and train a GMM with mixture components matching those states. When receiving a microphone signal, we can then determine the likelihood of each mixture component and thus obtain the likelihood that the signal is speech or noise.

The algorithm we have used in this project Gaussian mixture model

**GAUSSIAN MIXTURE MODEL**

Mixture of Gaussian as it is sometimes called, is not so much a model as it is a [probability distribution](https://www.mygreatlearning.com/blog/understanding-probability-distribution/). It is a universally used model for generative unsupervised learning or clustering. It is also called Expectation-Maximization Clustering or EM Clustering and is based on the optimization strategy. Gaussian Mixture models are used for representing Normally Distributed subpopulations within an overall population. The advantage of Mixture models is that they do not require which subpopulation a data point belongs to. It allows the model to learn the subpopulations automatically. This constitutes a form of unsupervised learning.

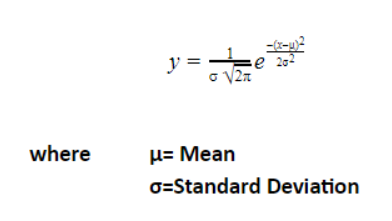
A Gaussian is a type of distribution, and it is a popular and mathematically convenient type of distribution. A distribution is a listing of outcomes of an experiment and the probability associated with each outcome.



**Figure 3.4.1 Gaussian mixture model diagram**

**NORMAL DISTRIBUTION:**

Normal distribution, also known as the Gaussian distribution, is a probability distribution that is symmetric about the mean, showing that data near the mean are more frequent in occurrence than data far from the mean. In graph form, normal distribution will appear as a bell curve.



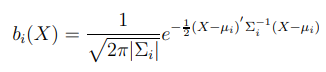
**PROBABILITY DISTRIBUTION**

A probability distribution is a statistical function that describes all the possible values and likelihoods that a random variable can take within a given range.

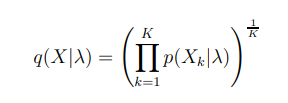
GMM is that the model is trained using unsupervised computer clustering which means that the individual Gaussian distributions are unlabeled. Therefore, we may not know which Gaussian distribution captures which features of the human voice.

The GMM is a collection of weighted Gaussian distributions λ which reflects the real distribution of mass2 . A GMM is denoted by λ = {pi , µi , Σi} i = 1, 2, ...N where pi gives the weight of i th component. Therefore, Ppi = 1. The mean and variance of the ith component are represented by µi and Σi , respectively. N represents the number of Gaussian components.

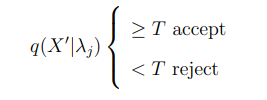
Where X is a random vector, bi(X) is probability density function of ith component explicitly given as



Given K observations of the random vector X, the probability of X following the GMM λ can be expressed as

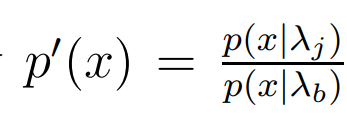


where Xk is the kth observation of X. For a known speaker j, the GMM model λj is computed such as to maximize the overall probability q(Xj |λj ). Therefore, the GMM λj provides a voice template. In GMM based biometric verification system, a two phase scenario is applied. In the enrollment phase, a feature Xj extracted from a person j, is used to generate a template GMM λj a decision function.



is computed. Where T is the pre-defined constant threshold and X′ is a fresh feature extracted from an unknown person who claims to be j. If the likelihood q(X′ |λj ) is greater than the thresh hold, the unknown person passes the verification as j otherwise the authorization fails.

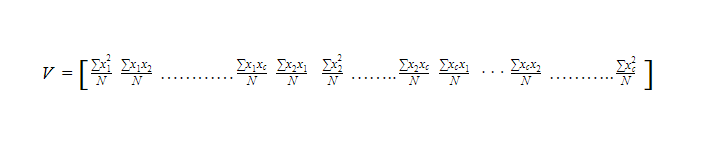
Finally, we note that a more popular version of the GMM, namely the Adapted Gaussian Mixture Model, is in use today [3]. In the Adapted GMM, a universal background model λb is generated by training with samples collected from all speakers. Afterwards, each speaker is modeled by adapting the background model. In the verification phase, the Gaussian mixture density p(x) in Equation is substituted by



The details of the adapted GMM modeling algorithm can be found in . The main advantage of the adapted GMM is that the training phase for a speaker is much faster while at the same time it gives a more accurate verification performance. In this dissertation we will base our analysis on the more popular adapted GMM.

## ****Why do we use the Variance-Covariance Matrix?****

The Covariance is a measure of how changes in one variable are associated with changes in a second variable. It’s not about the independence of variation of two variables but how they change depending on each other. The [variance-covariance matrix](https://www.mygreatlearning.com/blog/covariance-vs-correlation/) is a measure of how these variables are related to each other, and in that way it’s very similar to the standard deviation except when we have more dimension, the covariance matrix against the standard deviation gives us a better more accurate result.

[](data:image/svg+xml,%3csvg%20xmlns=)

Where, V= c x c variance-covariance matrix

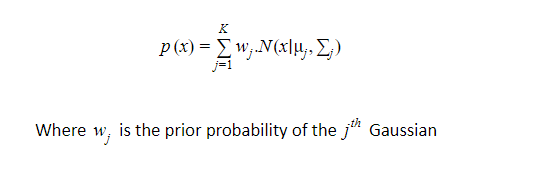
N = the number of scores in each of the c datasets

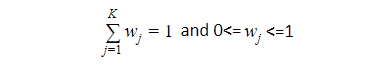
xi= is a deviation score from the ith dataset

xi2/N= is the variance of element from the ith dataset

xixj/N= is the covariance for the elements from the ith and jth datasets

and the probability given in a mixture of K Gaussian where K is a number of distributions:

[](data:image/svg+xml,%3csvg%20xmlns=)

[](data:image/svg+xml,%3csvg%20xmlns=)

Once we multiply the probability distribution function of d-dimension by W, the prior probability of each of our gaussians, it will give us the probability value X for a given X data point. If we were to plot multiple Gaussian distributions, it would be multiple bell curves. What we really want is a single continuous curve that consists of multiple bell curves. Once we have that huge continuous curve then for the given data points, it can tell us the probability that it is going to belong to a specific class.

Now, we would like to find the maximum likelihood estimate of X (the data point we want to predict the probability) i.e. we want to maximize the likelihood that X belongs to a particular class or we want to find a class that this data point X is  most likely to be part of.

It is very similar to the k-means algorithm. It uses the same optimization strategy which is the expectation maximization algorithm.

**EXPECTATION MAXIMIZATION ALGORITHM:**EM can be used for variables that are not directly observable and deduce from the value of other observed variables. It can be used with unlabeled data for its classification. It is one of the popular approaches to maximize the likelihood.

##### FEATURE EXTRACTION:

Mel-frequency cepstral coefficients (MFCC) have been shown to outperform any other Short Term Cepstrum feature extraction technique in speech recognition and later on widely used in speaker verification tasks. Similar to other cepstral features, MFCC is obtained from a speech signal through a combination of transforms . Particularly, MFCC can be carried out with the following steps.

1. Break the input into a number of time frames to be processed independently. Each frame is typically 20 − 30 ms.
2. Using a Fast Fourier Transform (FFT) compute the frequency components of each of the time frames and take the amplitude.
3. Use a number of triangular band-pass filters in order to project the frequency components of each frame into the Mel-scale.
4. Take the logarithm.
5. Apply a discrete cosine transform (DCT) on the output of the filters in order to compute the MFCC for each frame.

MFCC processing is invertible by inverting each step in the MFCC computation steps. However, because some of the MFCC processing steps are non-linear, the inversion will be a lossy process. The inversion details can be found.

##### 4. ACCURACY MEASURES

The accuracy of model is to be evaluated for training and testing. The proposed model got the 80% - 90% Accuracy.

**5.VISUALIZATION:**

Visualization is the presentation of information other visual format. It communicates relationships of data with images.

For the visualization the tkinter is used and PIL libraries.

**TKINTER:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the *Tkinter* module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user.

**PIL:**

PIL is the Python Imaging Library which provides the python interpreter with image editing capabilities. The Image module provides a class with the same name which is used to represent a PIL image. The module also provides a number of factory functions, including functions to load images from files, and to create new images.

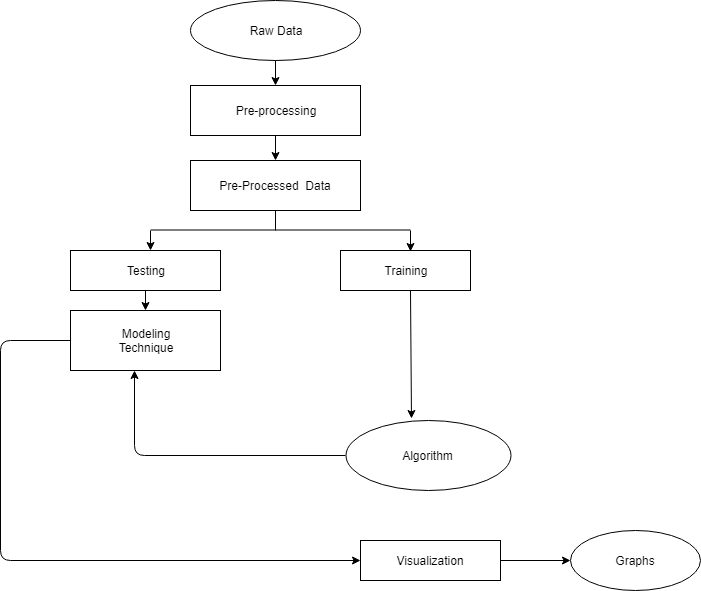
**CHAPTER -4**

**DESIGN**

System design is the process of defining the architecture, modules, interfaces and datafor a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product management.

##### 4.1 BLOCK DIAGRAM:

##### The block diagram is typically used for a higher level, less detailed description aimed more at understanding the overall concepts and less at understanding the details of implementation.



##### Figure 4.1.1 Block diagram

##### 4.2 DATA FLOW DIAGRAMS:

Data flow diagram (DFD) is a graphical representation of “flow” of data through an information system, modelling its process concepts. Often, they are a preliminary step used to create an overview of the system which can later be elaborated. DFD’s can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It doesn’t show information about timing of processes, or information about whether processes will operate in sequence or parallel. A DFD is also called a “bubble chart”.

##### DFD Symbols:

In the DFD, there are four symbols:

1. Quire Defines A Source Or Destination Of System Data.
2. An Arrow Indicates Dataflow. It Is The Pipeline Through Which The Information Flows.
3. A Circle Or A Bubble Transformsdata Flow Into Outgoing Dataflow.
4. An Open Rectangle Is A Store, Data At Reset Or At Temporary Repository Of Data.

**Dataflow:** Data moves in a specific direction from an origin to a destination.

**Process:** People, procedures or devices that use or produce (Transform) data. The physical

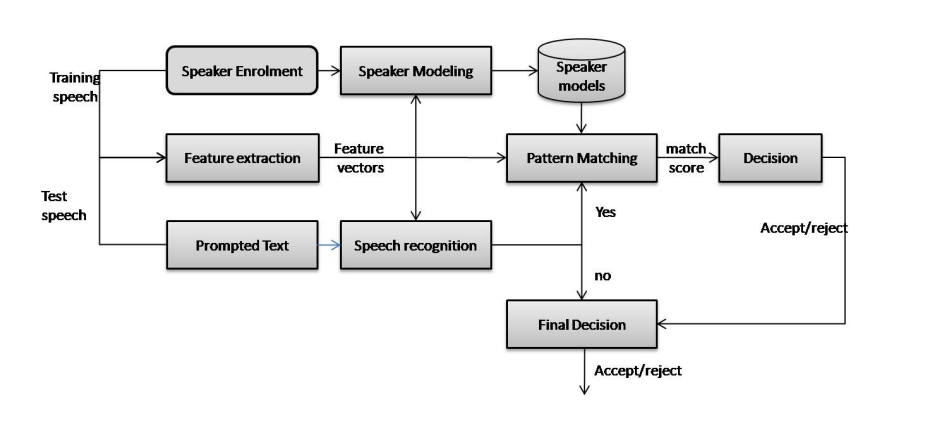
component is not identified.

**Sources**: External sources or destination of data, which may be programs, organizations or

other entities.

**Data store:** Here data is stored or referenced by a process in the systems.

In our project, we had built the data flow diagrams at the very beginning of business process modelling in order to model the functions that our project has to carry out and the interaction between those functions together with focusing on data exchanges between processes.



**Figure 4.2.1 Dataflow Diagram**

##### 4.3 UNIFIED MODELLING LANGUAGE DIAGRAMS:

The Unified Modelling Language (UML) is a Standard language for specifying, visualizing, constructing and documenting the software system and its components. The UML focuses on the conceptual and physical representation of the system. It captures the decisions and understandings about systems that must be constructed. A UML system is represented using five different views that describe the system from a distinctly different perspective. Each view is defined by a set of diagrams, which is as follows.

1. **User Model View** 
   1. This view represents the system from the user’s perspective.
   2. The analysis representation describes a usage scenario from the end user's perspective.
2. **Structural Model View** 
   1. In this model the data and functionality are arrived from inside the system.
   2. This model view models the static structures.
3. **Behavioral Model View**

It represents the dynamic of behavior as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

1. **Implementation model View**

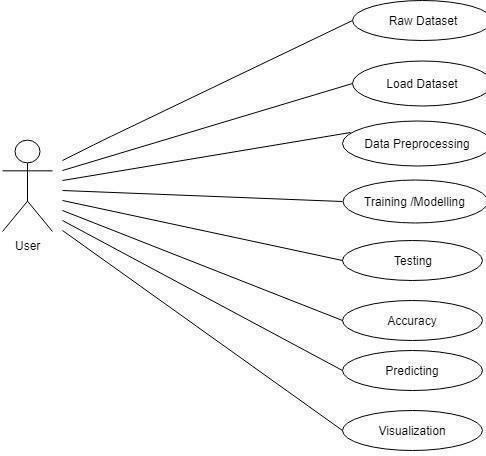
In this the structural and behavioral parts of the system are represented as they are to be built.

1. **Environmental Model View**

In this the structural and behavioral aspect of the environment in which the system is to be implemented are represented.

##### 4.3.1 USE CASE DIAGRAM:

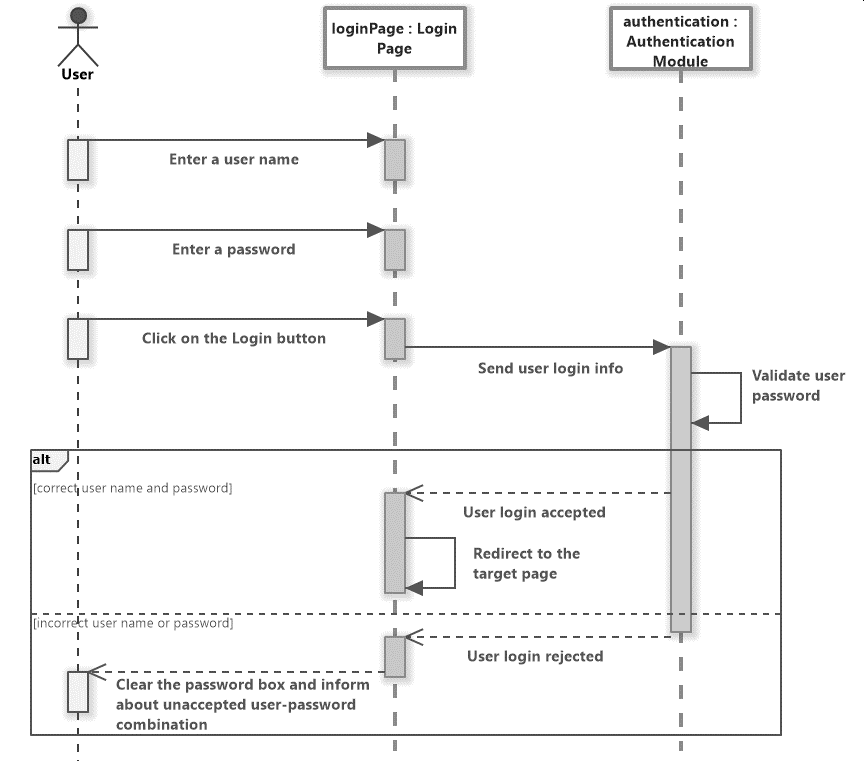
Use case diagrams are one of the five diagrams in the UML for modelling the dynamic aspects of the systems (activity diagrams, sequence diagram, state chart diagram, collaboration diagram are the four other kinds of diagrams in the UML for modelling the dynamic aspects of systems).Use case diagrams are central to modelling the behavior of the system, a sub-system, or a class. Each one shows a set of use cases and actors and relations.



##### Figure4.3.1 Use case Diagram

**4.3.2 SEQUENCE DIAGRAM:**

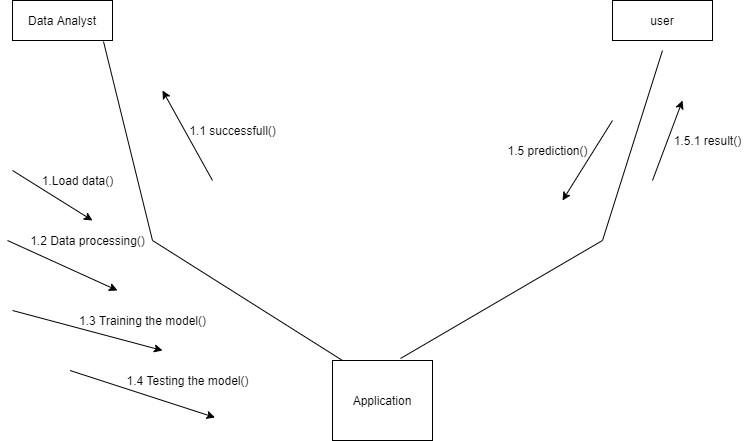
Sequence diagrams are a popular dynamic modelling solution. Dynamic modelling focuses on the interactions occurring within the system. Sequence diagrams specifically focus on the "lifelines" of an object and how they communicate with other objects to perform a function before the lifeline ends.



**Figure 4.3.2 Sequence Diagram**

**4.3.3 COLLABORATION DIAGRAM:**

Collaboration diagram is an interaction diagram that emphasizes the structural organization of the objects that send and receive messages. Collaboration diagram and sequence diagram are isomorphic*.*

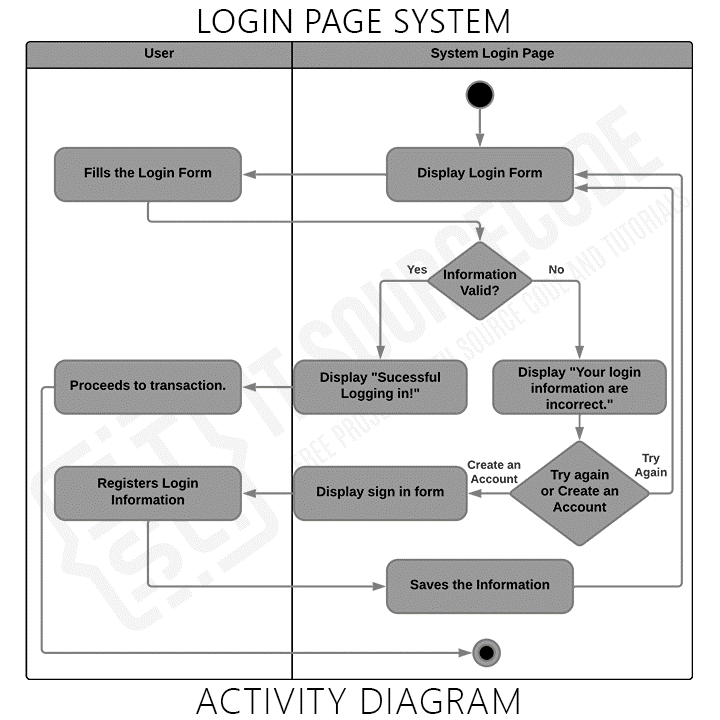


## Figure 4.3.3 Collaboration diagram

## 

## 4.3.4­ ACTIVITY DIAGRAM:

Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc.



**Figure 4.3.4 Activity diagram**

## CHAPTER - 5

## IMPLEMENTATION

## Implementation is the stage of the project when the theoretical design is turned into a new and in giving the user confidence that the new system will work and be effective.

The implementation stage involves careful planning, investigation of the existing system and it’s constraints on implementation, designing of methods to achieve changeover and evaluating the working system. Thus it can be considered to be the most critical stage in achieving a successful non of changeover methods.

The project is implemented by accessing simultaneously from more than one system and more than one window in one system. The application is implemented in the Internet Information Services 5.0 web server under Windows XP and accessed from various clients.

##### 5.1 TECHNOLOGIES USED

##### WHAT IS PYTHON?

Python is an interpreter, high-level programming language for general-purpose programming by “Guido van Rossum” and first released in 1991, Python has a design philosophy that emphasizes code readability, and a syntax that allows programmers to express concepts in fewer lines of code, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales. Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object- oriented, imperative, functional, procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open-source software and has a community-based development model, as do nearly all of its variant implementations. Python is managed by the non-profit Python Software Foundation.

Python is a general purpose, dynamic, high level and interpreted programming language. It supports object-oriented programming approaches to develop applications. It is simple and easy to learn and provides lots of high-level data structures.

1. Windows XP
2. Python Programming
3. Open-source libraries: Numpy, tkinter, PIL.

**Python Versions:**

Python 2.0 was released on 16 October 2000 and had many major new features, including a cycle-detecting, garbage collector, and support for Unicode. With this release, the development process became more transparent and community-backed.

Python 3.0 (initially called Python 3000 or py3k) was released on 3 December 2008 after a long testing period. It is a major revision of the language that is not completely backward-compatible with previous versions. However, many of its major features have been back ported to the Python 2.6.x and 2.7.x version series, and releases of Python 3 include the 2to3 utility, which automates the translation of Python 2 code to Python 3.

Python 2.7's end-of-life date (a.k.a. EOL, sunset date) was initially set at 2015, then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. In January 2017, Google announced work on a Python 2.7 to Go Trans compiler to improve performance under concurrent workloads.

Python 3.6 had changes regarding UTF-8 (in Windows, PEP 528 and PEP 529) and P[ython 3.7.0b1 (PEP 540)](https://www.python.org/dev/peps/pep-0540/) added a new "UTF-8 Mode" (and overrides [POSIX locale).](https://en.wikipedia.org/wiki/POSIX_locale)

##### WHY PYTHON?

1. Python is a scripting language like PHP, Perl, and Ruby.
2. No licensing, distribution, or development fees
3. It is a Desktop application.
4. Linux, windows
5. Excellent documentation
6. Thriving developer community
7. For us job opportunity

##### LIBRARIES OF PYTHON:

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited too many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing.

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation wsgiref follows PEP 33), but most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2018, the Python Package Index (PyPI), the official repository for third party Python software, contains over 130,000 packages with a wide range of functionality, including:

1. Graphical user interfaces
2. Web frameworks
3. Multimedia
4. Databases
5. Networking
6. Test frameworks

**5.2 WHAT IS MACHINE LEARNING?**

Machine Learning is an application of artificial intelligence (AI) that provides system 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 to learn for themselves.

##### Basics of python machine learning:

1. You'll know how to use Python and its libraries to explore your data with the help of matplotlib and Principal Component Analysis (PCA).
2. And you'll pre-processes your data with normalization and you'll split your data into training and test sets.
3. Next, you'll work with the well-known K-Means algorithm to construct an unsupervised model, fit this model to your data, predict values, and validate the model that you have built.
4. As an extra, you'll also see how you can also use Support Vector Machines (SVM) to construct another model to classify your data.

**5.3 WHY MACHINE LEARNING?**

1. It was born from pattern recognition and theory that computers can learn without being programmed to specific tasks.
2. It is a method of Data analysis that automates analytical model building.

Machine learning tasks are typically classified into two broad categories, depending on whether there is a learning "signal" or "feedback" available to a learning system. They are

computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that [maps](https://en.wikipedia.org/wiki/Map_(mathematics)) inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:

[**Semi-supervised learning:**](https://en.wikipedia.org/wiki/Semi-supervised_learning)

The computer is given only an incomplete training signal: a training set with some (often many) of the target outputs missing.

[**Active learning:**](https://en.wikipedia.org/wiki/Active_learning_(machine_learning))

The computer can only obtain training labels for a limited set of instances (based on a budget), and also has to optimize its choice of objects to acquire labels for. When used interactively, these can be presented to the user for labelling.

[**Reinforcement learning:**](https://en.wikipedia.org/wiki/Reinforcement_learning)

Training data (in form of rewards and punishments) is given only as feedback to the program's actions in a dynamic environment, such as [driving a vehicle](https://en.wikipedia.org/wiki/Autonomous_car) or playing a game against an opponent.

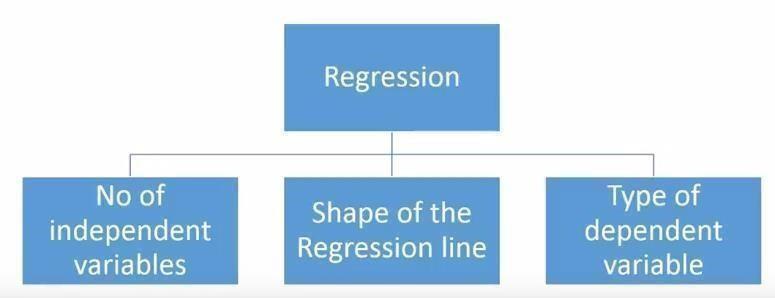
[**Unsupervised learning:**](https://en.wikipedia.org/wiki/Unsupervised_learning)

No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end ([feature learning)](https://en.wikipedia.org/wiki/Feature_learning).

In [regression,](https://en.wikipedia.org/wiki/Regression_analysis) also a supervised problem, the outputs are continuous rather than discrete.

**Regression**:

The analysis or measure of the association between one variable (the dependent variable) and one or more other variables (the independent variables), usually formulated in an equation in which the independent variables have parametric coefficients, which may enable future values of the dependent variable to be predicted.



**Figure 5.3.1 Regression Structure**

**5.4 MODULES:**

A module allows you to logically organize your Python code. Grouping related code into a module makes the code easier to understand and use. A module is a Python object with arbitrarily named attributes that you can bind and reference.

**NUMPY: -**

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided.

All this is explained with the help of examples for better understanding.

NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of arrays. Numeric, the ancestor of NumPy, was developed by Jim Humulin. Another package Numara was also developed, having some additional functionalities. In 2005, Travis Oliphant created the NumPy package by incorporating the features of Numara into Numeric package. There are many contributors to this open-source project.

##### Operations using NumPy: -

Using NumPy, a developer can perform the following operations –

1. Mathematical and logical operations on arrays.
2. Fourier transforms and routines for shape manipulation.
3. Operations related to linear algebra. NumPy has in-built functions for linear algebra and random number generation.

**SCIKIT-LEARN:**

Scikit-learn (formerly scikits. learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

The scikit-learn project started as scikits. learn, a Google Summer of Code project by David Courmayeur. Its name stems from the notion that it is a “SciKit” (SciPy Toolkit), a separately-developed and distributed third-party extension to SciPy.

The original codebase was later rewritten by other developers. In 2010 Fabian Pedrosa, Gael Viroqua, AlexandreGramfort and Vincent Michel, all from INRIA took leadership of the project and made the first public release on February the 1st 2010. Of the various scikits, scikit-learn as well as scikit-image were described as “well-maintained and popular” in November 2012. Scikit-learn is largely written in Python, with some core algorithms written in Python to achieve performance. Support vector machines are implemented by a Python wrapper around LIBSVM; logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

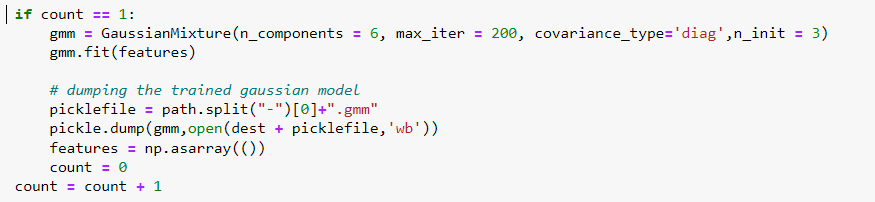
Some popular groups of models provided by scikit-learn include:

1. **Ensemble methods:** for combining the predictions of multiple supervised models.
2. **Feature extraction**: for defining attributes in image and text data.
3. **Feature selection:** for identifying meaningful attributes from which to create

supervised models.

1. **Parameter Tuning:** for getting the most out of supervised models.
2. **Manifold Learning:** For summarizing and depicting complex multi-dimensional data.
3. **Supervised Models:** a vast array not limited to generalize linear models, discriminant analysis, naive bayes, lazy methods, neural networks, support vector machines and decision trees.

Training the model by using Gaussian mixture model



**Figure 5.4.1 Gaussian Mixture Model Formula**

## TKINTER:

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

The [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) package (“Tk interface”) is the standard Python interface to the Tcl/Tk GUI toolkit. Both Tk and [tkinter](https://docs.python.org/3/library/tkinter.html#module-tkinter) are available on most Unix platforms, including macOS, as well as on Windows systems.

Python implements the Tkinter as a module. Tkinter is a wrapper of C extensions that use Tcl/Tk libraries.

Tkinter allows you to develop desktop applications. It’s a very good tool for GUI programming in Python.

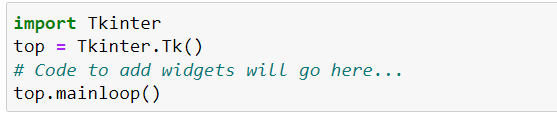
Tkinter is a good choice because of the following reasons:

1. Easy to learn.
2. Use very little code to make a functional desktop application.
3. Layered design.
4. Portable across all operating systems including Windows, macOS, and Linux.
5. Pre-installed with the standard Python library.

This tutorial assumes that you already have Python 3.x installed on your computer. If it’s not the case, you need to [install Python](https://www.pythontutorial.net/getting-started/install-python/) first.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

1. Import the *Tkinter* module.
2. Create the GUI application main window.
3. Add one or more of the above-mentioned widgets to the GUI application.
4. Enter the main event loop to take action against each event triggered by the user.



**Figure 5.4.2 Tkinter widgets**

**TKINTER WIDGETS**

Tkinter provides various controls, such as buttons, labels and text boxes used in a GUI application. These controls are commonly called widgets.

**SQL:**

Databases offer numerous functionalities by which one can manage large amounts of information easily over the web and high-volume data input and output over a typical file such as a text file. SQL is a query language and is very popular in databases. Many websites use MySQL. SQLite is a “light” version that works over syntax very much similar to SQL. SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. It is the most used database engine on the world wide web. Python has a library to access SQLite databases, called sqlite3, intended for working with this database which has been included with Python package since version 2.5. SQLite has the following features.

1. Serverless
2. Self-Contained
3. Zero-Configuration
4. Transactional

**Connecting to the Database:**

Connecting to the SQLite Database can be established using the connect() method, passing the name of the database to be accessed as a parameter. If that database does not exist, then it’ll be created.

### CHAPTER - 6

### TESTING

### It is the process of testing the functionality and it is the process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. Software testing is usually performed for one of two reasons:

1. Defect Detection
2. Reliability estimation

##### 6.1 BLACK BOX TESTING:

The base of the black box testing strategy lies in the selection of appropriate data as per functionality and testing it against the functional specifications in order to check for normal and abnormal behavior of the system. Nowadays, it is becoming easier to route the testing work to a third party as the developer of the system knows too much of the internal logic and coding of the system, which makes it unfit to test applications by the developer. The following are different types of techniques involved in black box testing. They are:

1. Decision Table Testing
2. All pairs testing
3. State transition tables testing
4. Equivalence Partitioning

Software testing is used in association with Verification and Validation. Verification is the checking of or testing of items, including software, for conformance and consistency with an associated specification. Software testing is just one kind of verification, which also uses techniques such as reviews, inspections, walk-through. Validation is the process of checking what has been specified is what the user actually wanted.

1. Validation: Are we doing the right job?
2. Verification: Are we doing the job right?

In order to achieve consistency in the Testing style, it is imperative to have and follow a set of testing principles. This enhances the efficiency of testing within SQA team members and thus contributes to increased productivity. The purpose of this document is to provide an overview of the testing, plus the techniques. Here, after training is done on the training dataset, testing is done.

**6.2 WHITE BOX TESTING:**

In designing a database the flow of specific inputs through the code, expected output and the functionality of conditional loops are tested.

At SDEI, 3 levels of software testing is done at various SDLC phases.

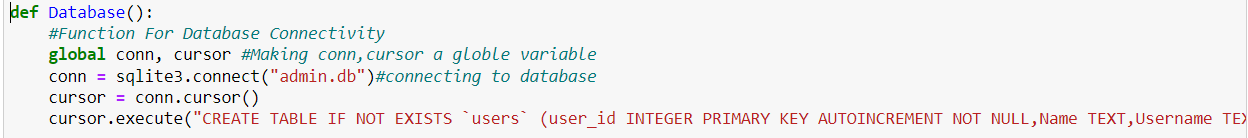
1. **UNIT TESTING**: in which each unit (basic component) of the software is tested to verify that the detailed design for the unit has been correctly implemented.
2. **INTEGRATION TESTING**: in which progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a whole.
3. **SYSTEM TESTING**: in which the software is integrated to the overall product and tested to show that all requirements are met. A further level of testing is also done, in accordance with requirements:
4. **REGRESSION TESTING**: is used to refer the repetition of the earlier successful tests to ensure that changes made in the software have not introduced new bugs/side effects.
5. **ACCEPTANCE TESTING:** Testing to verify a product meets customer specified requirements. The acceptance test suite is run against supplied input data. Then the results obtained are compared with the expected results of the client. A correct match was obtain.

**CHAPTER - 7**

**OUTPUT SCREEN**

**7.1 DATABASE:**

Connecting to database and storing the data in the csv file.

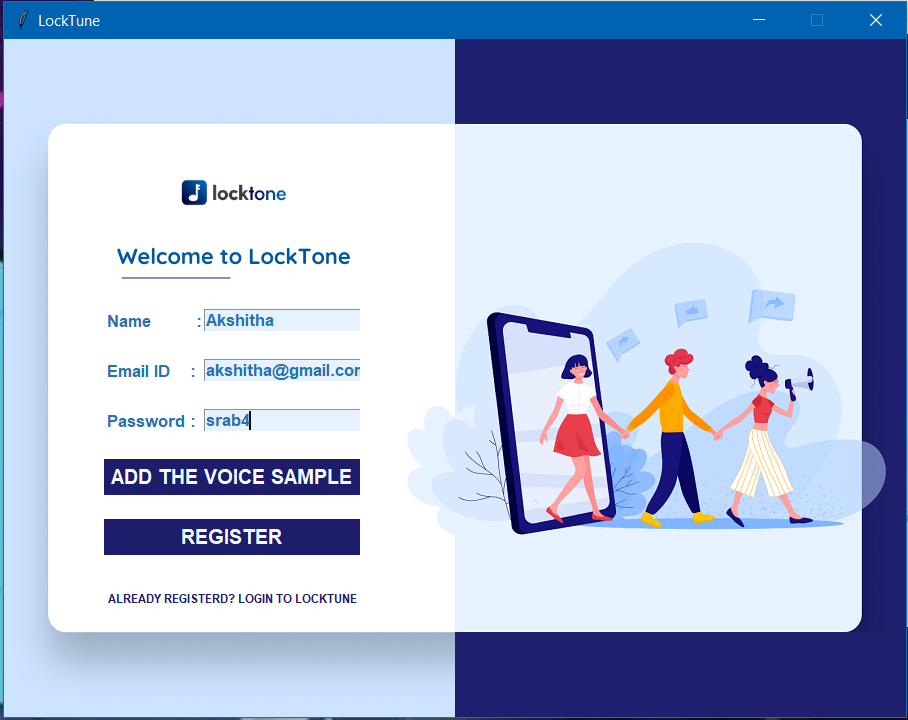
****

**Figure 7.1.1 Database Connect**

In the backend the database for storing the data and also for training, testing we have used the Gaussian mixture model and about frontend there is a library called tkinter for the user interface.

**7.2 REGISTER:**

Register window will open when we first run the code. There the user can register.

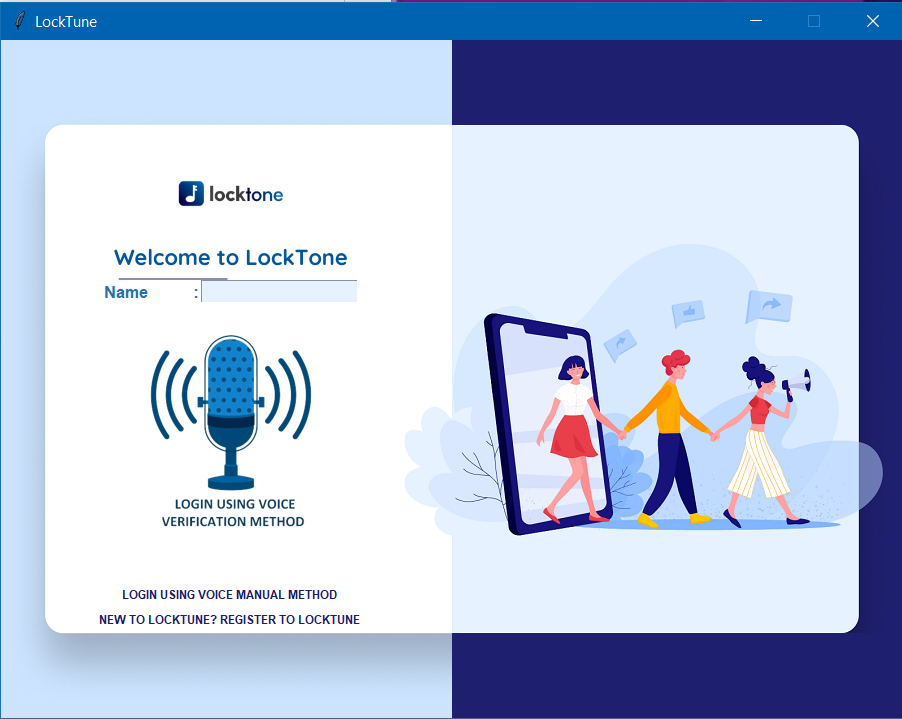


**Figure 7.2.1 Register window**

This is where user can register with their name, email, password, also can add the voice. For more security. The user can add the voice and then, register.

**7.3 LOGIN:**

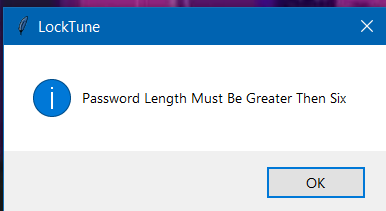
In the below window the user can login by entering the name and voice.



**Figure 7.3.1 Login window**

After the user login. The user can manipulate like adding of file, removing of file, opening file.

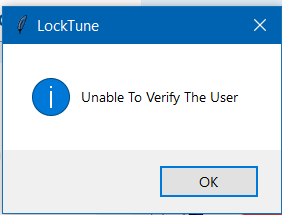
If user did not enter the given length password. The dialog box will open.



**Figure 7.3.2 Tkinter message box for password length.**

If user’s voice is not accurate or the user try to use another’s voice they cannot open.

Then this dialog will appear. This is more secure then, other biometric authentication.

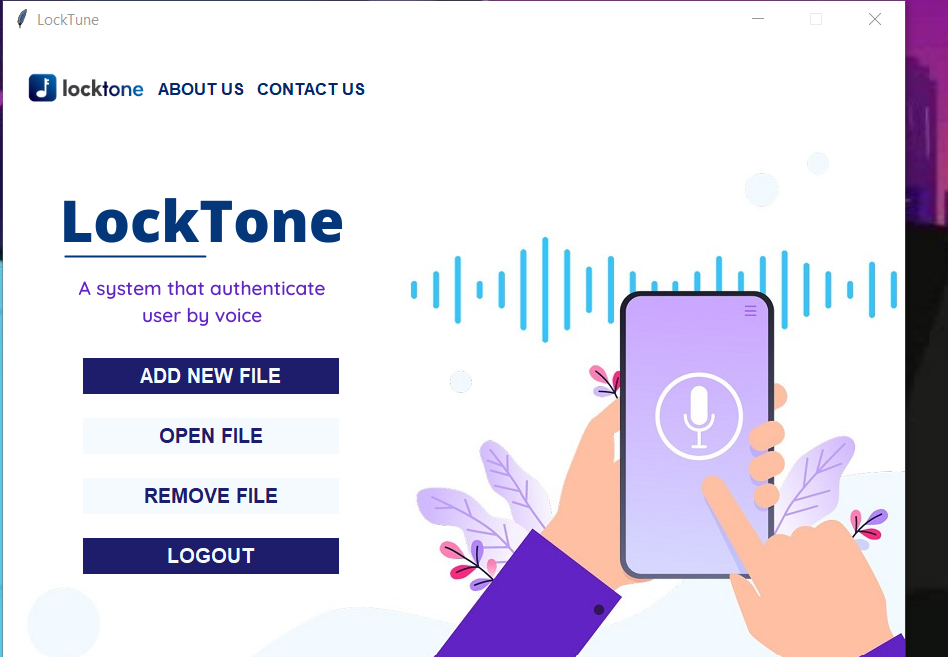


**Figure 7.3.3 Tkinter message box of Unable to verify user.**

**7.4 AFTER LOGIN:**

Once user as sign in this window will open.

Where the user can upload the file, If not the user can able to remove the file. And also can open file.



**Figure 7.4.1 Window after login**

**OPEN FILE:**

If the user click on the open file. The dialog box will open.

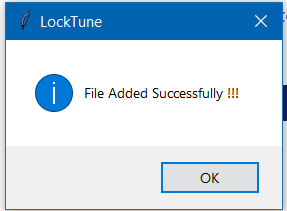
To open the user as to click on that file.

## C:\Users\sbpra\AppData\Local\Microsoft\Windows\INetCache\Content.Word\dialog box for open.png

**Figure 7.4.2 Tkinter widget for open file.**

## If the user as uploaded file successfully the dialog window will be opened.

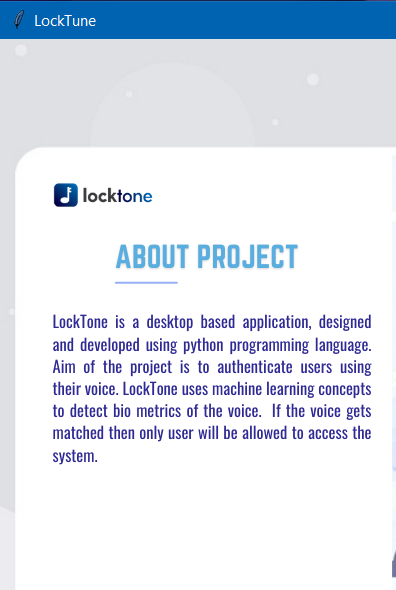
## 



**Figure 7.4.3 Tkinter message box for file upload**

## If user want to know about the. The user has to click on the about. So, Tkinter message box for file upload that the about window will be opened.

## 



**Figure 7.4.4 About window**

## If user want logout the user has to click on the logout botton. So, that the window will be closed and again comes to the login window.

## This are the results of our project.

## CHAPTER - 8

## CONCLUSION

In this paper our aim is to enhance that the biometric voice authentication is also important for security. The accuracy of using voice biometric is 80%.

The system is designed to work in an indoor environment with an average amount of noise. The presence of background noise to a large extent might disturb either the Training or the Authentication process. It also depends on the microphone that is used. If it does not capture the background noise, there won't be a hindrance to the phases. But the background noise should be minimal in order for the system to work properly.

The results we got was acuurate. Voice authentication could offer banks a number of advantages over existing verification methods, including greater security and a more convenient user experience.

However, it may not be wise to use voice on its own as a complete replacement to current verification methods.

### CHAPTER - 9

### FUTURE SCOPE

We will keep working on training the model to get a better accuracy giving more voices so that we can increase the performance .And in the future we can train the model with the latest ML algorithms.

### CHAPTER - 10

### BIBLIOGRAPHY

### <https://www.aware.com/voice-authentication/>

### <https://www.sensory.com/face-voice-biometrics/>

### <https://www.biometricupdate.com/202103/third-party-test-shows-id-rd-leap-forward-in-voice-biometrics-accuracy>

### <https://softjourn.com/insights/security-considerations-in-voice-authentication>

### <https://biz30.timedoctor.com/voice-authentication/>

### <https://www.claysys.com/blog/voice-biometric-authentication>

### <https://www.claysys.com/blog/voice-biometric-authentication/>

### <https://www.phonexia.com/blog/7-benefits-of-voice-biometric-authentication-in-call-centers/>