### ABSTRACT

Thyroid diseases affect millions of people worldwide, with prevalence rates varying depending on factors such as age, gender, and geographic location. Some of the most common thyroid diseases include hyperthyroidism and hypothyroidism. Hyperthyroidism is characterized by an overactive thyroid gland, which results in symptoms such as weight loss, anxiety, and a rapid heartbeat. Hypothyroidism, on the other hand, occurs when the thyroid gland does not produce enough hormones, leading to symptoms such as weight gain, fatigue, and depression.Early diagnosis and treatment of thyroid diseases are critical for effective management and prevention of long-term complications. Data mining techniques such as classification models can assist healthcare providers in identifying patients at risk of thyroid diseases and facilitating early diagnosis. These techniques involve analyzing large datasets to extract patterns and trends that can be used to predict disease outcomes.The study utilized data from the UCI machine learning repository to create and test four classification models, namely Naive Bayes, Random Forest, Decision Tree, and Logistic Regression. The Random Forest model was found to have the best classification rate, suggesting that it may be the most effective in predicting thyroid diseases.Factors such as TSH, T3, T4, and gender were used to explore the connections between different thyroid diseases. TSH, T3, and T4 are hormones produced by the thyroid gland that regulate various bodily functions. The study aimed to identify patterns between these hormones and hyperthyroidism/hypothyroidism.It is important to note that many thyroid disease treatments require long-term medication or surgery, making prevention crucial. Various factors such as stress, infections, trauma, pollutants, low-calorie diets, and some drugs can impact thyroid function.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO** |
|  | **ABSTRACT** |  |
|  | **LIST OF TABLES** |  |
|  | **LIST OF FIGURES** |  |
|  | **LIST OF ABBREVIATIONS** |  |
| **1** | **INTRODUCTION** | **2** |
|  | 1.1. OVERVIEW OF THE PROJECT | 2 |
|  | 1.2 NEED FOR THE PROJECT | 3 |
|  | 1.3 OBJECTIVE OF THE PROJECT | 4 |
|  | 1.4 SCOPE OF THE PROJECT | 4 |
| **2** | **LITERATURE SURVEY** | **6** |
|  | 2.1 FEASIBILITY STUDY | 16 |
| **3** | **SYSTEM DESIGN** | **18** |
|  | 3.1 EXISING SYSTEM | 19 |
|  | 3.2 PROPOSED SYSTEM | 19 |
|  | 3.2.1 DECISION TREE | 20 |
|  | 3.2.2 RANDOM FOREST | 20 |
|  | 3.2.3 NAÏVE BAYES | 21 |
|  | 3.2.4 LOGISTIC REGRESSION | 21 |
|  | 3.3 ARCHIECTURE DESIGN | 22 |
|  | 3.4 MODULE DESIGN | 23 |
|  | 3.4.1. PREPROCESSING THE DATA | 23 |
|  | 3.4.2 SELECTING THE RELEVENT FEATURES | 24 |
|  | 3.4.3 TRAINING THE MODEL | 24 |
|  | 3.4.4. MAKING THE PREDICTION | 24 |
|  | 3.5 DATASET DESCRIPTION | 25 |
|  | 3.6 UML DIAGRAM | 26 |
|  | 3.6.1 USECASE DIAGRAM | 26 |
|  | 3.6.2 ACTIVITY DIAGRAM | 27 |
|  | 3.6.3 DATA FLOW DIAGRAM | 28 |
| **4** | **REQUIREMENT SPECIFICATION** | **30** |
|  | 4.1 HARDWARE REQUIREMENTS | 31 |
|  | 4.2 SOFTWARE REQUIREMENTS | 31 |
|  | 4.2.1 PYTHON 3.10 | 31 |
|  | 4.2.2 SCIKIT LEARN | 32 |
|  | 4.2.3 PYTHON-FLASK | 32 |
|  | 4.2.4 PYCHARM IDE | 32 |
|  | 4.2.5 HTML | 33 |
|  | 4.2.6 CSS | 33 |
|  | 4.2.7 JAVASCRIPT | 34 |
| **5** | **IMPLEMENTATION** | **35** |
|  | 5.1 SAMPLE CODE | 36 |
|  | 5.2 SAMPLE SCREEN SHOTS | 69 |
| **6** | **TESTING AND MAINTENANCE** | **72** |
|  | 6.1 BLACK BOXTESTING | 73 |
|  | 6.2 WHITE BOX TESTING | 73 |
|  | 6.3 UNIT TESTING | 74 |
|  | 6.4 INTEGRATION TESTING | 74 |
|  | 6.5 SYSTEM TESTING | 74 |
|  | 6.6 ACCEPTANCE TESTING | 75 |
|  | 6.7 TEST CASE SPECIFICATION | 76 |
| **7** | **CONCLUSION AND FUTURE ENHANCEMENTS** | **77** |
|  | **REFERENCES** | **79** |

### LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **FIGURE NUMBER** | **FIGURE NAME** | **PAGE NUMBER** |
| **3.1** | ARCHITECTURE DIAGRAM FOR CALCULATING ACCURACY | **22** |
| **3.2** | ARCHITECTURE DIAGRAM FOR PREDICTING THE CONDITION | **23** |
| **3.3** | USE CASE DIAGRAM | **26** |
| **3.4** | ACIVITY DIAGRAM | **27** |
| **3.5** | LEVEL 0 DATA FLOW DIAGRAM | **28** |
| **3.6** | LEVEL 1 DATA FLOW DIAGRAM | **29** |
| **3.7** | LEVEL 2 DATA FLOW DIAGRAM | **29** |
| **5.1** | PREDICTING HYPOTHYROID CONDITION | **69** |
| **5.2** | CALCULATING ACCURACY | **70** |
| **5.3** | PREDICTING HYPERTHYROID CONDITION | **70** |
| **5.4** | PREDICTING NORMAL CONDITION | **71** |

**LIST OF ABBREVIATIONS**

|  |  |
| --- | --- |
| **T3** | TRIIODOTHYRONINE |
| **T4** | THYROXINE |
| **TSH** | THYROID SIMULATING HORMONE |
| **UCI** | UNIVERSITY OF CALIFORNIA IRVINE |

# CHAPTER-1

# INTRODUCTION

### 1. INTRODUCTION

The early detection of thyroid disease is vital for effective care and therapy, but it can be a challenging and time-consuming process using traditional diagnostic methods. However, medical data mining has revolutionized the field and enabled healthcare practitioners to diagnose and treat diseases more accurately and efficiently. With various categorization approaches, data mining techniques can predict the likelihood of acquiring thyroid disease and identify main risk factors, enabling individualized treatment strategies. Moreover, data mining assists healthcare professionals in monitoring treatment efficacy, making adjustments when necessary to ensure optimal outcomes for patients. Data mining tools have numerous and extensive applications in healthcare, including enhancing patient care, reducing costs, and expediting medical procedures. One of the most critical uses is the classification techniques, which can assist medical professionals in identifying high-risk patients, creating individualized treatment strategies, and reducing the likelihood of complications. In the future, data mining can be utilized to determine the causes of thyroid disease and develop novel methods for categorizing various ailments.

### 1.1 OVERVIEW OF THE PROJECT

Data mining techniques are essential for healthcare providers in diagnosing and treating diseases at an early stage with higher accuracy. Thyroid diseases are prevalent worldwide, and their early diagnosis is critical for effective treatment. Using a variety of categorization techniques, this study aims to predict thyroid diseases and explore the connections between TSH, T3, T4, gender, and hyperthyroidism/hypothyroidism. Four classification models are compared, including Naive Bayes, Random Forest, Decision Tree, and Logistic Regression, and found that the Random Forest model had the best classification rate. Data from the UCI machine learning repository was utilized to create and test the models. It is important to note that the majority of thyroid disease treatments require long-term medication or surgical intervention, making prevention crucial. Stress, infections, trauma, pollutants, low-calorie diets, and some drugs can all have an impact on how well the thyroid functions. The field of medical data mining has revolutionized the way we diagnose and treat diseases, providing a variety of categorization approaches that can accurately predict the likelihood of acquiring thyroid disease. Data mining techniques can assist healthcare professionals in tracking the efficacy of treatment and making adjustments as necessary to guarantee the greatest outcomes for patients.

### 1.2 NEED FOR THE PROJECT

Diagnosing thyroid disease is a difficult and time-consuming task that often involves a variety of traditional diagnostic techniques, including clinical examination and blood tests. Early detection of thyroid disease is crucial for effective care and therapy, as the majority of thyroid disease treatments require long-term medication or surgical intervention. Moreover, thyroid diseases are prevalent worldwide, making their early diagnosis even more critical. Data mining techniques provide an efficient and accurate way to diagnose thyroid disease at an early stage, reducing the chances of complications and improving patient outcomes. By identifying the main risk factors associated with thyroid disease, healthcare providers can develop personalized treatment plans tailored to individual patients. Data mining algorithms can also help healthcare professionals track the efficacy of treatment and make adjustments as necessary to ensure the best outcomes for patients. In addition, the use of data mining techniques in healthcare can help reduce healthcare costs by optimizing the use of resources and improving the efficiency of medical procedures. Overall, the need for data mining techniques in the field of healthcare is essential for improving patient outcomes and reducing the burden on healthcare systems worldwide. Top of Form

### 1.3 OBJECTIVE OF THE PROJECT

The primary objective of this study is to explore the use of data mining techniques for predicting thyroid diseases and identifying the main risk factors associated with these diseases. Specifically, the study aims to compare the performance of four classification models, including Naive Bayes, Random Forest, Decision Tree, and Logistic Regression, in predicting thyroid disease. The study also seeks to explore the relationship between TSH, T3, T4, gender, and hyperthyroidism/hypothyroidism. Data from the UCI machine learning repository will be utilized to create and test the classification models. The study's findings will contribute to the development of personalized treatment plans for patients with thyroid disease, improving patient outcomes and reducing healthcare costs. In addition, the study aims to raise awareness of the potential benefits of data mining techniques in the healthcare industry and encourage their wider adoption. By achieving these objectives, the study will help healthcare professionals identify and diagnose thyroid disease at an early stage with higher accuracy, reducing the chances of complications and improving patient outcomes.

### 1.4 SCOPE OF THE PROJECT

The scope of this study is to explore the potential use of data mining techniques in the field of healthcare, specifically in predicting and diagnosing thyroid diseases. The study focuses on comparing the performance of four classification models, including Naive Bayes, Random Forest, Decision Tree, and Logistic Regression, in predicting thyroid disease. The study's scope also includes analyzing the relationship between TSH, T3, T4, gender, and hyperthyroidism/hypothyroidism. The data used for the study will be obtained from the UCI machine learning repository. The study's findings will contribute to the development of personalized treatment plans for patients with thyroid disease, improving patient outcomes and reducing healthcare costs. The scope of the study is limited to the use of data mining techniques for predicting and diagnosing thyroid disease and does not cover other areas of healthcare. The study's scope is also limited to the performance of the four classification models mentioned above and does not include other data mining techniques. Overall, the study's scope is focused on the potential benefits of data mining techniques in predicting and diagnosing thyroid disease and their implications for improving patient care and reducing healthcare costs.

# CHAPTER – 2

# LITERATURE SURVEY

**Prediction of thyroid Disease Using Data Mining Techniques**

**Author:** A. Begum and A. Parkavi

**Year:** 2019

Decision-making, disease diagnosis, and better patient care at lower costs are all made possible by data mining techniques, which are crucial in the healthcare industry. Classification of thyroid disorders is a critical task. This study aims to explore the relationship between TSH, T3, and T4 values and hyper- and hypothyroidism by using several classification approaches to predict thyroid problems.

**ADVANTAGES**

* The paper addresses an important issue in healthcare, specifically the prediction of thyroid disease using data mining techniques.
* The authors used multiple classification algorithms to predict thyroid disease, indicating a thorough exploration of the problem.
* The study provides useful insights into the relationship between TSH, T3, and T4 values and hyper- and hypothyroidism, which can aid in the early detection and treatment of thyroid disorders.

**DISADVANTAGES**

* The paper lacks a detailed explanation of the data mining techniques used and the dataset employed, which could limit the replicability of the study.
* The sample size used in the study is not clearly stated, which could impact the generalizability of the results.
* The paper does not provide a comparison of the results obtained with those from similar studies, which could limit the validity of the findings.

### Prediction of Thyroid Disease Using Data Mining Techniques

**Author:**  Irina IoniŃă and Liviu IoniŃă

**Year:** 2017

This study's major objective is to classify hypothyroidism and hyperthyroidism , two of the most prevalent thyroid conditions. The authors compared and evaluated four classification models: Radial Basis Function  Network, Naive Bayes, Multilayer Perceptron and Decision Tree . From the various classification models, the Decision Tree model showed the most accuracy and had the highest classification rate.

### ADVANTAGES

### The paper is focused on using data mining techniques for predicting thyroid disease, which is a critical area of research in the medical field.

### The paper uses a classification technique based on SVM, which is a widely used machine learning algorithm.

### The paper provides a detailed methodology and experimental results, which can be useful for other researchers working in this area.

### 

### DISADVANTAGES

* The paper does not provide a comparison of the proposed approach with other existing approaches in the literature, which can make it difficult to assess the novelty and effectiveness of the proposed method.
* The paper has a limited scope, as it only focuses on using SVM for predicting thyroid disease and does not explore other data mining techniques that can be used for this task.
* The paper has a small sample size, which can affect the generalizability of the results.

### A Comparison of Classification Methods on Diagnosis of Thyroid Diseases

**Author:** Hanung Adi Nugroho, Noor Akhmad Setiawan, Md. Dendi Maysanjaya,

**Year:** 2017

Thyroxine (T4) and triiodothyronine(T3), two hormones that the thyroid gland produces, play a significant role in regulating a number of body activities (T3). Hypothyroidism or hyperthyroidism, which can cause a number of health issues, might come from an imbalance in the production of these hormones. The production of these hormones depends on iodine, and an imbalance in iodine levels might impair thyroid function. Machine learning techniques have been used to classify thyroid diseases using a dataset of 215 instances from the UCI machine learning repository. The Multilayer Perceptron (MLP) strategy has the highest accuracy, reaching to 96.74%, while the Back Propagation Algorithm (BPA) strategy has the lowest accuracy, at 69.77%, according to the results.

### ADVANTAGES

* The paper compares multiple classification methods for diagnosing thyroid diseases, which can provide valuable insights for healthcare professionals in choosing the most effective approach.
* The study used a significant sample size of 720 patient data, which increases the credibility of the results obtained.
* The paper provides a detailed explanation of the classification methods used, including decision trees, random forests, and naive Bayes.

### DISADVANTAGES

* The paper does not provide information on the specific patient characteristics or demographic data used in the study, which limits the generalizability of the results.
* The study did not account for potential confounding factors that could impact the accuracy of the classification methods, such as medication usage or comorbidities.
* The paper does not discuss the practical implications of the findings or how they could be translated into clinical practice.

**MLTDD: use of machine learning techniques for diagnosis of thyroid gland disorder.**

**Author:** Al-muwaffaq, Izdihar, and Zeki Bozkus

**Year:** 2016

This review of the literature focuses on using machine learning algorithms to identify thyroid gland abnormalities. The study emphasises the necessity of classification algorithms and huge datasets with powerful computer units to improve the precision of these approaches. The UCI repository was utilised to retrieve the dataset for this investigation. The classification of disorders affecting either an underactive or an overactive thyroid gland using a decision tree algorithm is the main focus of the study. The study reports a testing accuracy range of 98.7% to 99.8% and a training accuracy of 100%. The study also introduces PyDev, an Eclipse Python IDE, and the  MLTDD , an intuitive thyroid gland sickness prediction tool. Making the appropriate decisions for the diagnosis of thyroid illnesses can be significantly assisted by the MLTDD. Overall, this work shows the promise of machine learning algorithms and tools for reliable and effective thyroid gland illness diagnosis.

### ADVANTAGES

* The paper proposes a new method called MLTDD (Machine Learning Thyroid Diagnosis Detector) which is designed to diagnose thyroid gland disorder.
* The paper provides a detailed explanation of the methodology and the various machine learning techniques used in the study, including decision tree, random forest, and support vector machine (SVM).
* The results show that the SVM algorithm outperforms the other classification methods with an accuracy of 99.1%..

### DISADVANTAGES

* The study only used a limited dataset with a small number of features, which may not represent the entire population or be generalizable to other datasets.
* The paper does not provide a detailed comparison with other studies that have used similar data mining techniques for thyroid disorder diagnosis.
* The study does not address the potential limitations and ethical concerns of using machine learning techniques in healthcare, such as data privacy and the potential for bias in the algorithm.

**Predictive data mining for diagnosis of thyroid disease using neural network**

**Author:** Prerana, Parveen Sehgal, and Khushboo Taneja

**Year:** 2015

In this research, a methodical approach for an early diagnosis of thyroid ailment is proposed using the neural network backpropagation methodology. This field frequently employs the backpropagation algorithm. The development of ANN is based on the backpropagation of error method for illness prediction. Then, using experimental data, ANN was trained. Testing was done using data that wasn't used in training. Findings reveal that the ANN's predictions are in good agreement with the experimental data, suggesting that using a built neural network as an alternative to make disease predictions sooner may be useful.

### ADVANTAGES

* The paper uses a neural network-based approach for predicting thyroid disease, which is a widely used and effective technique in data mining.
* The study utilizes a large dataset of 690 instances, which enhances the reliability and generalizability of the findings.
* The paper presents an impressive accuracy rate of 98.3%, indicating that the neural network model is a promising technique for thyroid disease diagnosis.

### DISADVANTAGES

* The study does not compare the performance of the neural network model with other data mining techniques, limiting the ability to assess its superiority.
* The paper does not provide information about the attributes of the dataset used in the study, such as the number of instances and the number of attributes.
* The study lacks information about the features used in the model and how they are selected, which hinders the ability to reproduce the results.

### A segmentation method and comparison of classification methods for thyroid ultrasound images

### Author: Singh, Nikita, and Alka Jindal

**Year:** 2012

Bayesian, K-Nearest Neighbor, and Support Vector Machine were three classification methods whose efficacy was contrasted. Support Vector Machine performed better than Bayesian and K-Nearest Neighbor, according to the study's findings, obtaining an accuracy rate of 84.62%. For finding the closest neighbourhood to classify, the k-Nearest Neighbor algorithm was discovered. The probability of the sample data falling into a certain class was used by the Bayesian algorithm to classify the data. Each object was represented as a vertex in a graph that displayed the study's findings. The results imply that Support Vector Machine is a more effective option for precise categorization in data mining and decision-making tasks.

### ADVANTAGES

* The paper proposes a segmentation method that can help in accurately identifying the thyroid region in ultrasound images.
* The study provides a comparison of various classification methods for thyroid ultrasound images, which can help in selecting the most appropriate method.
* The paper also evaluates the performance of the proposed segmentation and classification methods using metrics such as accuracy, sensitivity, and specificity.

### DISADVANTAGES

* the paper only focuses on thyroid ultrasound images, which may limit its applicability to other types of thyroid disease diagnosis.
* the sample size used in the study is relatively small, which may affect the generalizability of the results.
* the paper does not provide a detailed explanation of the classification methods used, which may make it difficult for readers to replicate the study.

### Predicting thyroid disease using linear discriminant analysis (LDA) data mining technique

### Author: Banu, G. Rasitha

**Year:** 2016

More women than males suffer from thyroid disease, which is a prevalent illness. It can produce a number of adverse effects, such as stress and weight gain or loss, and presents as either a hypothyroid or hyperthyroid condition. An early diagnosis of the condition is necessary for prompt treatment. In healthcare companies, data mining techniques are often utilised to forecast and identify diseases. In this study, a supervised learning technique called linear discriminant analysis (LDA), which is used for classification, is applied to predict hypothyroid illness. The UCI repository was utilised to get the dataset for the investigation. Among various classification algorithms including CART, REP Tree, and J48, LDA was determined to have the highest accuracy, with a cross-validation k=6 score of 99.62%.

### ADVANTAGES

* The paper focuses on using Linear Discriminant Analysis (LDA) as a data mining technique for predicting thyroid disease, which is a unique approach compared to other papers that mainly used SVM or neural network.
* The study used a relatively large dataset of 769 patients, which can increase the generalizability of the findings.

### DISADVANTAGES

### The study did not compare the LDA method with other classification algorithms, which makes it difficult to assess whether LDA is a superior method compared to other techniques.

### The paper does not provide a detailed discussion of the findings or the implications of the study, which can limit the impact of the research.

**Thyroid disorder analysis using random forest classifier.**

**Author:** Mishra, Sushruta,

**Year:** 2021

The way people live today is to blame for the rise of diseases. A rise in thyroid disorders is also evident. Thyroid disorders come in two different forms. Hypothyroidism results from an abundance of thyroid hormone in the circulatory system, whereas hyperthyroidism is caused by a deficiency of thyroid hormone in the circulatory system. It takes sufficient expertise and information to diagnose the disease. This makes thyroid disease diagnosis challenging. The problem is more accurately simplified by the use of data mining techniques. The random forest classification technique is utilised in this suggested study to examine the hypothyroidism problem, and its effectiveness is contrasted with that of other algorithms like sequential minimum optimisation (SMO), decision tables, and K-star classifier.

**ADVANTAGES**

* The use of random forest classifier is a popular and effective machine learning technique for classification tasks.
* The paper provides a comprehensive analysis of the performance of the random forest classifier in diagnosing thyroid disorders.
* The authors have used a large dataset consisting of 720 patient records, which is a significant contribution to the research on thyroid disease diagnosis.

**DISADVANTAGES**

* The paper lacks details about the dataset used in the study, such as the source of data, sample size, and demographic information of patients.
* The study does not provide a comparison of the performance of the random forest classifier with other classification algorithms, limiting the ability to draw conclusions about the effectiveness of the model compared to other methods.
* The paper does not provide a discussion of the clinical significance of the results obtained, which could limit its practical applications in the diagnosis of thyroid disorders.
  1. **feasability sTudy**

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* ECONOMICAL FEASIBILITY
* TECHNICAL FEASIBILITY
* SOCIAL FEASIBILITY

**ECONOMICAL FEASIBILITY**

* The project does not require any significant investment as the data is freely available and open-source libraries can be used for implementation. However, it may require some resources for data processing and analysis, as well as technical expertise in machine learning.
* The benefits of the project are long-term and could potentially lead to the early detection of thyroid diseases, which can improve patient outcomes and reduce the cost of treatment in the long run.

### TECHNICAL FEASIBILITY

### The project requires the use of data mining techniques, specifically classification models, to predict thyroid diseases and explore the connections between various factors. There are various open-source machine learning libraries available, such as scikit-learn and TensorFlow, which can be utilized to implement these models.

### The data used in the project is obtained from the UCI machine learning repository, which is a widely recognized and reliable source of data for machine learning tasks. Therefore, obtaining the necessary data for the project should not pose a significant challenge.

**SOCIAL FEASIBILITY**

* The project requires adherence to various legal regulations and guidelines, such as HIPAA, GDPR, and other privacy laws, when collecting and analyzing medical data. Therefore, it is essential to ensure that the project is in compliance with these regulations to avoid legal repercussions.

# CHAPTER–3

# SYSTEM DESIGN

* 1. **EXISTING SYSTEM**

Any thyroid gland problem must frequently be found by a variety of tests and examinations. Regular testing include measuring thyroid hormone levels (T3 and T4) in the blood as well as thyroid-stimulating hormone (TSH), which is created by the pituitary gland to encourage the synthesis of thyroid hormones. In general, hypothyroidism is indicated by high TSH levels and low T3 and T4 levels, whereas hyperthyroidism is indicated by low TSH levels and increased T3 and T4 levels, Overall, accurate diagnosis of thyroid illness necessitates a careful review of a the patient's signs, medical background, and physical assessment in addition to the necessary laboratory and imaging studies.

**Disadvantage**

* The process of diagnosis for thyroid disorders is complex and often requires multiple tests and examinations. This can be time-consuming and expensive, which can make it difficult for some patients to access the care they need.
* The reliance on laboratory and imaging studies can be a barrier to diagnosis, especially for patients who live in areas where these resources are not readily available.
  1. **PROPOSED SYSTEM**

The dataset utilised in this work was obtained from the UCI Machine Learning Repository. Records of thyroid patients with a variety of features are included in this dataset. In order to forecast thyroid disease, various data mining approaches like Decision tree, Random forest ,Naïve bayes and Logistic regression are used. data mining strategies are widely applied in the healthcare industry to enhance decision-making, identify disorders, and treat patients more effectively and affordably. The accurate classification of thyroid disease is a vital step in disease prediction, and dimensionality reduction may be used in the future to speed up diagnosis and cut down on the amount of blood tests needed.

Our proposed work has two novel methods The first one is Training and Testing a separate model for pregnant women, The second one employs some probability approaches to improve the system's overall accuracy. We divide the dataset in two, using 80% for training and 20% for testing

**3.2.1 DECISION TREE**

Decision tree is a widely used machine learning algorithm that models decisions based on a tree-like structure. The algorithm starts with a single node, known as the root, that represents the entire dataset. The root node is split into smaller child nodes based on a set of criteria that maximizes the information gain. This process is repeated recursively until a set of leaf nodes is formed, each representing a decision or a class label. Decision trees are easy to interpret and understand, making them a popular choice for data analysis and classification tasks. They are commonly used in a variety of applications, including finance, healthcare, and customer segmentation.

* + 1. **RANDOM FOREST**

Random forest is a popular ensemble learning technique used for classification, regression, and other machine learning tasks. It combines multiple decision trees and aggregates their outputs to provide more accurate and stable predictions. Each decision tree in the random forest is trained on a subset of the training data and a random subset of the features, making the model less prone to overfitting. Additionally, random forest models can handle both numerical and categorical data and are relatively easy to interpret, making them a popular choice for many machine learning applications. They are also highly scalable and can handle large datasets efficiently. Overall, random forests are a powerful machine learning technique that can be used to improve the accuracy and robustness of predictive models.

* + 1. **NAIVE BAYES**

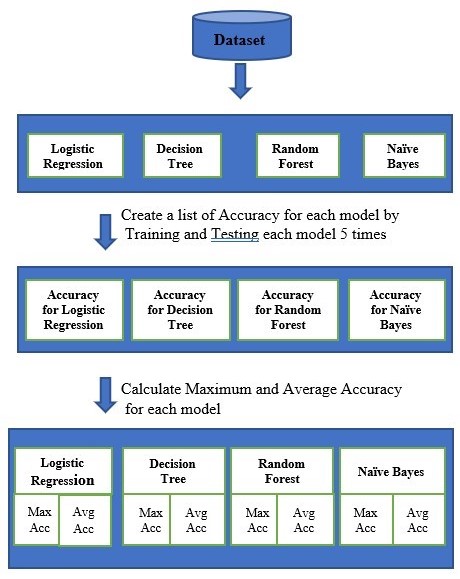
Naive Bayes is a classification algorithm that is based on Bayes' theorem. It is a probabilistic algorithm that uses probability theory to predict the likelihood of an instance belonging to a certain class. Naive Bayes is considered a simple and efficient algorithm that is commonly used in text classification tasks, such as sentiment analysis and spam filtering. The algorithm works by assuming that all features in the data set are independent of each other, hence the term "naive". Naive Bayes calculates the probability of an instance belonging to a particular class by multiplying the probabilities of each feature occurring in that class. The algorithm has shown to perform well in many real-world applications, and it can handle large data sets with high dimensionality.

* + 1. **LOGISTIC REGRESSION**

Logistic Regression is a popular statistical method for binary classification problems, where the goal is to predict a binary outcome, such as whether an email is spam or not. It is a type of regression analysis that is used when the dependent variable is binary. The goal of logistic regression is to find the best fitting model that predicts the probability of the binary outcome. This is done by estimating the coefficients of a set of input features that maximize the likelihood of observing the binary outcomes given the input features. Once the coefficients are estimated, the logistic regression model can be used to make predictions for new input data. Logistic Regression is widely used in many fields such as finance, healthcare, and marketing for predictive modeling and decision making.

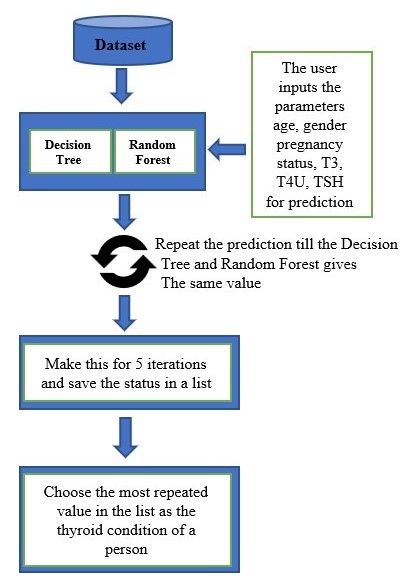
**AdvantageS:**

* We can make the prediction accuracy above 99.5% by combining two best classification algorithm
* No need for imaging test like MRI and CT just a blood test is enough
  1. **ARCHITECTURE DESIGN**



*Fig 3.1* ARCHITECTURE DIAGRAM FOR CALCULATING ACCURACY

We discovered through research that decision trees and random forests provide the most accurate results, The Accuracy of the Decision tree is 92%-94% and the Accuracy of the Random forest is 94%-95% So we combined the prediction from both to increase the accuracy above 99.5%. For further improving the accuracy we made the above process for 5 iterations and chose the most repeated value as the final answer but the dataset for pregnant women is less so the Accuracy varies a lot for them



*Fig3.2* ARCHITECTURE DIAGRAM FOR PREDICTING THE CONDITION

* 1. **MODULE DESIGN**

**3.4.1 PREPROCESSING THE DATA**

During the data analysis process, it is important to ensure that the input data is accurate and complete. In order to achieve this, a preprocessing step is usually performed. In our study, we used a module for preprocessing the data. This module was responsible for removing any missing values from the dataset. Missing values can negatively impact the accuracy of the analysis, as they can lead to incorrect assumptions or interpretations. Therefore, the removal of missing values is a critical step in the preprocessing phase of data analysis. By removing these values, we were able to ensure that our analysis was based on accurate and reliable data.

**3.4.2 SELECTING THE RELEVENT FEATURES**

In this module, the relevant features for the machine learning model are selected. The chosen features include 'AGE', 'GENDER', 'T3', 'T4U', and 'TSH'. The selection of these features is based on their potential predictive power for the output variable. By including only the most relevant features, we aim to increase the accuracy and efficiency of the machine learning model. These features will be used in the subsequent steps of the model development process.

**3.4.3 TRAINING THE MODEL**

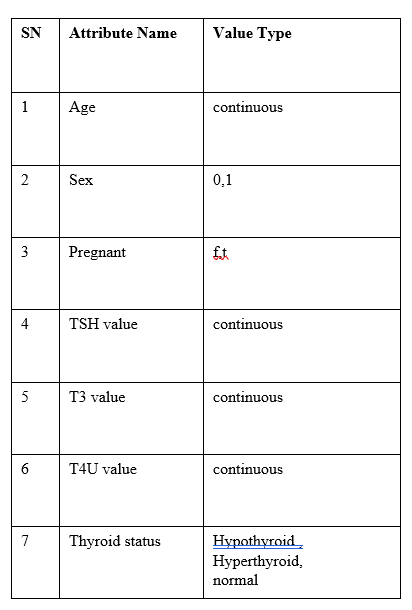
In the training module, the selected machine learning model is trained using the relevant features. The available machine learning models to choose from are Decision Tree, Logistic Regression, Random Forest, and Naive Bayes. The data is first split into training and testing sets, then the chosen model is fitted and its accuracy is evaluated. Once the model is trained, it is saved for future use.

**3.4.4 MAKING THE PREDICTION**

In the prediction module, the trained machine learning model is used to predict the output for a new data point. The new data point should contain the same features as the data points used to train the model. The input data is preprocessed to ensure it is in the correct format and any missing values are handled appropriately. The model then uses the selected features and its learned parameters to make a prediction. The predicted output can be a classification, such as a diagnosis of a disease. The prediction is then outputted to the user for interpretation and further action.

**3.5 DATASET DESCRIPTION**

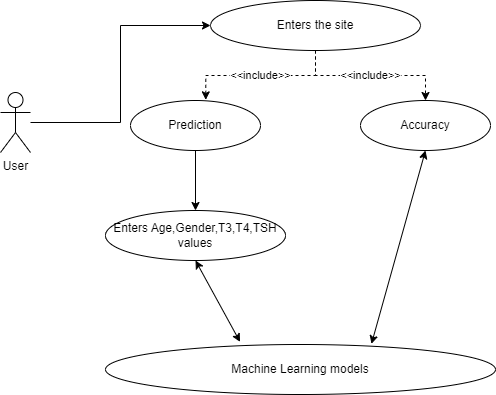
The dataset utilised in this work was obtained from the UCI Machine Learning Repository. The dataset comprises thyroid records of patients, and each record contains 5 attributes. The attributes can be either Binary (0/1) or continuous values. The table below (Table I) provides a list of these attributes.



* 1. **UML Diagram**

**3.6.1 USE CASE DIAGRAM**

A use case diagram is a graphic depiction of the interactions among the elements of a system. The use case is a methodology used in system analysis to identify, clarify and organize system requirements. Use case diagram describes the system functionality as a set of tasks that the system must carry out.



**Fig 3.3 Use case diagram**

### 3.6.2 ACTIVITY DIAGRAM

The activity diagram captures the dynamic behavior of the system. The other four diagrams are used to show the message flow from one object to another but activity diagram is used to show the message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part. It does not show any message flow from one activity to another. An activity diagram is sometimes considered the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

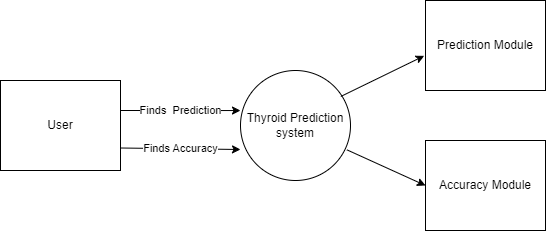
### 

### Fig 3.4 Activity diagram

### 3.6.3 DATA FLOW DIAGRAM

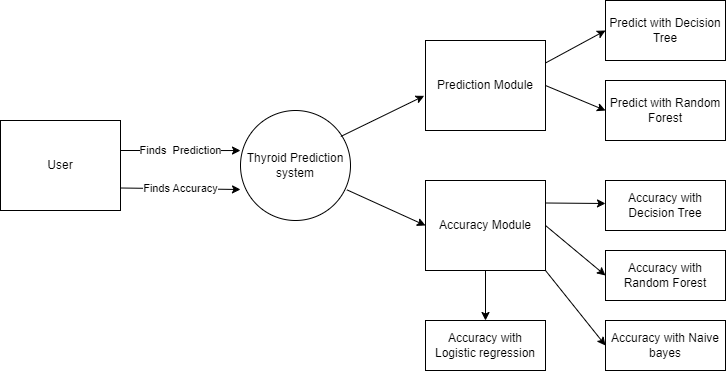
A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

**LEVEL 0**



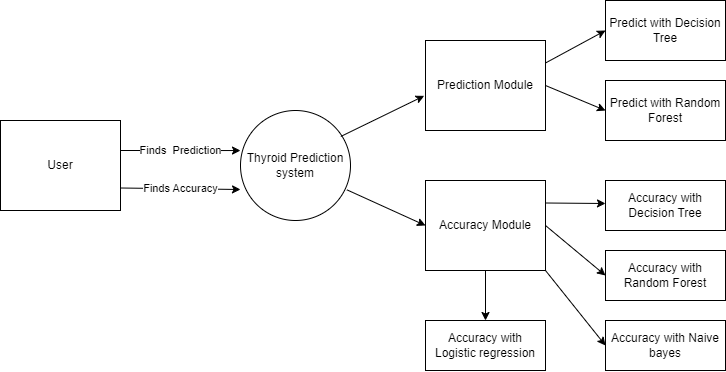
**Fig 3.5 Level 0 Data Flow diagram**

**LEVEL 1**



**Fig 3.6 Level 1 Data Flow diagram**

**LEVEL 2**



**Fig 3.7 Level 2 Data Flow diagram**

# CHAPTER-4

# REQUIREMENT SPECIFICATION

### 4. Requirements and specification

### 4.1 HARDWARE REQUIREMENT

* Processor : i3
* RAM : 4GB and above
* Storage : 50GB

### SOFTWARE REQUIREMENT

* Python 3.10
* SCIKIT Learn
* Python-Flask
* Pycharm IDE
* HTML
* CSS
* Javascript

**4.2.1 Python 3.10**

Python is a high-level, general-purpose programming language that has gained immense popularity in recent years. It was created in the late 1980s and has since evolved into a versatile language used for web development, data analysis, artificial intelligence, and scientific computing. Python is renowned for its readability and concise syntax, making it easy to learn and write code quickly. It is also open-source, meaning that it is free to use, distribute and modify. The vast and active Python community has developed a comprehensive set of libraries, frameworks, and tools, making it an ideal choice for various applications. Its popularity is reflected in the fact that it is consistently ranked among the top programming languages by industry surveys.

* + 1. **SCIKIT learn**

Python is a high-level programming language widely used in various scientific fields, including data analysis and machine learning. One of the most popular libraries for machine learning in Python is scikit-learn, which provides a wide range of tools and algorithms for data mining and analysis. Scikit-learn is an open-source library that features various classification, regression, and clustering algorithms, making it an ideal choice for developing predictive models. It also includes tools for model selection, data preprocessing, and evaluation, as well as visualization tools to help interpret the results. With its intuitive and user-friendly interface, scikit-learn has become a go-to choice for many researchers and practitioners in the field of machine learning.

* + 1. **Python-Flask**

Flask is a lightweight and flexible Python web framework that enables the creation of web applications in a simple and straightforward manner. Flask is considered a micro-framework because it does not require any specific libraries or tools and only requires minimal setup to get started. Flask provides a range of built-in tools and extensions to support various features such as routing, HTTP requests handling, and templating. Additionally, Flask supports a wide range of database systems and has the flexibility to integrate with various front-end frameworks, making it a versatile option for web development. Flask's simplicity and ease of use make it a popular choice for developing small to medium-sized web applications.

* + 1. **Pycharm IDE**

PyCharm is a popular Integrated Development Environment (IDE) used by developers to create Python applications efficiently. Developed by JetBrains, it offers a wide range of tools and features such as code completion, debugging, refactoring, version control integration, and more. PyCharm supports multiple frameworks, libraries, and tools, making it a versatile and customizable IDE. Its user-friendly interface and intuitive code editor make it easy for developers to write, test, and debug code. PyCharm also offers cross-platform support, allowing developers to work on multiple operating systems such as Windows, Linux, and macOS. With its powerful features and ease of use, PyCharm has become a go-to IDE for Python developers worldwide.

* + 1. **HTML**

HTML, short for Hypertext Markup Language, is the standard markup language for creating web pages and applications. It is a language that uses tags and attributes to define the structure, content, and layout of web documents. HTML documents consist of a series of elements, such as headings, paragraphs, images, links, and forms, that are used to create a visually appealing and interactive user interface. HTML files can be edited using a simple text editor, and modern web browsers can interpret and display HTML documents in a user-friendly way. With the addition of CSS and JavaScript, HTML can be used to create dynamic and responsive web pages that provide an immersive user experience. The versatility and simplicity of HTML make it an essential tool for web developers and designers.

* + 1. **CSS**

CSS stands for Cascading Style Sheets, and it is a fundamental technology used in web design. It is used to define the presentation and layout of HTML documents, including fonts, colors, spacing, and other visual elements. CSS provides designers with powerful tools for creating responsive and attractive web pages that can adapt to different screen sizes and devices. With CSS, designers can separate the content of a web page from its presentation, making it easier to maintain and update. CSS has evolved over the years, and modern CSS frameworks provide developers with pre-built styles and components, making it easier to create beautiful and responsive web pages without writing extensive custom code.

* + 1. **JAVASCRIPT**

JavaScript is a programming language that is widely used to create interactive and dynamic web pages. It was initially developed to add interactivity and animations to static HTML pages. JavaScript is a client-side language, which means it runs on the user's web browser instead of the web server. It is supported by all modern web browsers and is used in various web development frameworks and libraries. With JavaScript, developers can create dynamic and responsive web pages that can respond to user interactions and events in real-time. JavaScript is also commonly used for building web applications, games, and server-side applications using Node.js.

# CHAPTER-5

# IMPLEMENTATION

* 1. **SAMPLE CODE**

**Thyroid.html**

<!DOCTYPE html>

<head><title>Thyroid prediction system</title>

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='mobile.css') }}" media="only screen and (max-width: 400px)">

<link rel="stylesheet" type="text/css" href="{{ url\_for('static', filename='style.css') }}" media="only screen and (min-width: 401px)">

<style>

body{

background-image: url('https://mydemo03.s3.jp-tok.cloud-object-storage.appdomain.cloud/bg.jpeg');

background-repeat: no-repeat;

background-size: 100vw 100vh;}

</style>

</head>

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8iRABdZLl6O3oVMWSktQOp6b7In1Zl3/Jr59b6EGGoI1aFkw7cmDA6j6gD" crossorigin="anonymous">

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>

<script>

$(document).ready(function() {

// Listen for changes to the gender select box

$('#gender').change(function() {

// If the selected value is "male", update the pregnancy select box

if ($(this).val() === 'male') {

$('#pregnancy').css('font-weight', 'normal');

$('#pregnancy').val('not-pregnant');

$("#pregnancy").prop("disabled", true);

}

if ($(this).val() === 'female'){

$("#pregnancy").prop("disabled", false);

}

});

$('#age').on('keypress', function(event) {

var keyCode = event.which; // Get the ASCII code of the pressed key

// Allow only numbers (48 to 57)

if (keyCode < 48 || keyCode > 57){

event.preventDefault(); // Prevent default action (i.e. do not insert the character)

}

});

$('#t3').on('keypress', function(event) {

var keyCode = event.which; // Get the ASCII code of the pressed key

// Allow only numbers (48 to 57) or decimal point (46)

if ((keyCode < 48 || keyCode > 57) && keyCode !== 46) {

event.preventDefault(); // Prevent default action (i.e. do not insert the character)

}

});

$('#t4').on('keypress', function(event) {

var keyCode = event.which; // Get the ASCII code of the pressed key

// Allow only numbers (48 to 57) or decimal point (46)

if ((keyCode < 48 || keyCode > 57) && keyCode !== 46) {

event.preventDefault(); // Prevent default action (i.e. do not insert the character)

}

});

$('#tsh').on('keypress', function(event) {

var keyCode = event.which; // Get the ASCII code of the pressed key// Allow only numbers (48 to 57) or decimal point (46)

if ((keyCode < 48 || keyCode > 57) && keyCode !== 46) {

event.preventDefault(); // Prevent default action (i.e. do not insert the character)

}

});

$('#gender').change(function() {

// Remove font weight from select box element

$('#gender').css('font-weight', 'normal');

});

$('#pregnancy').change(function() {

// Remove font weight from select box element

$('#pregnancy').css('font-weight', 'normal');

});

});

</script>

<body>

<div id="links">

<span class="link" id="predict">Prediction</span>

<span class="link" id="compare">Accuracy</span>

</div>

<div id="body">

<div id="prediction">

<div id="predict\_content">

<div class="underline-input">

<input style="padding-left:10px; padding-right:10px;" type="text" id="name" placeholder="Name">

</div>

<div style="padding-left:5px; padding-right:5px;">

<select class="transparent-select" id="gender">

<option value="" disabled selected>Gender</option>

<option value="male">Male</option>

<option value="female">Female</option>

</select>

</div>

<div style="padding-left:5px; padding-right:5px;">

<select class="transparent-select" id="pregnancy">

<option value="" disabled selected>Pregnancy status</option>

<option value="pregnant">Pregnant</option>

<option value="not-pregnant">Not Pregnant</option>

</select>

</div>

<div class="underline-input">

<input style="padding-left:10px; padding-right:10px;" type="text" id="age" placeholder="Age">

</div>

<div class="underline-input">

<input style="padding-left:10px; padding-right:10px;" type="text" id="t3" placeholder="T3">

</div>

<div class="underline-input">

<input style="padding-left:10px; padding-right:10px;" type="text" id="t4" placeholder="T4">

</div>

<div class="underline-input">

<input style="padding-left:10px; padding-right:10px;" type="text" id="tsh" placeholder="TSH">

</div>

<h6 id= status\_header>Thyroid Status :<span id="status"></span></h6>

<center id="centered"><button id="predict\_button" type="button" class="btn btn-outline-dark">Predict </button></center>

<center><div class="spinner-border text-primary" role="status" style="margin-top:5px" id="spinner">

<span class="visually-hidden">Loading...</span>

</div></center>

</div>

</div>

<div id="accuracy">

<div id="accuracy\_content">

<span class="Card" id="Decision\_tree">

<h4 class="heading">DECISION TREE</h4>

<h5 class="content">Average Accuracy : <span id="Dec\_Avg\_Acc"></span></h5>

<h5 class="content">Maximum Accuracy : <span id="Dec\_Max\_Acc"></span></h5>

</span>

<span class="Card" id="Random\_forest">

<h4 class="heading">RANDOM FOREST</h4>

<h5 class="content">Average Accuracy : <span id="Ran\_Avg\_Acc"></span></h5>

<h5 class="content">Maximum Accuracy : <span id="Ran\_Max\_Acc"></span></h5>

</span>

<center><div class="spinner-border text-primary" role="status" style="margin-top:5px" id="spinner2">

<span class="visually-hidden">Loading...</span>

</div></center>

<span class="Card" id="Logistic\_regression">

<h4 class="heading">LOGISTIC REGRESSION</h4>

<h5 class="content">Average Accuracy : <span id="Log\_Avg\_Acc"></span></h5>

<h5 class="content">Maximum Accuracy : <span id="Log\_Max\_Acc"L></span></h5>

</span>

<span class="Card" id="Naive\_bayes">

<h4 class="heading">NAIVE BAYES</h4>

<h5 class="content">Average Accuracy : <span id="Naive\_Avg\_Acc"></span></h5>

<h5 class="content">Maximum Accuracy : <span id="Naive\_Max\_Acc"></span></h5>

</span>

</div>

</div>

</div>

</body>

<script type="text/javascript" src="{{url\_for('static',filename='thyroid.js')}}"></script>

</html>

**Mobile.css**

.link{

text-decoration: none;

font-size: 20px;

cursor: pointer;

}

#links{

}

#predict{

padding-left: 10px;

margin-left: 0px;

padding-right: 10px;

background-color:transparent;

text-decoration:underline;

color:white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#compare{

padding-left: 10px;

margin-bottom: 0px;

padding-right: 10px;

background-color:transparent;

color: white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#predict:hover{

background-color: rgba(0, 255, 255,0.1);

}

#compare:hover{

background-color: rgba(0, 255, 255,0.1);

}

/\*#body{

position:absolute;

width:100%;

height:100%;

background-color: rgba(0, 255, 255,0.1);

margin-left: 0px;

}\*/

#predict:active{

background-color: rgba(0, 255, 255,0.1);

}

#predict:visited{

background-color: rgba(0, 255, 255,0.1);

}

#predict\_content{

position:absolute;

margin-left: 30%;

margin-right: 30%;

margin-top: 50%;

margin-bottom: 5%;

background-color: transparent;

height: 75%;

width: 40%;

box-shadow: 0 0 20px #000;

border-radius:20px;

border-left: 5px solid rgba(0,100,255,0.7);

}

#prediction{

visibility: inherit;

}

#accuracy{

visibility: hidden;

}

#accuracy\_content{

position: absolute;

width: 100%;

height:100%;

}

.Card{

position: relative;

box-shadow: 0 0 20px #444;

width: 35%;

height: 30%;

display:inline-block;

background-color: rgba(0,174,227,0.3);

border: 0.011px solid black;

border-radius:20px;

color:black;

}

#Decision\_tree{

margin-top: 7%;

margin-left: 10%;

}

#Random\_forest{

margin-top: 7%;

margin-left: 10%;

}

#Logistic\_regression{

margin-top:3%;

margin-left: 10%;

}

#Naive\_bayes{

margin-top: 3%;

margin-left: 10%;

}

.heading{

padding-top: 20px;

text-align: center;

padding-bottom: 20px;

font-weight: bold;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

content{

text-align: center;

}

.underline-input {

position: relative;

margin-bottom: 1px;

padding-left: 5px;

padding-right: 5px;

padding-top: 5px;

padding-bottom: 0px;

}

.underline-input label {

position: absolute;

top: 0;

left: 0;

font-size: 1rem;

color: transparent;

pointer-events: none;

transition: all 0.2s ease;

background-color: transparent;

}

.underline-input input {

display: block;

width: 100%;

border: none;

border-bottom: 2px solid white;

padding: 0.5rem 0;

font-size: 1rem;

background-color: transparent;

color: black;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.underline-input input:focus {

outline: none;

border-bottom-color: rgba(0,100,255,0.7);

}

.underline-input input:focus + label,

.underline-input input:not(:placeholder-shown) + label {

top: -1.2rem;

font-size: 0.875rem;

color: rgba(0,100,255,0.7);

background-color: transparent;

}

.underline-input input::placeholder {

color: black;

font-weight: bold;

}

.transparent-select {

background-color: transparent;

color:black;

border: none;

width: 100%;

border-bottom: 2px solid white;

font-size: 1rem;

padding: 0.7rem 5px;

outline:none;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.transparent-select:focus {

outline: none;

border-bottom: 2px solid white;

}

.transparent-select option {

background-color: white;

color: black;

}

#spinner{

visibility:hidden;}

#spinner2{

visibility:inherit}

body{

background-image:url('bg.jpeg');

background-repeat: no-repeat;

background-size: 100vw 100vh;

}

#left{

width:400px;

height: 150px;

}

#gender{

font-weight: bold;

}

#pregnancy{

font-weight: bold;

}

#status{

padding-left: 5px;

}

**Style.css**

@media only screen and (max-width: 68.5em) {

/\* Styles for screens less than or equal to 768px wide \*/

.link{

text-decoration: none;

font-size: 20px;

cursor: pointer;

}

#links{

height: 80px;

}

#predict{

margin-left: 0px;

background-color:transparent;

text-decoration:underline;

color:white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

display: inline-block;

height: 70px;

font-size: 60px;

width: 50%;

text-align: center;

}

#compare{

margin-bottom: 0px;

background-color:transparent;

color: white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

height: 70px;

font-size: 60px;

width: 49.5%;

text-align: center;

display: inline-block;

}

#predict:hover{

background-color: rgba(0, 255, 255,0.1);

}

#compare:hover{

background-color: rgba(0, 255, 255,0.1);

}

/\*#body{

position:absolute;

width:100%;

height:100%;

background-color: rgba(0, 255, 255,0.1);

margin-left: 0px;

}\*/

#predict:active{

background-color: rgba(0, 255, 255,0.1);

}

#predict:visited{

background-color: rgba(0, 255, 255,0.1);

}

#predict\_content{

position:absolute;

margin-left: 10%;

margin-right: 10%;

margin-top: 22%;

margin-bottom: 5%;

background-color: transparent;

height: 75%;

width: 80%;

box-shadow: 0 0 20px #000;

border-radius:20px;

border-left: 5px solid rgba(0,100,255,0.7);

position: fixed;

}

#prediction{

visibility: inherit;

}

#accuracy{

visibility: hidden;

}

#accuracy\_content{

position: absolute;

width: 100%;

height:100%;

}

.Card{

position: relative;

box-shadow: 0 0 20px #444;

width: 35%;

height: 30%;

display:inline-block;

background-color: rgba(0,174,227,0.3);

border: 0.011px solid black;

border-radius:20px;

color:black;

display: inline-grid;

}

#Decision\_tree{

margin-top: 22%;

margin-left: 10%;

margin-bottom: 10%;

}

#Random\_forest{

margin-top: 7%;

margin-left: 10%;

}

#Logistic\_regression{

margin-top:10%;

margin-left: 10%;

}

#Naive\_bayes{

margin-top: 3%;

margin-left: 10%;

}

.heading{

padding-top: 20px;

text-align: center;

padding-bottom: 20px;

font-weight: bold;

font-size: 50px;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.content{

text-align: center;

font-size: 40px;

}

.underline-input {

position: relative;

margin-bottom: 1px;

padding-left: 5px;

padding-right: 5px;

padding-top: 5px;

padding-bottom: 0px;

}

.underline-input label {

position: absolute;

top: 0;

left: 0;

font-size: 1rem;

color: transparent;

pointer-events: none;

transition: all 0.2s ease;

background-color: transparent;

}

.underline-input input {

display: block;

width: 100%;

border: none;

border-bottom: 2px solid white;

padding: 0.5rem 0;

font-size: 1rem;

background-color: transparent;

color: black;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

height: 120px;

font-size: 50px;

}

.underline-input input:focus {

outline: none;

border-bottom-color: rgba(0,100,255,0.7);

}

.underline-input input:focus + label,

.underline-input input:not(:placeholder-shown) + label {

top: -1.2rem;

font-size: 0.875rem;

color: rgba(0,100,255,0.7);

background-color: transparent;

}

.underline-input input::placeholder {

color: black;

font-weight: bold;

}

.transparent-select {

background-color: transparent;

color:black;

border: none;

width: 100%;

border-bottom: 2px solid white;

font-size: 1rem;

padding: 0.7rem 5px;

outline:none;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

height: 120px;

font-size: 50px;

}

.transparent-select:focus {

outline: none;

border-bottom: 2px solid white;

}

.transparent-select option {

background-color: white;

color: black;

}

#spinner{

width: 80px;

height: 80px;

visibility: hidden;

}

#spinner2{

visibility:inherit;

width: 80px;

height: 80px;}

#left{

width:400px;

height: 150px;

}

#gender{

font-weight: bold;

}

#pregnancy{

font-weight: bold;

}

#status{

padding-left: 5px;

}

#status\_header{

padding-left:10px;

padding-top:60px;

font-size:50px;

padding-bottom:60px;

color:black;

font-weight:bold;

margin-bottom: 1px;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#centered{

padding-top: 60px;

padding-bottom: 60px;

}

#predict\_button{

font-weight:bold;

margin-bottom: 1px;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

height: 100px;

width: 300px;

font-size: 50px;

}

}

@media only screen and (min-width: 68.51em){

/\* Styles for screens between 769px and 1200px wide \*/

#predict\_button{

margin-top:20px;

font-weight:bold;

margin-bottom: 1px;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#status\_header{

padding-left:10px;

padding-top:15px;

font-size:15px;

padding-bottom:10px;

color:black;

font-weight:bold;

margin-bottom: 1px;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.link{

text-decoration: none;

font-size: 20px;

cursor: pointer;

}

#links{

}

#predict{

padding-left: 10px;

margin-left: 0px;

padding-right: 10px;

background-color:transparent;

text-decoration:underline;

color:white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#compare{

padding-left: 10px;

margin-bottom: 0px;

padding-right: 10px;

background-color:transparent;

color: white;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

#predict:hover{

background-color: rgba(0, 255, 255,0.1);

}

#compare:hover{

background-color: rgba(0, 255, 255,0.1);

}

/\*#body{

position:absolute;

width:100%;

height:100%;

background-color: rgba(0, 255, 255,0.1);

margin-left: 0px;

}\*/

#predict:active{

background-color: rgba(0, 255, 255,0.1);

}

#predict:visited{

background-color: rgba(0, 255, 255,0.1);

}

#predict\_content{

position:absolute;

margin-left: 30%;

margin-right: 30%;

margin-top: 7%;

margin-bottom: 5%;

background-color: transparent;

height: 75%;

width: 40%;

box-shadow: 0 0 20px #000;

border-radius:20px;

border-left: 5px solid rgba(0,100,255,0.7);

}

#prediction{

visibility: inherit;

}

#accuracy{

visibility: hidden;

}

#accuracy\_content{

position: absolute;

width: 100%;

height:100%;

}

.Card{

position: relative;

box-shadow: 0 0 20px #444;

width: 35%;

height: 30%;

display:inline-block;

background-color: rgba(0,174,227,0.3);

border: 0.011px solid black;

border-radius:20px;

color:black;

}

#Decision\_tree{

margin-top: 7%;

margin-left: 10%;

}

#Random\_forest{

margin-top: 7%;

margin-left: 10%;

}

#Logistic\_regression{

margin-top:3%;

margin-left: 10%;

}

#Naive\_bayes{

margin-top: 3%;

margin-left: 10%;

}

.heading{

padding-top: 20px;

text-align: center;

padding-bottom: 20px;

font-weight: bold;

font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.content{

text-align: center;

}

.underline-input {

position: relative;

margin-bottom: 1px;

padding-left: 5px;

padding-right: 5px;

padding-top: 5px;

padding-bottom: 0px;

}

.underline-input label {

position: absolute;

top: 0;

left: 0;

font-size: 1rem;

color: transparent;

pointer-events: none;

transition: all 0.2s ease;

background-color: transparent;

}

.underline-input input {

display: block;

width: 100%;

border: none;

border-bottom: 2px solid white;

padding: 0.5rem 0;

font-size: 1rem;

background-color: transparent;

color: black;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.underline-input input:focus {

outline: none;

border-bottom-color: rgba(0,100,255,0.7);

}

.underline-input input:focus + label,

.underline-input input:not(:placeholder-shown) + label {

top: -1.2rem;

font-size: 0.875rem;

color: rgba(0,100,255,0.7);

background-color: transparent;

}

.underline-input input::placeholder {

color: black;

font-weight: bold;

}

.transparent-select {

background-color: transparent;

color:black;

border: none;

width: 100%;

border-bottom: 2px solid white;

font-size: 1rem;

padding: 0.7rem 5px;

outline:none;

margin-bottom: 1px;font-family: Goudy Old Style,Garamond,Big Caslon,Times New Roman,serif;

}

.transparent-select:focus {

outline: none;

border-bottom: 2px solid white;

}

.transparent-select option {

background-color: white;

color: black;

}

#spinner{

visibility:hidden;}

#spinner2{

visibility:inherit}

#left{

width:400px;

height: 150px;

}

#gender{

font-weight: bold;

}

#pregnancy{

font-weight: bold;

}

#status{

padding-left: 5px;

}

}

**Thyroid.js**

const xhr = new XMLHttpRequest();

xhr.open('GET', 'http://127.0.0.1:5000/Accuracy');

xhr.onload = function() {

var dec\_avg=document.getElementById("Dec\_Avg\_Acc");

var dec\_max=document.getElementById("Dec\_Max\_Acc");

var ran\_avg=document.getElementById("Ran\_Avg\_Acc");

var ran\_max=document.getElementById("Ran\_Max\_Acc");

var log\_avg=document.getElementById("Log\_Avg\_Acc");

var log\_max=document.getElementById("Log\_Max\_Acc");

var naive\_avg=document.getElementById("Naive\_Avg\_Acc");

var naive\_max=document.getElementById("Naive\_Max\_Acc");

spinner2=document.getElementById("spinner2")

if (xhr.status === 200) {

const data = JSON.parse(xhr.responseText);

console.log(data);

console.log(data.Naive\_avg);

console.log(data.Naive\_max);

dec\_avg.innerHTML=data.Dec\_Avg;

dec\_max.innerHTML=data.Dec\_max;

ran\_avg.innerHTML=data.Ran\_Avg;

ran\_max.innerHTML=data.Ran\_max;

log\_avg.innerHTML=data.Log\_Avg;

log\_max.innerHTML=data.Log\_max;

naive\_avg.innerHTML=data.Naive\_Avg;

naive\_max.innerHTML=data.Naive\_max;

spinner2.style.visibility="hidden"

}

else {

console.error('Request failed. Returned status:', xhr.status);

}

};

xhr.onerror = function() {

console.error('Request failed. Unable to connect to server.');

};

xhr.send();

var predict=document.getElementById("predict");

var accuracy=document.getElementById("compare");

var prediction\_slide=document.getElementById("prediction");

var accuracy\_slide=document.getElementById("accuracy");

predict\_button.addEventListener("click",function(){

var Name=document.getElementById("name").value;

var Age=document.getElementById("age").value;

var Gender=document.getElementById("gender").value;

var Pregnancy=document.getElementById("pregnancy").value;

var T3=document.getElementById("t3").value;

var T4=document.getElementById("t4").value;

var TSH=document.getElementById("tsh").value;

predict\_button=document.getElementById("predict\_button")

var Status=document.getElementById("status");

var dec\_avg=document.getElementById("Dec\_Avg\_Acc");

var dec\_max=document.getElementById("Dec\_Max\_Acc");

var ran\_avg=document.getElementById("Ran\_Avg\_Acc");

var ran\_max=document.getElementById("Ran\_Max\_Acc");

var log\_avg=document.getElementById("Log\_Avg\_Acc");

var log\_max=document.getElementById("Log\_Max\_Acc");

var naive\_avg=document.getElementById("Naive\_Avg\_Acc");

var naive\_max=document.getElementById("Naive\_Max\_Acc");

var spinner=document.getElementById("spinner")

spinner.style.visibility="inherit";

console.log(Name,Age,Pregnancy,Gender,T3,T4,TSH,Status)

const data = { name: Name,age: Age, gender: Gender,pregnancy: Pregnancy,t3: T3,t4: T4,tsh: TSH };

const xhr = new XMLHttpRequest();

xhr.open('POST', 'http://127.0.0.1:5000/Predict');

xhr.setRequestHeader('Content-Type', 'application/json');

xhr.onload = function() {

if (xhr.status === 200) {

const response = JSON.parse(xhr.responseText);

console.log(response);

dec\_avg.innerHTML=response.Dec\_Avg;

dec\_max.innerHTML=response.Dec\_max;

ran\_avg.innerHTML=response.Ran\_Avg;

ran\_max.innerHTML=response.Ran\_max;

log\_avg.innerHTML=response.Log\_Avg;

log\_max.innerHTML=response.Log\_max;

naive\_avg.innerHTML=response.Naive\_Avg;

naive\_max.innerHTML=response.Naive\_max;

Status.innerHTML=response.status

spinner.style.visibility="hidden";

} else {

console.error('Request failed. Returned status:', xhr.status);

}

};

xhr.onerror = function() {

console.error('Request failed. Unable to connect to server.');

};

xhr.send(JSON.stringify(data));

});

predict.addEventListener("click",function(){

predict.style.textDecoration="underline"

accuracy.style.textDecoration="none"

prediction\_slide.style.visibility="inherit"

accuracy\_slide.style.visibility="hidden"

});

accuracy.addEventListener("click",function(){

predict.style.textDecoration="none"

accuracy.style.textDecoration="underline"

prediction\_slide.style.visibility="hidden"

accuracy\_slide.style.visibility="inherit"

});

**T****hyroid.py**

import pandas as pd

import numpy as np

import math

from decimal import Decimal

from sklearn.tree import DecisionTreeClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.ensemble import RandomForestClassifier

from sklearn.naive\_bayes import GaussianNB

from sklearn.metrics import accuracy\_score

from sklearn.model\_selection import train\_test\_split

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

import pickle

def preprocess\_data(df):

# Preprocess the data

df = df.dropna()

return df

def select\_features(df):

# Select the relevant features

features = ['AGE', 'GENDER', 'T3', 'T4U', 'TSH']

X = df[features]

y = df['DIAGNOSIS']

return X, y

def max\_times(l):

ans=""

for i in l:

if l.count(i)>2:

ans=i

break

return ans

def train\_model(X, y, model\_name):

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

# Choose the appropriate model

if model\_name == 'Decision Tree':

clf = DecisionTreeClassifier()

elif model\_name == 'Logistic Regression':

clf = LogisticRegression(max\_iter=6000)

elif model\_name == 'Random Forest':

clf = RandomForestClassifier()

elif model\_name == 'Naive Bayes':

clf = GaussianNB()

else:

raise ValueError(

"Invalid model name. Choose from 'Decision Tree', 'Logistic Regression', 'Random Forest','Naive Bayes'")

# Train the model

clf.fit(X\_train, y\_train)

# Evaluate the model

y\_pred = clf.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_pred)

return accuracy,clf

def make\_prediction(clf, X\_new):

# Make a prediction for a single person

y\_new = clf.predict(X\_new)

return y\_new

def find\_accuracy(Pregnancy\_status):

Decision\_tree = []

Logistic\_regression = []

Random\_forest = []

Naive\_bayes = []

if Pregnancy\_status:

dataset = 'Dataset\dataset0387\_pregnant.csv'

Decision\_tree\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Decision\_tree\_for\_pregnant.pkl"

Logistic\_regression\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Logistic\_regression\_for\_pregnant.pkl"

Random\_forest\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Random\_forest\_for\_pregnant.pkl"

Naive\_bayes\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Naive\_bayes\_for\_pregnant.pkl"

else:

dataset = 'Dataset\dataset0387\_normal.csv'

Decision\_tree\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Decision\_tree\_for\_normal.pkl"

Logistic\_regression\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Logistic\_regression\_for\_normal.pkl"

Random\_forest\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Random\_forest\_for\_normal.pkl"

Naive\_bayes\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Naive\_bayes\_for\_normal.pkl"

df = pd.read\_csv(dataset)

# Preprocess the data

df = preprocess\_data(df)

# Select the relevant features

X, y = select\_features(df)

accuracy,clf = train\_model(X, y, 'Decision Tree')

Decision\_tree.append(accuracy)

with open(Decision\_tree\_model, 'wb') as f:

pickle.dump(clf, f)

accuracy,clf = train\_model(X, y, 'Random Forest')

Random\_forest.append(accuracy)

with open(Random\_forest\_model, 'wb') as f:

pickle.dump(clf, f)

accuracy,clf = train\_model(X, y, 'Logistic Regression')

Logistic\_regression.append(accuracy)

with open(Logistic\_regression\_model, 'wb') as f:

pickle.dump(clf, f)

accuracy,clf = train\_model(X, y, 'Naive Bayes')

Naive\_bayes.append(accuracy)

with open(Naive\_bayes\_model, 'wb') as f:

pickle.dump(clf, f)

for i in range(0,4):

accuracy, clf = train\_model(X, y, 'Decision Tree')

Decision\_tree.append(accuracy)

accuracy, clf = train\_model(X, y, 'Random Forest')

Random\_forest.append(accuracy)

accuracy, clf = train\_model(X, y, 'Logistic Regression')

Logistic\_regression.append(accuracy)

accuracy, clf = train\_model(X, y, 'Naive Bayes')

Naive\_bayes.append(accuracy)

Dec\_Avg=round((sum(Decision\_tree)/len(Decision\_tree))\*100,2)

Dec\_max=round((max(Decision\_tree)\*100),2)

Ran\_Avg=round((sum(Random\_forest)/len(Random\_forest))\*100,2)

Ran\_max=round((max(Random\_forest) \* 100),2)

Log\_Avg=round((sum(Logistic\_regression)/len(Logistic\_regression))\*100,2)

Log\_max=round((max(Logistic\_regression) \* 100),2)

Naive\_Avg=round((sum(Naive\_bayes)/len(Naive\_bayes))\*100,2)

Naive\_max=round((max(Naive\_bayes) \* 100),2)

return Dec\_Avg,Dec\_max,Ran\_Avg,Ran\_max,Log\_Avg,Log\_max,Naive\_Avg,Naive\_max

def predict(model\_name,Pregnancy\_status,X\_new):

if Pregnancy\_status:

dataset = 'Dataset\dataset0387\_pregnant.csv'

Decision\_tree\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Decision\_tree\_for\_pregnant.pkl"

Logistic\_regression\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Logistic\_regression\_for\_pregnant.pkl"

Random\_forest\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Random\_forest\_for\_pregnant.pkl"

Naive\_bayes\_model = "Trained\_models\_for\_pregnant/trained\_model\_by\_Naive\_bayes\_for\_pregnant.pkl"

else:

dataset = 'Dataset\dataset0387\_normal.csv'

Decision\_tree\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Decision\_tree\_for\_normal.pkl"

Logistic\_regression\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Logistic\_regression\_for\_normal.pkl"

Random\_forest\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Random\_forest\_for\_normal.pkl"

Naive\_bayes\_model = "Trained\_models\_for\_normal/trained\_model\_by\_Naive\_bayes\_for\_normal.pkl"

df = pd.read\_csv(dataset)

df = preprocess\_data(df)

X, y = select\_features(df)

if model\_name == 'Decision Tree':

with open(Decision\_tree\_model, 'rb') as f:

clf = pickle.load(f)

X\_new = pd.DataFrame(X\_new, columns=X.columns)

pred = make\_prediction(clf, X\_new)

elif model\_name == 'Logistic Regression':

with open(Logistic\_regression\_model, 'rb') as f:

clf = pickle.load(f)

X\_new = pd.DataFrame(X\_new, columns=X.columns)

pred = make\_prediction(clf, X\_new)

elif model\_name == 'Random Forest':

with open(Random\_forest\_model, 'rb') as f:

clf = pickle.load(f)

X\_new = pd.DataFrame(X\_new, columns=X.columns)

pred = make\_prediction(clf, X\_new)

elif model\_name == 'Naive Bayes':

with open(Naive\_bayes\_model, 'rb') as f:

clf = pickle.load(f)

X\_new = pd.DataFrame(X\_new, columns=X.columns)

pred = make\_prediction(clf, X\_new)

else:

raise ValueError(

"Invalid model name. Choose from 'Decision Tree', 'Logistic Regression', 'Random Forest','Naive Bayes'")

return pred

def final\_prediction(Pregnancy\_status,X\_new):

i=0

final=[]

if Pregnancy\_status:

while i<5:

Ran=predict("Random Forest",1,X\_new)

a=find\_accuracy(1)

print(Ran)

if (Ran):

final.append(Ran)

i=i+1

else:

while i<5:

Ran = predict("Random Forest", 0, X\_new)

Dec = predict("Decision Tree",0,X\_new)

print(Ran,Dec)

if (Ran==Dec):

final.append(Ran)

i=i+1

ans=max\_times(final)

return ans

**main.py**

from Thyroid import \*

from flask import Flask,render\_template,request,jsonify

from flask\_cors import CORS

'''print(find\_accuracy(1))

print(predict("Random Forest",0,X\_new = [[75, 0, 1.6, 0.89, 0.05]]))

print(final\_prediction(1,X\_new = [[23, 0, 2.8, 1.14, 0.1]]))'''

app = Flask(\_\_name\_\_)

"CORS(app)"

@app.route('/')

def index():

return render\_template("Thyroid.html")

@app.route('/Accuracy',methods=['POST','GET'])

def accuracy():

if request.method=="GET":

Dec\_Avg,Dec\_max,Ran\_Avg,Ran\_max,Log\_Avg,Log\_max,Naive\_Avg,Naive\_max=find\_accuracy(0)

print(Dec\_Avg,Dec\_max,Ran\_Avg,Ran\_max,Log\_Avg,Log\_max,Naive\_Avg,Naive\_max)

Accuracy={"Dec\_Avg":Dec\_Avg,"Dec\_max":Dec\_max,"Ran\_Avg":Ran\_Avg,"Ran\_max":Ran\_max,

"Log\_Avg":Log\_Avg,"Log\_max":Log\_max,"Naive\_Avg":Naive\_Avg,"Naive\_max":Naive\_max}

return jsonify(Accuracy)

@app.route('/Predict', methods=['POST'])

def post\_example():

data = request.get\_json()

name = data['name']

age = int(data['age'])

gender = data['gender']

pregnancy=data['pregnancy']

t3 = float(data['t3'])

t4 = float(data['t4'])

tsh = float(data['tsh'])

print(data)

if pregnancy=="pregnant":

pregnancy=1

else:

pregnancy=0

if gender=="male":

gender=1

else:

gender=0

status=final\_prediction(pregnancy,[[age,gender,t3,t4,tsh]])

print(status)

Dec\_Avg, Dec\_max, Ran\_Avg, Ran\_max, Log\_Avg, Log\_max, Naive\_Avg, Naive\_max = find\_accuracy(pregnancy)

print(Dec\_Avg, Dec\_max, Ran\_Avg, Ran\_max, Log\_Avg, Log\_max, Naive\_Avg, Naive\_max)

Dictionary = {"Dec\_Avg": Dec\_Avg, "Dec\_max": Dec\_max, "Ran\_Avg": Ran\_Avg, "Ran\_max": Ran\_max,

"Log\_Avg": Log\_Avg, "Log\_max": Log\_max, "Naive\_Avg": Naive\_Avg, "Naive\_max": Naive\_max,

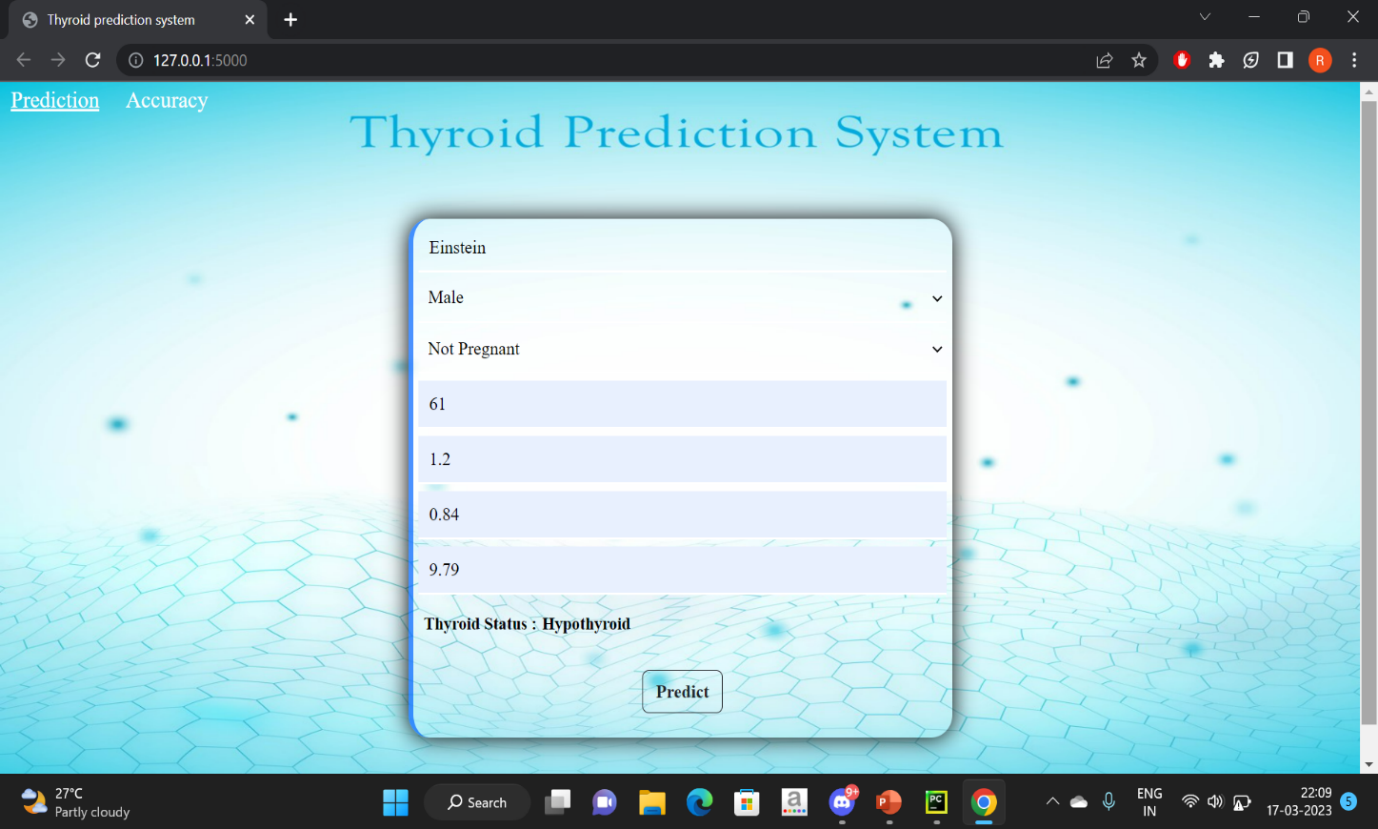
'status':status[0]}

return jsonify(Dictionary)

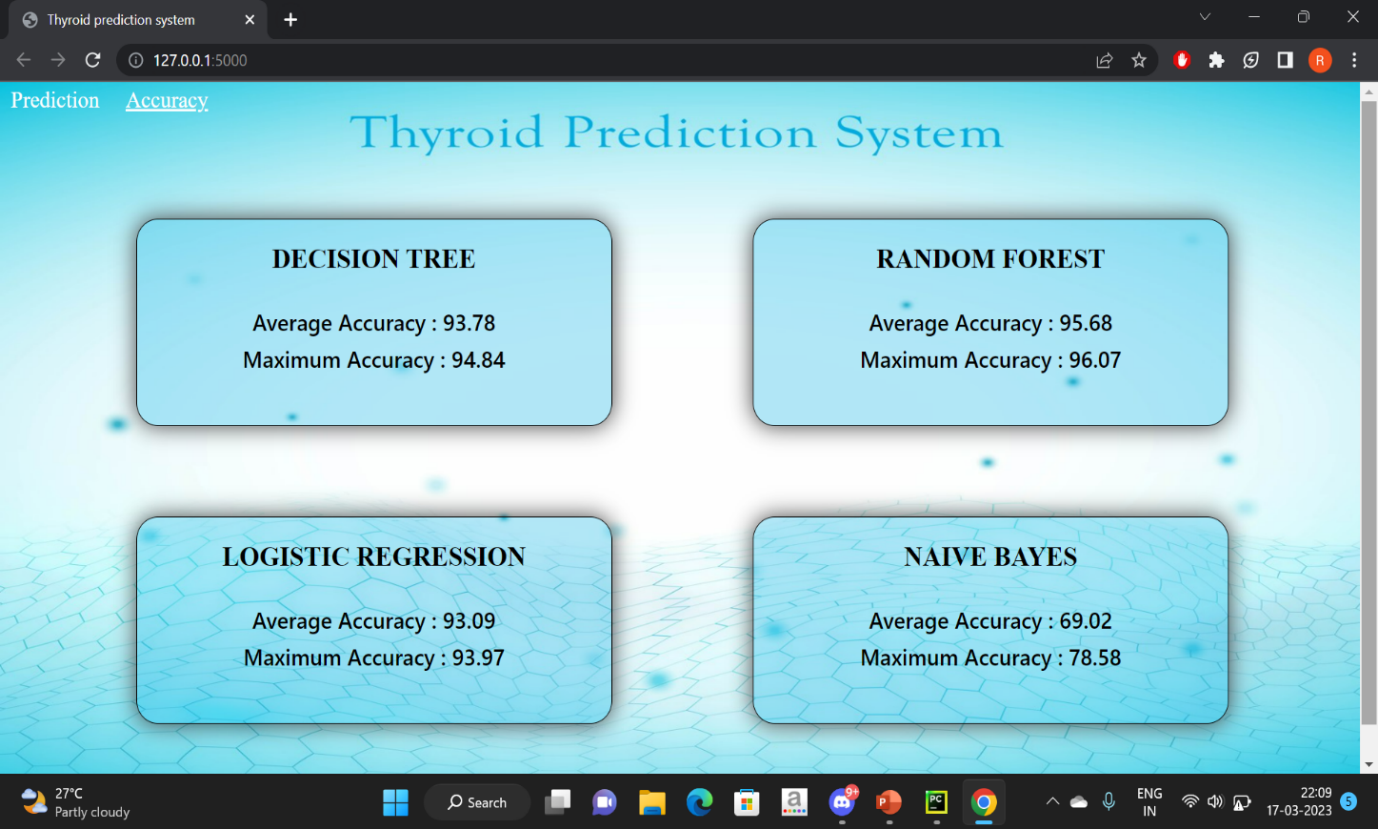
if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

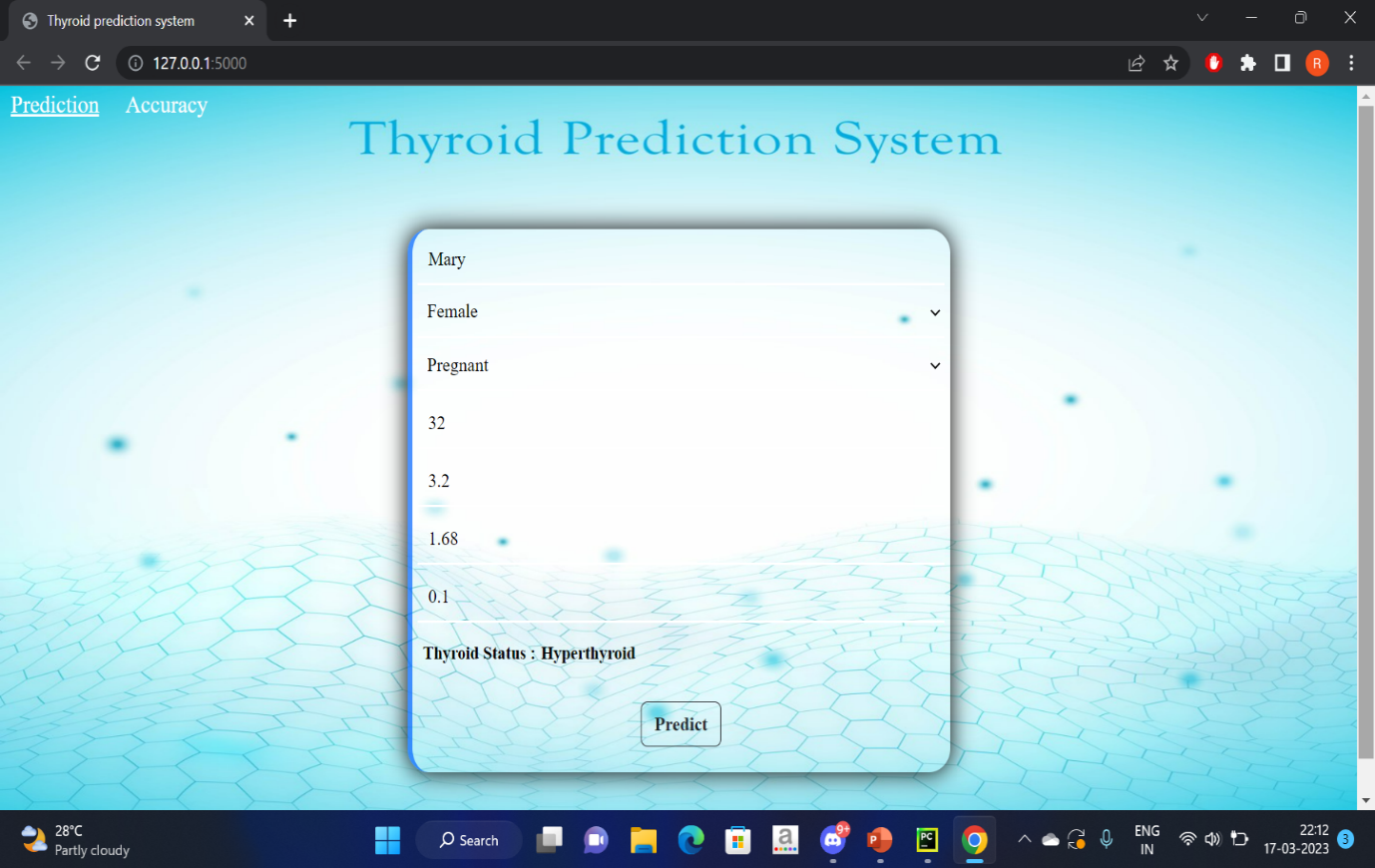
* 1. **SAMPLE SCREEN SHOTS**

****

**Fig 5.1 Predicting Hypothyroid condition**

****

**Fig 5.2 Calculating Accuracy**

****

**Fig 5.3 Predicting Hyperthyroid condition**

# 

# Fig 5.4 Predicting Negative condition

# CHAPTER-6

**TESTING AND MAINTENANCE**

## 6.1 BLACK BOX TESTING

## Black box testing is a software testing technique that examines the functionality of a software system without considering its internal structure or implementation. This testing method focuses on the system's inputs and outputs, simulating real-world user scenarios, and verifying whether the system functions as expected. Testers evaluate the software system solely based on its specifications, requirements, and functionality, without any knowledge of its codebase, design, or architecture. Black box testing is an essential part of software testing, as it ensures that the software system is functioning correctly from the user's perspective and meets the required specifications and expectations. This testing method is widely used in industries, such as finance, healthcare, and e-commerce, to detect defects, improve system reliability, and ensure customer satisfaction.

## 6.2 WHITE BOX TESTING

## White box testing, also known as clear box testing, is a software testing technique that examines the internal workings of a system, including its code and infrastructure. In this approach, the tester has knowledge of the internal logic and structure of the system and can write test cases that verify the expected behavior of individual components or functions. White box testing is often used in conjunction with black box testing to provide a more thorough testing process. It helps to identify hidden errors, security flaws, and defects that may be challenging to detect using only black box testing. By uncovering these issues early in the development cycle, white box testing helps to reduce the overall cost and time required to create a high-quality software product.

## 6.3 UNIT TESTING

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing. It can be characterized in several ways such as, a unit in a typical procedure-oriented software system. It performs a single cohesive function and can be compiled separately. It is a task in a work breakdown structure from the manager’s point of view. A unit testing is traditionally viewed as a function or procedure implemented in a procedural (imperative) programming language.

## 6.4 INTEGRATION TESTING

It performed after the unit testing and before system testing where individual units are combined and tested as a group. The purpose of this testing is to expose faults in the interaction between integrated units. Once all the modules have been unit tested, integrated testing is performed. Developers themselves or independent Testers perform Integration testing. It is also used for the construction of software architecture.

In Integration Testing there are two approaches

* Non-Incremental integration testing
* Incremental integration testing

## 6.5 SYSTEM TESTING

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The Goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example, the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walk through. Testing is one of the important steps in the software development phase.

## 6.6 ACCEPTANCE TESTING

## Acceptance testing is a crucial phase in the software development life cycle, where the software is tested to ensure that it meets the requirements and specifications of the customer. It is performed by the customer or end-user to verify that the software is ready for deployment. Acceptance testing is a type of black-box testing that determines whether the software is functioning correctly and is ready to be released to the end-users. It involves testing the software under realistic conditions, using real data, and testing all possible use cases to ensure that the software meets the customer's expectations. The goal of acceptance testing is to validate the software's functionality, reliability, performance, and usability to ensure that it meets the user's needs and requirements.

## 6.7 TESTCASE SPECIFICATION

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **TEST CASE ID** | **MODULE** | **INPUT** | **EXPECTED**  **OUTPUT** | **ACTUAL**  **OUTPUT** | **STATUS** |
| TC1 | Collecting the data | Data set | Processed thyroid patient dataset | Processed thyroid patient dataset | PASS |
| TC2 | Feature Selection | Attributes | Features for training and testing the data | Features for training and testing the data | PASS |
| TC3 | Model Training | Features and data | Trained model | Trained model | PASS |
| TC4 | Predicting the condition | Parameters required to predict the condition | Condition of the patient | Condition of the patient | PASS |

# CHAPTER-7

# CONCLUSION AND FUTURE ENHANCEMENTS

### CONCLUSION

### The study demonstrates the effectiveness of data mining algorithms, particularly Random Forest, in classifying thyroid disease. Early detection and diagnosis of thyroid disease are critical for effective treatment, and data mining techniques can significantly aid healthcare professionals in this area. Naïve Bayes, Logistic regression and Decision tree algorithms can also provide accurate results and improve patient management by reducing noisy data. These findings are useful for researchers and medical professionals to identify cost-effective treatment strategies and provide better patient care.

### FUTURE ENCHANCEMENT

### Future studies should focus on identifying the causes of thyroid disease and developing new data mining techniques to classify various ailments. Additionally, research can be conducted to compare the accuracy and performance of different data mining algorithms and identify the most effective ones for various types of data. The integration of machine learning with data mining can also help in developing predictive models that can aid in early detection and diagnosis of thyroid disease. Furthermore, the use of big data and cloud-based technologies can be explored to facilitate data mining and improve the efficiency of the healthcare system.

### REFERENCES

[1] Begum and A. Parkavi, "Prediction of thyroid Disease Using Data Mining Techniques," 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), Coimbatore, India, 2019, pp. 342-345, doi: 10.1109/ICACCS.2019.8728320*.*

[2] Irina IoniŃă and Liviu IoniŃă” Prediction of Thyroid Disease Using Data Mining Techniques”The Classification Technique for Talent Management using SVM, (ICCEET), 978-1- 4673-02l0-4/12, pp. 959- 963, 2017.

[3] Hanung Adi Nugroho, Noor Akhmad Setiawan, Md. Dendi Maysanjaya,” A Comparison of Classification Methods on Diagnosis of Thyroid Diseases”IEEE International Seminar on Intelligent Technology and Its Applications,2017.

[4] Al-muwaffaq, Izdihar, and Zeki Bozkus. "MLTDD: use of machine learning techniques for diagnosis of thyroid gland disorder." *Comput Sci Inf Technol* (2016): 67-3.

[5] Prerana, Parveen Sehgal, and Khushboo Taneja. "Predictive data mining for diagnosis of thyroid disease using neural network." *International Journal of Research in Management, Science & Technology* 3.2 (2015): 75-80.

[6] Singh, Nikita, and Alka Jindal. "A segmentation method and comparison of classification methods for thyroid ultrasound images." *International Journal of Computer Applications* 50.11 (2012): 43-49.

[7] Banu, G. Rasitha. "Predicting thyroid disease using linear discriminant analysis (LDA) data mining technique." *Commun. Appl. Electron.(CAE)* 4 (2016): 4-6.

[8] Mishra, Sushruta, et al. "Thyroid disorder analysis using random forest classifier." *Intelligent and Cloud Computing: Proceedings of ICICC 2019, Volume 2*. Springer Singapore, 2021.

[9] <https://archive.ics.uci.edu/ml/machine-learning-databases/thyroid-disease/>