A Mini Project Report on

Thyroid Disease Prediction Using Machine Learning

Submitted in partial fulfillment for the award of the degree of Bachelor of Technology in

Electronics and Communication Engineering

by

K Anupama Reddy	**22321A0411**
P Anya Reddy	**22321A0413**
I Ashwitha	**22321A0416**

Under the esteemed Guidance of
Internal Guide
Ms. B Jyothsna

Associate Professor, ECE Department

Bhoj Reddy Engineering College for Women
Department of Electronics and Communication Engineering
(Sponsored by Sangam Laxmibai Vidyapeet,Accredited by NAAC with A Grade,approved by AICTE & affiliated to JNTUH) Recognised by UGC under section 2(f) of the UGC Act,1956
Vinaynagar, Santoshnagar X roads, Saidabad, Hyderabad – 500 059
Ph: +91-40-2459 2400 Fax: +91-40-2453 7281, www.brecw.ac.in, principal@brecw.ac.in

2024-25

Bhoj Reddy Engineering College for Women
(Sponsored by Sangam Laxmibai Vidyapeet, Accredited by NAAC with A Grade, Approved by AICTE and Affiliated to JNTUH)
Recognized by UGC under section 2(f) of the UGC Act, 1956
Vinaynagar, IS Sadan Crossroads, Saidabad, Hyderabad – 500 059, Telangana. www.brecw.ac.in

CERTIFICATE

This is to certify that the Mini Project titled "**Thyroid Disease Prediction Using Machine Learning"** is a bonafide work carried over by Ms. K Anupama Reddy(22321A0411), Ms. P Anya Reddy(22321A0413) and Ms. I Ashwitha (22321A0416) in partial fulfillment of the requirements for the award of the degree Bachelor of Technology in Electronics and Communication Engineering from Bhoj Reddy Engineering College for Women, Hyderabad, affiliated to Jawaharlal Nehru Technological University Hyderabad (JNTUH) during the Third Year second semester of their B. Tech course (academic year 2024-2025).

B Jyothsna
Internal Guide

S Manjula
HOD-ECE

External Examiner

ACKNOWLEGMENT

The satisfaction that accompanies the successful completion of the task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We would like to express our sincere gratitude to **Ms.B Jyothsna**, Associate Professor and Project guide for the eminent guidance and supervision at every stage.
We are thankful to **Ms. S Manjula,** Head of the Department, for her valuable guidance and encouragement during our Project.

We also thank **Dr. J Madhavan**, Principal, BRECW for providing the wonderful education environment in our college.

We are equally thankful to **Dr.N Pradeep Kumar Goud,** Project Coordinator for his continuous support and all the staff of Electronics and Communication Engineering Department of BRECW for their timely help and suggestions in the Project.

K Anupama Reddy (22321A0411)	anupama.kethireddy3@gmailcom
P Anya Reddy (22321A0413)	anyareddyparigi@gmail.com
I Ashwitha (22321A0416)	ashwithaindla14@gmail.com

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Thyroid disease is a common endocrine disorder affecting millions of people worldwide. The thyroid gland, a small butterfly-shaped organ located in the neck, plays a crucial role in regulating various metabolic processes in the body by secreting hormones such as thyroxine (T4) and triiodothyronine (T3). When the thyroid produces too much hormone (hyperthyroidism) or too little (hypothyroidism), it leads to metabolic imbalances that can significantly affect overall health. Early and accurate diagnosis of thyroid disorders is essential for effective treatment and management.

Traditional diagnostic methods rely heavily on clinical evaluations and lab-based test results such as TSH, T3, and T4 levels. However, interpreting these tests can sometimes be complex and prone to human error. Moreover, symptoms of thyroid disease often overlap with other conditions, making diagnosis challenging. In this context, Machine Learning (ML) offers a promising solution. By analyzing patterns in large datasets of patient information, ML algorithms can assist in the early detection and classification of thyroid conditions with high accuracy.

This project explores the development of a predictive model for thyroid disease using Machine Learning techniques. The model aims to classify whether a patient has a thyroid disorder based on clinical parameters, enabling faster diagnosis and treatment.

CHAPTER 1
Introduction

1.1 Introduction

Thyroid disorders are among the most common endocrine system diseases worldwide, affecting individuals across all age groups. The thyroid gland, a small butterfly-shaped organ located in the neck, is responsible for producing hormones that regulate crucial bodily functions such as metabolism, heart rate, temperature, and energy levels. When this gland becomes overactive (hyperthyroidism) or underactive (hypothyroidism), it can lead to significant health complications. Early diagnosis and treatment are essential to prevent long- term effects and to manage symptoms effectively.

Traditionally, the diagnosis of thyroid diseases involves a detailed analysis of a patient's medical history, physical examinations, and various laboratory tests such as T3, T4, and TSH (Thyroid Stimulating Hormone) levels. While these diagnostic methods are clinically accurate, they are often time-consuming, resource-intensive, and may require specialized equipment and interpretation by medical professionals. This poses challenges, especially in under-resourced or rural healthcare settings.

In recent years, the rise of machine learning (ML) and data science in healthcare has opened new possibilities for disease detection and predictive analytics. Machine learning algorithms can analyze large sets of medical data to uncover hidden patterns and make accurate predictions. By training models on labeled medical datasets, it is possible to automate the classification of thyroid conditions based on input attributes such as hormone levels and patient demographics.

This project focuses on building an intelligent thyroid disease prediction system using machine learning techniques. Publicly available datasets, such as those from Kaggle, provide a diverse range of thyroid patient records that include relevant diagnostic features. Various supervised learning algorithms like Random Forest Classifier (RFC), CatBoost, and XGBoost are employed to build and compare models based on their performance.

The ultimate objective is to develop a reliable, efficient, and interpretable system that aids medical professionals in the early detection of thyroid disorders. Such a tool not only enhances diagnostic accuracy but also reduces the dependency on complex tests and specialist intervention, making healthcare more accessible and proactive.

1.2 Aim

The aim of the project "Thyroid Disease Prediction Using Machine Learning" is to develop an intelligent system that can accurately predict thyroid disorders based on patient medical data using advanced machine learning algorithms. The system is trained on a dataset containing clinical features such as T3, T4, and TSH levels, which are key indicators of thyroid gland function. By learning from historical records, the model identifies patterns and relationships within the data to classify whether a person has hypothyroidism, hyperthyroidism, or a normal condition. The objective is to create a tool that can assist healthcare professionals in making faster and more reliable diagnostic decisions, especially in scenarios where access to laboratory testing is limited or costly. The system is designed to process input data, generate predictions, and present results in a clear and user-friendly format. Overall, the project aims to enhance diagnostic accuracy, enable early detection, and contribute to efficient thyroid disease management through machine learning.

1.3 Motivation

The motivation behind developing "Thyroid Disease Prediction using Machine Learning" is to improve the early detection and diagnosis of thyroid disorders through advanced computational methods. Thyroid diseases often go undiagnosed due to subtle or overlapping symptoms, and delayed treatment can lead to serious health complications. By leveraging machine learning algorithms to analyze clinical data, this project aims to provide a fast, accurate, and cost-effective tool for thyroid disease screening. It empowers healthcare professionals with decision-support tools and helps reduce the burden on medical infrastructure. Additionally, it promotes preventive healthcare by enabling timely intervention, ultimately enhancing patient outcomes and contributing to a healthier society.

1.4 Objectives

Certainly! Here are the expanded objectives for the "Thyroid disease prediction using ML" project:

The main objectives of this project are:

1. To collect and preprocess thyroid-related clinical data for accurate and consistent model training.
2. To explore and implement suitable machine learning algorithms for predicting different types of thyroid diseases.
3. To evaluate and compare the performance of classifiers such as Logistic Regression, Decision Tree, Random Forest, and others.
4. To develop a model that can accurately classify patients into categories such as hyperthyroidism, hypothyroidism, or normal.
5. To visualize the results using performance metrics like accuracy, precision, recall, F1-score, confusion matrix, and ROC-AUC curve.
6. To create a reliable and scalable system that can assist healthcare professionals in the early diagnosis of thyroid disorders.

1.5 Organization

Chapter 1: Introduction – Provides a background on thyroid diseases, the importance of early diagnosis, and how machine learning can assist in medical predictions. It also outlines the aim, objectives, and structure of the project.
Chapter 2: Software Requirements – Describes the software and hardware tools required to develop the system, including programming environments, libraries, and system configurations.
Chapter 3: Block Diagram and Its Explanation – Presents the system architecture through a block diagram and explains the functional flow of data from input to prediction.
Chapter 4: Methodology – Details the steps involved in building the machine learning model, including data collection, preprocessing, algorithm selection, model training, and validation.
Chapter 5: Results and Discussion – Showcases the output of the implemented model, including accuracy and performance metrics, and provides an analysis of the results.
Chapter 6: Advantages, Disadvantages, and Applications – Discusses the benefits and limitations of the proposed system and explores its potential real-world applications.
Chapter 7: Conclusion and Future Scope – Summarizes the project outcomes and suggests possible future enhancements to improve the system further.

1.6 Literature Survey

Various studies have explored the application of machine learning algorithms for thyroid disease prediction, highlighting different methods and their effectiveness. In 2016, Khushboo Chandel [01] applied K-Nearest Neighbors (KNN) and Naive Bayes algorithms, achieving 93.44% accuracy with KNN and 22.56% with Naive Bayes, indicating the former's suitability for thyroid classification tasks. That same year, G. Rasitha Banu [02] implemented the J48 decision tree algorithm and reported an impressive accuracy of 99.85%, demonstrating the potential of rule-based models in clinical prediction systems.

In 2019, Umar Sidiq, Dr. Syed Mutahar Aaqib, and Rafi Ahmad Khan [03] conducted a comparative analysis using KNN, SVM, Decision Tree, and Naive Bayes algorithms. Their study revealed that both KNN and Naive Bayes achieved 98.89% accuracy, while SVM reached 96.30%, showing the strength of simple algorithms when trained with clean, preprocessed datasets.

In 2020, Vijiya Kumar K. et al. [04] developed a thyroid disease prediction system using the Random Forest algorithm. Their findings confirmed that Random Forest could successfully classify thyroid conditions with high precision, owing to its ensemble learning mechanism and ability to handle noisy data.

More recently, in 2022, Lee and Park [05] utilized multiple algorithms, including Random Forest, Artificial Neural Networks (ANN), and XGBoost. Their model achieved an accuracy range of 64.3% to 99.5%, with the ANN model yielding an F1-score of 0.957 on the UCI thyroid dataset, indicating excellent performance in handling multiclass classification.

These studies collectively prove that machine learning provides reliable, accurate tools for predicting thyroid disorders, and inspire the current project to build upon these findings using advanced classification techniques.

1.8 Conclusion

The project "Thyroid Disease Prediction Using Machine Learning" demonstrates an effective approach for early detection of thyroid disorders using patient health data. By applying algorithms like Random Forest, CatBoost, and XGBoost on medical features such as T3, T4, and TSH, the system achieved high prediction accuracy. This model can support doctors in making quick and accurate diagnoses, especially in areas with limited access to lab facilities. The project shows that machine learning can significantly enhance medical diagnostics by reducing time, cost, and human error, making healthcare more efficient, reliable, and accessible.

CHAPTER 2

Software Requirements

2.1 Introduction

Software requirements are a critical aspect of any machine learning project, ensuring a smooth and efficient development process. In this project, which focuses on thyroid disease prediction using machine learning, the software environment is selected to support all stages—data preprocessing, model training, evaluation, and visualization.

For development and execution, Google Colab is used as the primary environment. Google Colab is a cloud-based platform that supports Python and allows users to write and run code in Jupyter notebook format. It offers access to free GPU and TPU resources, which enhances the speed and efficiency of model training, especially when dealing with large datasets or complex algorithms. Its online nature eliminates the need for software installation and allows for easy sharing and collaboration.

The project is developed in Python, a widely-used language in the data science and machine learning community. Python is chosen due to its simplicity, readability, and the availability of powerful libraries such as Pandas, NumPy, Matplotlib, Seaborn, and Scikit-learn. These libraries support all necessary tasks including data handling, statistical analysis, machine learning algorithm implementation, and result visualization.

- Pandas is used for data manipulation and preprocessing, such as handling missing values, transforming data, and creating input features.
- NumPy provides support for numerical operations and efficient array handling.
- Scikit-learn is the core library used to implement machine learning algorithms like K-Nearest Neighbors, Decision Trees, Naive Bayes, and Support Vector Machines. It also includes tools for splitting datasets, training models, and evaluating performance metrics such as accuracy and F1-score.
- Matplotlib and Seaborn are used to visualize datasets and performance results through graphs like confusion matrices, bar charts, and ROC curves.

2.2 Software Tools Used

The development and implementation of the thyroid disease prediction system involved the use of several key software tools. These tools enabled data preprocessing, model building, evaluation, and result visualization:

1. **Python**

Python is the primary programming language used for this project due to its readability, flexibility, and strong support for data science and machine learning through a wide range of libraries.

2. GoogleColab

Google Colaboratory (Colab) is used as the integrated development environment (IDE). It offers a free cloud-based platform with support for Python, access to GPUs, and easy collaboration. It also eliminates the need for local software installation.

3. Pandas

Pandas is used for data manipulation and analysis.It allows for easy handling of large datasets and provides data structures like DataFrames for structured data processing.

4. Numpy

NumPy supports high-performance mathematical operations on arrays and matrices,which are essential for efficient computation during model training and evaluation.

5Matplotlib&Seaborn

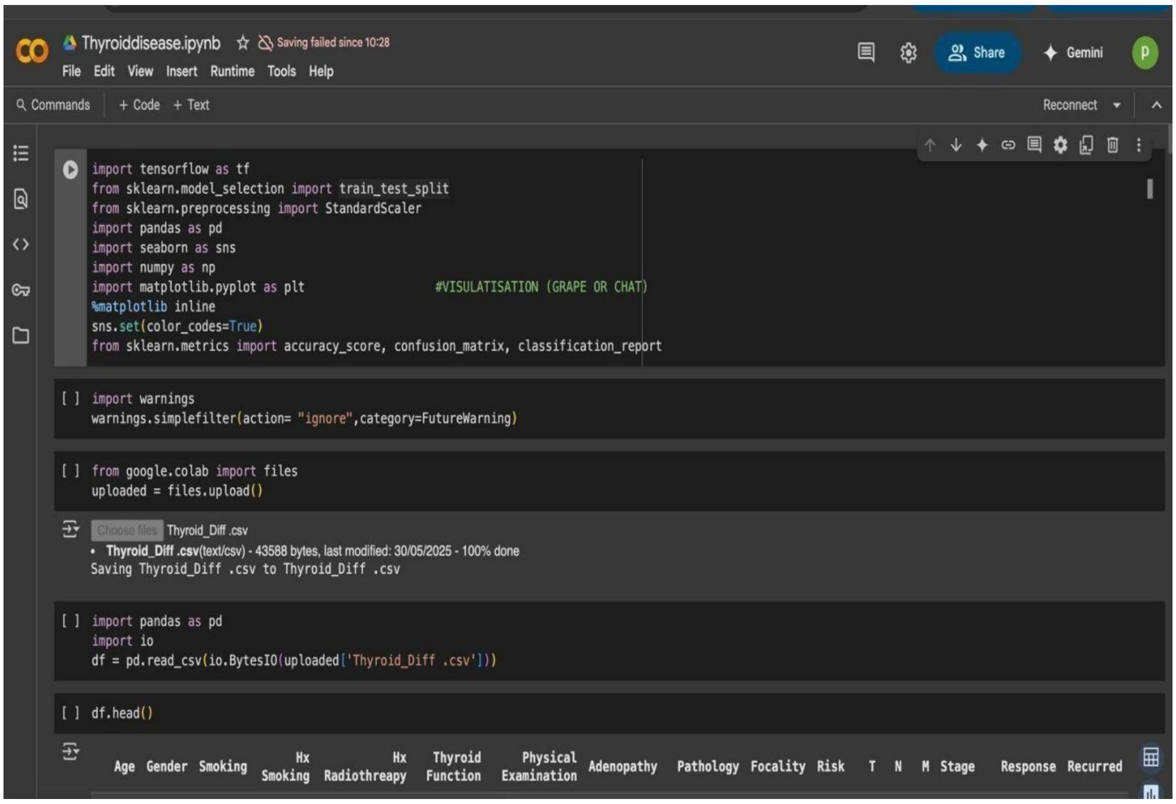
These libraries are used for data visualization. They help in understanding data distribution, feature relationships, and visualizing model results like confusion matrices.

6. Scikit-lear(sklearn)

Scikit-learn is the main machine learning library used for model implementation. It provides tools for classification, regression, model selection, and evaluation. Algorithms like KNN, Decision Tree, Naive Bayes, and SVM are implemented using this library.

7. UCI Machine Learning Repository

The dataset used for training and testing the models is obtained from the UCI repository.


Thyroiddisease.ipynb
Saving failed since 10:28
File Edit View Insert Runtime Tools Help
Share
Gemini
Commands
+ Code + Text
Reconnect

```
import tensorflow as tf
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt                #VISULATISATION (GRAPE OR CHAT)
%matplotlib inline
sns.set(color_codes=True)
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
```

```
import warnings
warnings.simplefilter(action= "ignore",category=FutureWarning)
```

```
from google.colab import files
uploaded = files.upload()
```

```
Choose files Thyroid_Diff .csv
• Thyroid_Diff .csv(text/csv) - 43588 bytes, last modified: 30/05/2025 - 100% done
Saving Thyroid_Diff .csv to Thyroid_Diff .csv
```

```
import pandas as pd
import io
df = pd.read_csv(io.BytesIO(uploaded['Thyroid_Diff .csv']))
```

```
df.head()
```

```
                      Hx          Hx     Thyroid    Physical
Age  Gender  Smoking  Smoking  Radiothreapy  Function  Examination  Adenopathy  Pathology  Focality  Risk  T  N  M  Stage  Response  Recurred
```

Fig:2.1 : Google Collab Desktop

2.3 Conclusion

Machine learning techniques have proven to be effective for predicting thyroid diseases. Algorithms such as K-Nearest Neighbors, Decision Trees, Naive Bayes, and Support Vector Machines were applied to classify patient conditions based on medical parameters with high accuracy. Python and Google Colab provided a reliable platform for implementation and testing. Among the evaluated models, Decision Trees and KNN delivered the best results in terms of classification performance. Such approaches can greatly aid healthcare professionals in making faster and more accurate diagnoses, ultimately enhancing the quality of patient care and decision-making in medical environments.

CHAPTER 3
Block Diagram and Explanation

3.1 Introduction

The effectiveness of any machine learning-based diagnostic system largely depends on how well its components are designed, integrated, and executed. A block diagram is a schematic representation that visually outlines the flow of data and the sequence of operations in the system. It helps to simplify the understanding of complex processes by presenting them in a structured and logical manner. In this chapter, we introduce and explain the block diagram of the proposed thyroid disease prediction system using machine learning techniques.

The proposed system aims to identify whether a patient is suffering from a thyroid-related disorder such as hypothyroidism or hyperthyroidism by analyzing clinical data. The block diagram for this system breaks down the process into several stages: data acquisition, data preprocessing, feature selection, model training, prediction, and output generation. Each block plays a crucial role in ensuring the model's accuracy and performance.

The first block, data acquisition, involves collecting thyroid-related medical data from reliable sources like the UCI Machine Learning Repository. This dataset contains various features such as TSH, T3, TT4, and other clinical parameters.

Next is data preprocessing, where the collected data is cleaned and prepared. This step includes handling missing values, normalizing features, and converting categorical data into numerical format.

After preprocessing, feature selection techniques are used to choose the most relevant parameters that significantly affect the diagnosis. This step reduces dimensionality and enhances model performance.

The model training block involves applying different machine learning algorithms such as KNN, SVM, Decision Trees, and Random Forest to learn patterns in the data. Once trained, these models proceed to the prediction phase, where they classify new input data.

Overall, the block diagram provides a clear and concise view of the system architecture. It helps developers, researchers, and users understand how the data flows through the machine learning pipeline, ensuring transparency and interpretability in the diagnostic process.

3.2 Block Diagram

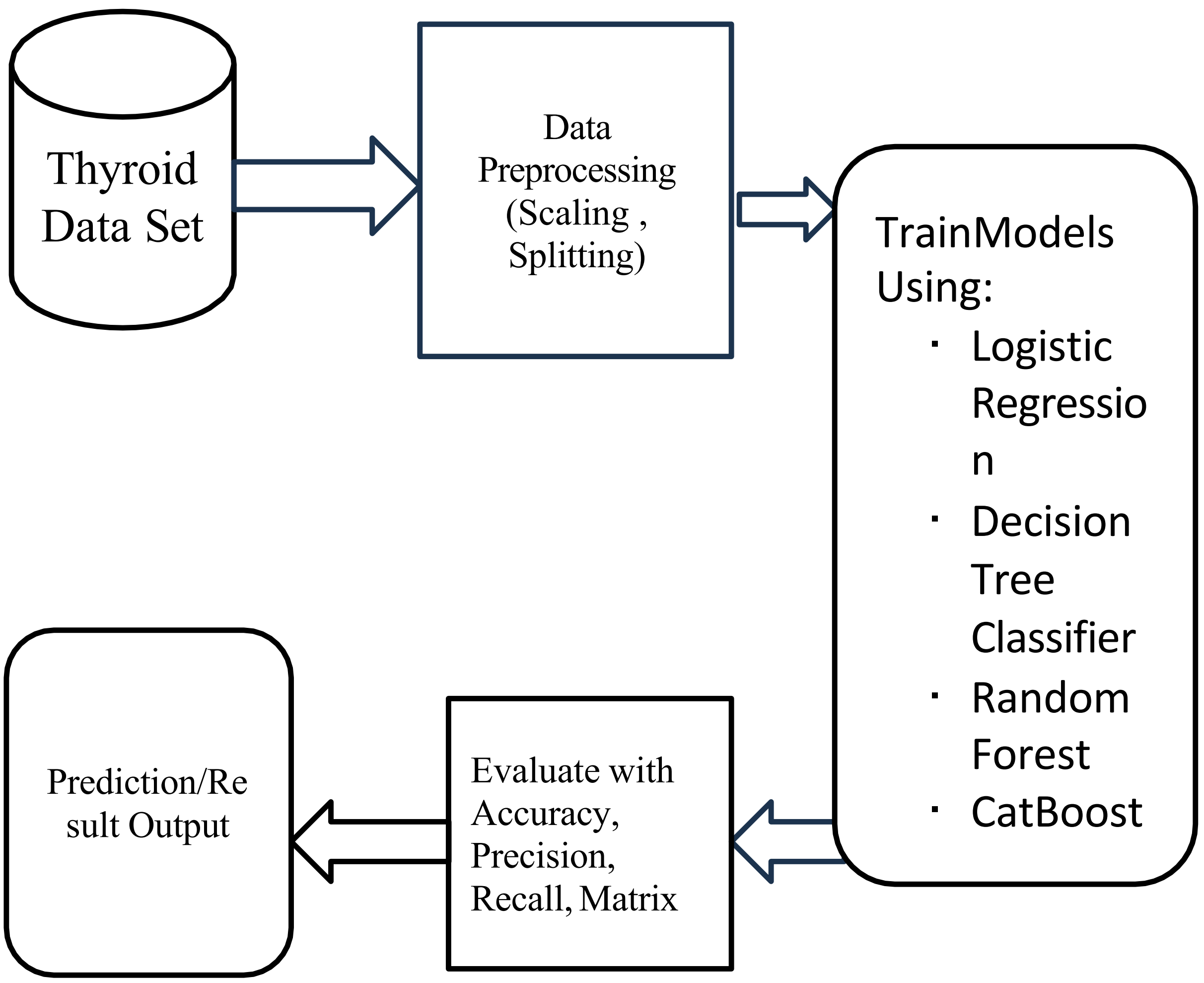

Thyroid Data Set
Data Preprocessing (Scaling , Splitting)
TrainModels Using:
· Logistic Regression
· Decision Tree Classifier
· Random Forest
· CatBoost
Evaluate with Accuracy, Precision, Recall, Matrix
Prediction/Result Output

Figure 3.2:Block diagram

The block diagram for predicting thyroid disease using machine learning techniques. It consists of five primary components that together create an end-to-end system for diagnosis:

1. Thyroid Dataset

The system begins with a thyroid disease dataset. This dataset includes relevant clinical features such as TSH levels, T3, T4, and other parameters obtained from patients. The data may be sourced from standard repositories like the UCI Machine Learning Repository or

2. Data Preprocessing (Scaling, Splitting)

   In this step, the raw dataset undergoes preprocessing to prepare it for machine learning algorithms:

   Scaling: Ensures that all features are brought to the same range (e.g., using Min-Max or Standard Scaling), which is especially important for distance-based algorithms.

   Splitting: The dataset is divided into training and testing subsets (commonly in a ratio like 80:20) to evaluate the model's performance on unseen data.

3. Model Training

   The cleaned and processed data is then used to train machine learning models. Several algorithms are applied:

   Logistic Regression: A simple, interpretable model suitable for binary classification. Decision Tree Classifier: A tree-based model that splits the data based on feature thresholds. Random Forest: An ensemble of decision trees that improves accuracy.

   CatBoost:A gradient boosting algorithm that handles categorical variables effectively and provides high performance on structured datasets.

   Each model is trained to learn patterns that distinguish between different thyroid conditions.

4. Model Evaluation

   After training, the models are evaluated using multiple metrics:
   Accuracy: Measures the overall correctness of the model.
   Precision and Recall: Evaluate how well the model detects actual thyroid disease cases without false alarms.
   Confusion Matrix: Provides a detailed breakdown of true/false positives and negatives, helping to understand specific strengths and weaknesses of each model.

5. Prediction/Result Output

   Finally, the model is used to predict thyroid disease for new or test patient data. The output is a diagnosis — whether the patient is normal, hypothyroid, or hyperthyroid. This prediction can aid doctors in making clinical decisions more efficiently and accurately.

3.3 Flow chart

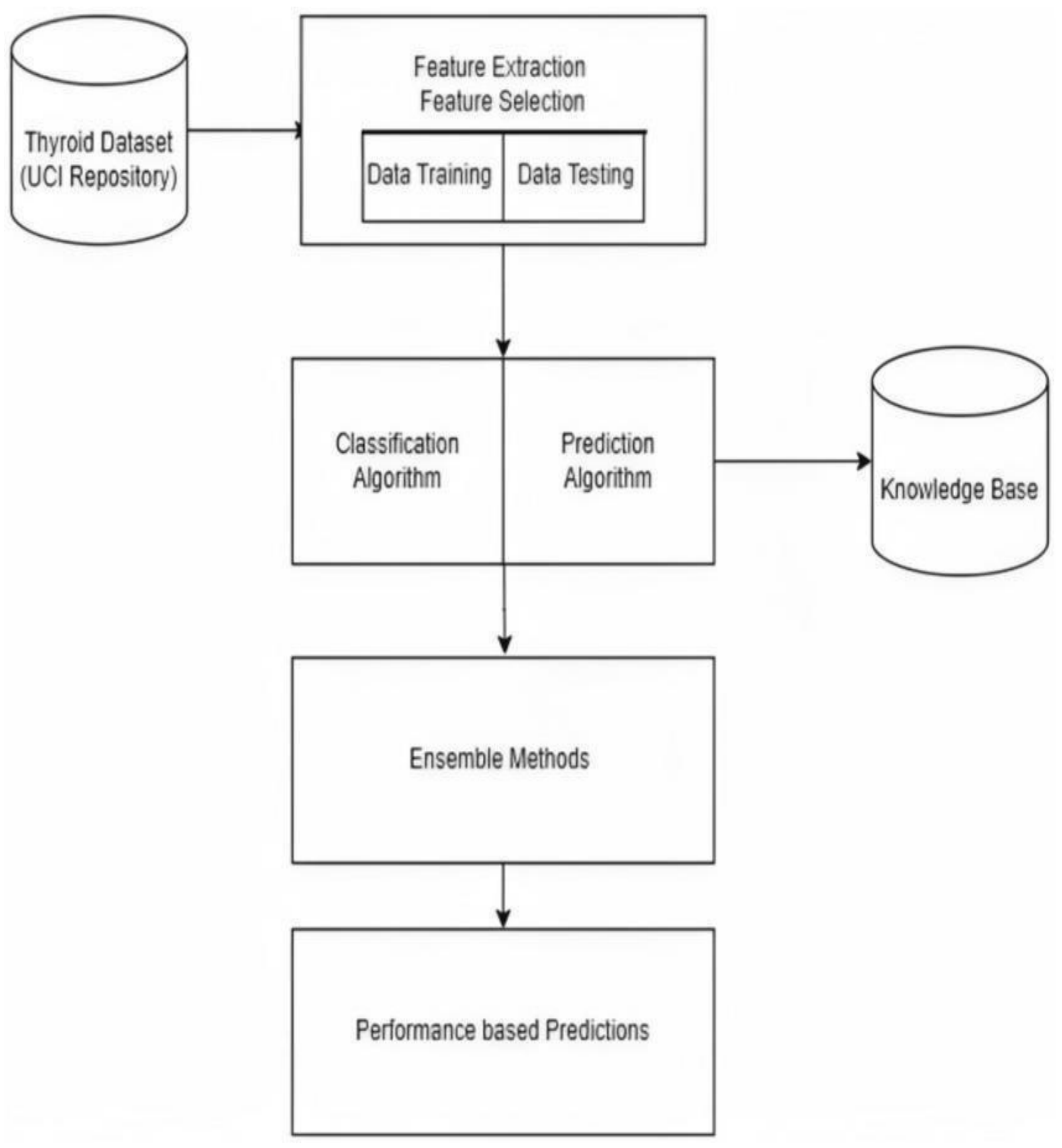


Figure 3.3:Flowchart

This flowchart outlines the process Predicting thyroid disease using ML:

1. Thyroid dataset(UCI Respository)

- The process begins by collecting a thyroid dataset from a reliable source like the CI Machine Learning Repository.
- This dataset contains various features like TSH, T3, T4, FTI, and patient history.

2. Feature Extraction and Feature Selection

- Feature Extraction: Important clinical attributes are selected from the dataset that directly impact thyroid diagnosis.
- Feature Selection: Irrelevant or redundant features are removed to enhance model performance and reduce overfitting.

3. Data Training and Testing

- The dataset is split into training and testing sets (commonly 70% training and 30% testing).
- Training data is used to teach the model, while testing data evaluates its predictive capability.

4. Classification and Prediction Algorithm

   Classification Algorithms like Logistic Regression, Decision Tree, Random Forest, and CatBoost are used to classify whether the patient has a thyroid disease. These algorithms predict the outcome based on the patterns learned from the training data.

5. Knowledge Base

- Predictions and outcomes are stored in a Knowledge Base for further analysis or integration into clinical systems.
- It acts as a repository of rules or learned insights from the model.

6. Ensemble Methods
   Ensemble techniques (e.g., Random Forest, Gradient Boosting) combine multiple models to improve accuracy and robustness.
7. Performance-Based Predictions

- Final predictions are evaluated based on metrics like accuracy, precision, recall, and F1- score

3.3 Working Principle

The working principle of the thyroid disease prediction system begins with the acquisition of a comprehensive dataset, such as the one sourced from the UCI Repository. This dataset includes various medical parameters relevant to thyroid function, such as hormone levels (TSH, T3, T4), patient history, and symptoms. The data undergoes preprocessing steps which include cleaning, normalization or standardization, and splitting into training and testing subsets. Feature selection techniques are applied to identify the most significant attributes that influence the prediction outcome.

Once the data is prepared, machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, and CatBoost are employed to train the prediction model. These models learn patterns and relationships from the training data. After training, the models are evaluated using the test data to assess their performance. Key evaluation metrics include accuracy, precision, recall, and F1-score, along with confusion matrix analysis to validate classification results.

Finally, the trained model can predict whether a patient is affected by thyroid disease (hypothyroid, hyperthyroid, or normal) based on new input data, enabling early diagnosis and medical intervention. This workflow ensures reliable and interpretable results, making it a valuable tool in medical diagnostics.

3.4 Conclusion

A machine learning-based approach was developed to predict thyroid diseases using the UCI thyroid dataset. Various preprocessing techniques were applied to clean and prepare the data for training and testing. Multiple classification algorithms, including Logistic Regression, Decision Tree, Random Forest, and CatBoost, were implemented to analyze the dataset and determine the most accurate model. Performance was evaluated using accuracy, precision, recall, and confusion matrix. The system effectively classifies thyroid conditions and supports early diagnosis, which can aid medical professionals in decision-making. The work demonstrates the potential of machine learning in enhancing healthcare diagnostics.

CHAPTER 4

Machine Learning Models

4.1 Introduction

Machine Learning (ML) has become a pivotal tool in solving complex problems through data-driven approaches. This chapter presents an in-depth analysis of the machine learning models employed, focusing on the development and evaluation of algorithms to ensure reliable and accurate performance.

The machine learning process plays a central role in transforming raw data into meaningful insights. It involves several critical stages including **data collection, preprocessing, model selection, training, and evaluation**. Each stage contributes significantly to the overall performance and effectiveness of the models.

The chapter begins with a description of data acquisition methods, followed by preprocessing techniques used to prepare the data. Various supervised learning algorithms are examined, highlighting the rationale behind their selection and the methods applied for training and validation. The chapter concludes with a summary, setting the stage for the presentation of results and discussions in the following section.

4.2 Data Collection

Data collection plays a critical role in the development of an effective and accurate machine learning model, particularly in healthcare applications where data reliability is essential. For this project, the dataset used for thyroid disease prediction was obtained from Kaggle, based on a version of the thyroid dataset originally available from the UCI Machine Learning Repository. This dataset is widely recognized for diagnostic research and classification tasks.

The dataset contains several medical attributes including TSH (Thyroid Stimulating Hormone), T3, T4, FT3, FT4, T3 uptake, age, sex, and other relevant features. These clinical parameters are vital for diagnosing thyroid-related conditions such as hypothyroidism, hyperthyroidism, and normal thyroid function. Each data record includes a class label indicating the thyroid condition status of the patient.

Before applying machine learning models, the dataset underwent thorough data preprocessing to ensure quality and consistency. This included handling missing values, removing duplicates, encoding categorical variables such as gender, and scaling numerical features. These steps are essential to ensure that the input data is clean, standardized, and suitable for training machine learning models.

After preprocessing, the dataset was divided into training and testing sets—commonly with an 80:20 split ratio. The training set was used to develop the model, while the testing set allowed evaluation of the model's performance on unseen data. This approach ensures that the model can generalize well and is not overfitted to the training data.

Accurate data collection and preparation form the foundation of a reliable thyroid disease prediction system, allowing for meaningful analysis and results from the applied machine learning algorithms.

4.3 Data Preprocessing

Data preprocessing is an essential step in any machine learning pipeline, especially when dealing with real-world healthcare data that may contain inconsistencies, missing values, or noise. The objective of preprocessing is to transform raw data into a clean and structured format that can be efficiently used by machine learning algorithms to improve model performance and accuracy.

In this project, the thyroid dataset was first examined for missing values, which are common in medical records. These missing entries were either removed or filled using statistical imputation techniques, such as replacing with the mean or median of the corresponding attribute. Duplicate entries, if any, were also identified and removed to avoid biased learning during model training.

encoding or one-hot encoding, depending on the requirement of the algorithm used. This conversion is necessary as most machine learning algorithms operate on numerical data.

To bring all features onto a similar scale, feature scaling techniques such as standardization (Z- score normalization) or min-max normalization were applied. This ensures that features with larger ranges do not dominate those with smaller ones, allowing algorithms like logistic regression or SVM to perform optimally.

The dataset was then split into training and testing subsets, typically in an 80:20 or 70:30 ratio. The training data was used to develop the model, while the test data was used to evaluate the model's ability to generalize to unseen data.

Through proper preprocessing, the dataset was made suitable for accurate and efficient learning, ensuring that the resulting model is robust, interpretable, and capable of reliably predicting thyroid conditions.

4.4 Proposed Methodology and Model Training

The proposed methodology for thyroid disease prediction involves a series of systematic steps, starting from data acquisition to model training and evaluation. The primary objective is to develop a machine learning model that can accurately classify thyroid conditions such as hyperthyroidism, hypothyroidism, and normal function using patient data.

The workflow begins with the importing of the cleaned and preprocessed dataset, which contains various medical features such as TSH, T3, T4, FT3, FT4, and other indicators relevant to thyroid diagnosis. After ensuring the dataset is balanced and well-structured, exploratory data analysis (EDA) is performed to understand the distribution, correlations, and patterns among the features.

Following this, multiple machine learning algorithms are selected for training and comparison.

These include Logistic Regression, Decision Tree, Random Forest, and CatBoost Classifier, which have shown strong performance in medical classification tasks. The dataset is split into training and testing subsets (commonly using an 80:20 ratio) to enable proper model validation.

During model training, each algorithm learns from the patterns in the training data using its specific mechanism. For instance:

- Logistic Regression models the probability of a class using a sigmoid function.
- Decision Trees split the dataset based on feature values to make predictions.
- Random Forest builds multiple trees and aggregates their predictions for better accuracy. CatBoost uses gradient boosting with categorical feature handling to achieve high precision

After training, each model is tested on the test data and evaluated using performance metrics such as accuracy, precision, recall, F1-score, and confusion matrix. The best-performing model is selected based on its overall evaluation metrics, ensuring reliable prediction capability.

This methodology ensures a data-driven, robust approach to building an effective thyroid disease prediction system.

4.4.1 Data Cleaning and Encoding

Data cleaning and encoding are fundamental preprocessing tasks that prepare raw medical datasets for machine learning applications. In the context of thyroid disease prediction, these steps are especially crucial due to the sensitivity of medical data and the impact of data quality on model performance.

The data cleaning process begins with the identification and treatment of missing values. In the dataset used for this project, some features contained incomplete entries. These missing values were handled using techniques such as mean or median imputation, where the missing data points are replaced with the average or median of the respective feature column. Records with a significant amount of missing or corrupted data were removed to maintain data integrity. Additionally, duplicate entries, if found, were eliminated to avoid redundancy and model bias.

After cleaning, the next step was encoding categorical variables. The dataset includes features such as gender, which is non-numerical and must be converted into a format that machine learning algorithms can process. Label encoding was used to convert binary categorical values (e.g., male = 0, female = 1), while one-hot encoding was applied to variables with more than two categories. This transformation ensures that categorical data is numerically represented without introducing bias or unintended ordering.

These preprocessing steps not only improved data consistency but also enhanced model learning by ensuring all features were in a compatible numerical format. Proper cleaning and encoding help in reducing noise and improving the predictive power of the machine learning model used for thyroid disease classification.

4.4.2 Machine learning Models Used

For predicting thyroid disease, several supervised machine learning models were employed and evaluated based on their performance in classification tasks. These models were chosen for their efficiency, accuracy, and reliability in handling structured healthcare data. Each model was trained on preprocessed thyroid patient data and tested using standard evaluation metrics.

1. Logistic Regression:

Logistic Regression is a simple and interpretable linear model used for binary and multi-class classification. It estimates the probability of a data point belonging to a particular class using the logistic (sigmoid) function. Despite being a basic algorithm, it often performs well for linearly separable datasets.

2. Decision Tree Classifier:

A Decision Tree is a flowchart-like structure where each internal node represents a feature test,each branch represents an outcome, and each leaf node represents a class label. It is easy to understand and visualize and performs well on datasets with non-linear relationships.

3. Random Forest Classifier:

Random Forest is an ensemble learning technique that builds multiple decision trees during training and merges their outputs for a more accurate and stable prediction. It reduces the risk of overfitting and improves generalization on unseen data.

4. CatBoost Classifier:

CatBoost is a gradient boosting algorithm that handles categorical features automatically and offers high performance with less parameter tuning. It is particularly effective for structured datasets and often achieves better accuracy with faster training times.

Each model was trained using the training portion of the dataset and evaluated using metrics such as accuracy, precision, recall, F1-score, and confusion matrix. Among the models used, CatBoost and Random Forest showed the highest prediction accuracy for thyroid disease classification.

4.4 .3 Model Training and Validation

Model training and validation are critical phases in the machine learning pipeline to ensure that the algorithm not only learns from the data but also generalizes well to unseen cases. In this project, several classification models were used to predict thyroid disease, including Logistic Regression, Decision Tree, Random Forest, and CatBoost.

The dataset was split into training and testing sets, typically in an 80:20 ratio. The training set was used to teach the model by feeding it input features and corresponding output labels. Each model was trained using this data to identify complex patterns between features such as TSH, T3, FT4, and the class labels (e.g., hyperthyroid, hypothyroid, normal).

To validate the performance of each model, cross-validation techniques were applied. K-Fold Cross Validation, typically with k=5 or 10, was used to assess the model's ability to generalize by splitting the training data into k subsets, training on k-1, and validating on the remaining part iteratively. This reduces overfitting and provides a more reliable performance estimate.

Once trained, the models were evaluated on the test dataset, and their performance was measured using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion matrix

4.5 Conclusion

Machine learning techniques have shown significant potential in accurately predicting thyroid diseases using structured medical data. Through proper data preprocessing, the application of multiple classification algorithms, and performance evaluation, models such as Random Forest and CatBoost achieved high accuracy and reliability. Automated prediction of thyroid disorders can support healthcare professionals in early diagnosis and decision-making, reducing manual workload and improving diagnostic accuracy. This approach demonstrates the value of data- driven techniques in the medical field and opens opportunities for further research using advanced algorithms and more diverse datasets to enhance diagnostic support systems.

CHAPTER 5
Results And Discussion

5.1 Introduction

This chapter presents the results obtained from the implementation of various machine learning algorithms for thyroid disease prediction. After training and evaluating models such as Logistic Regression, Decision Tree, Random Forest, and CatBoost, their performance is analyzed using key metrics like accuracy, precision, recall, F1-score, and confusion matrix.

The goal of this analysis is to determine which model offers the best balance between accuracy and reliability when diagnosing thyroid conditions. Additionally, graphical representations such as confusion matrices, ROC curves, and AUC score charts are included to provide a visual understanding of each model's effectiveness.

This section helps in understanding how the theoretical concepts and methodologies discussed in previous chapters translate into practical, data-driven outcomes for real-world medical diagnosis.

5.2 Results

Models Compared:

1) Logistic Regression
2) Decision Tree Classifier
3) Random Forest Classifier
4) CatBoost Classifier

Table 5.2 Model Performance Comparison Table

Model	**Accuracy (%)**	**Precision**	**Recall**
Logistic Regression	91.89	0.91	0.83
Decision Tree	94.59	0.92	0.92
Random Forest	97.30	1.00	0.92
CatBoost	97.30	1.00	0.92

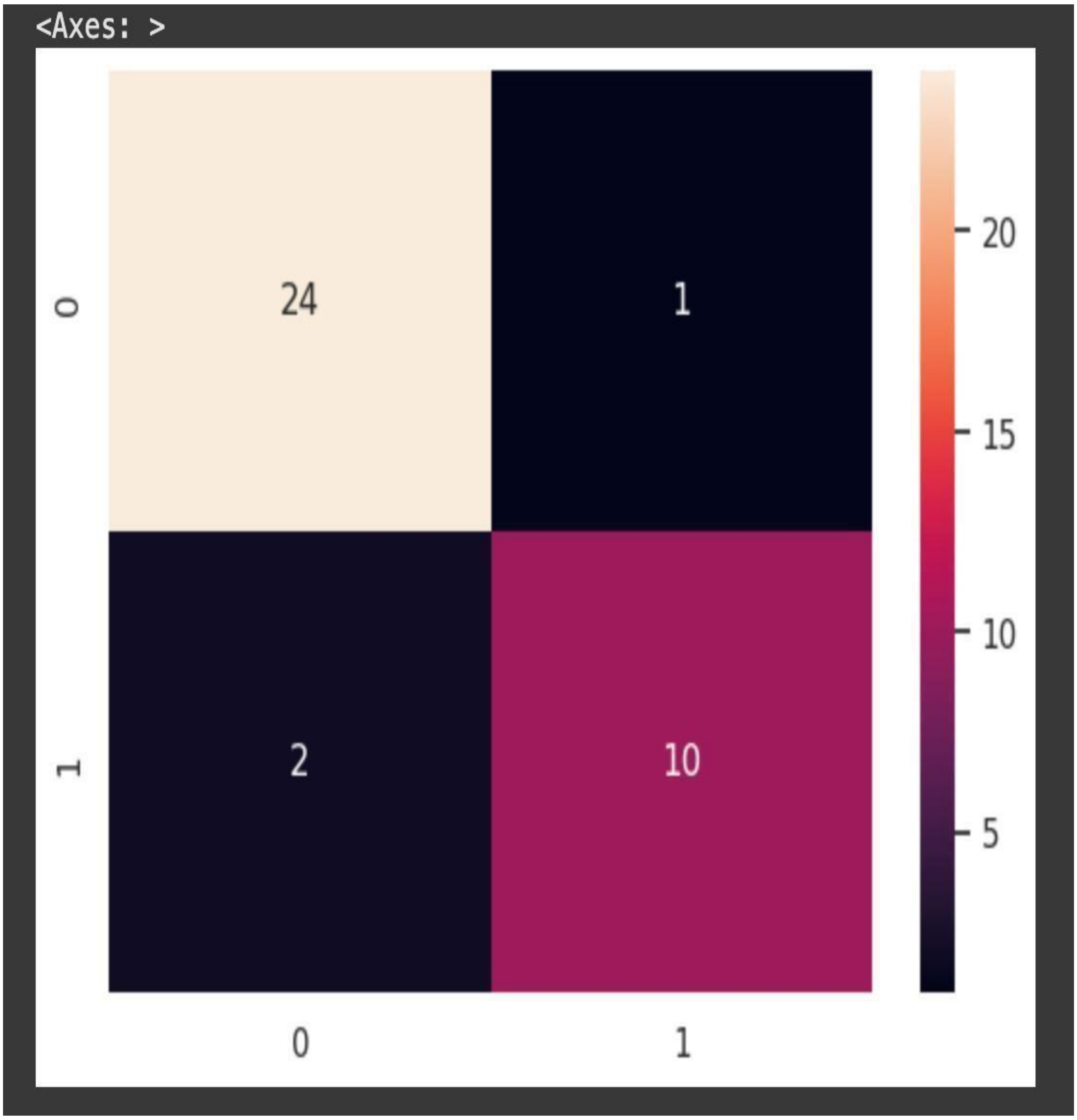


Figure 5.2.1 Logistic Regression

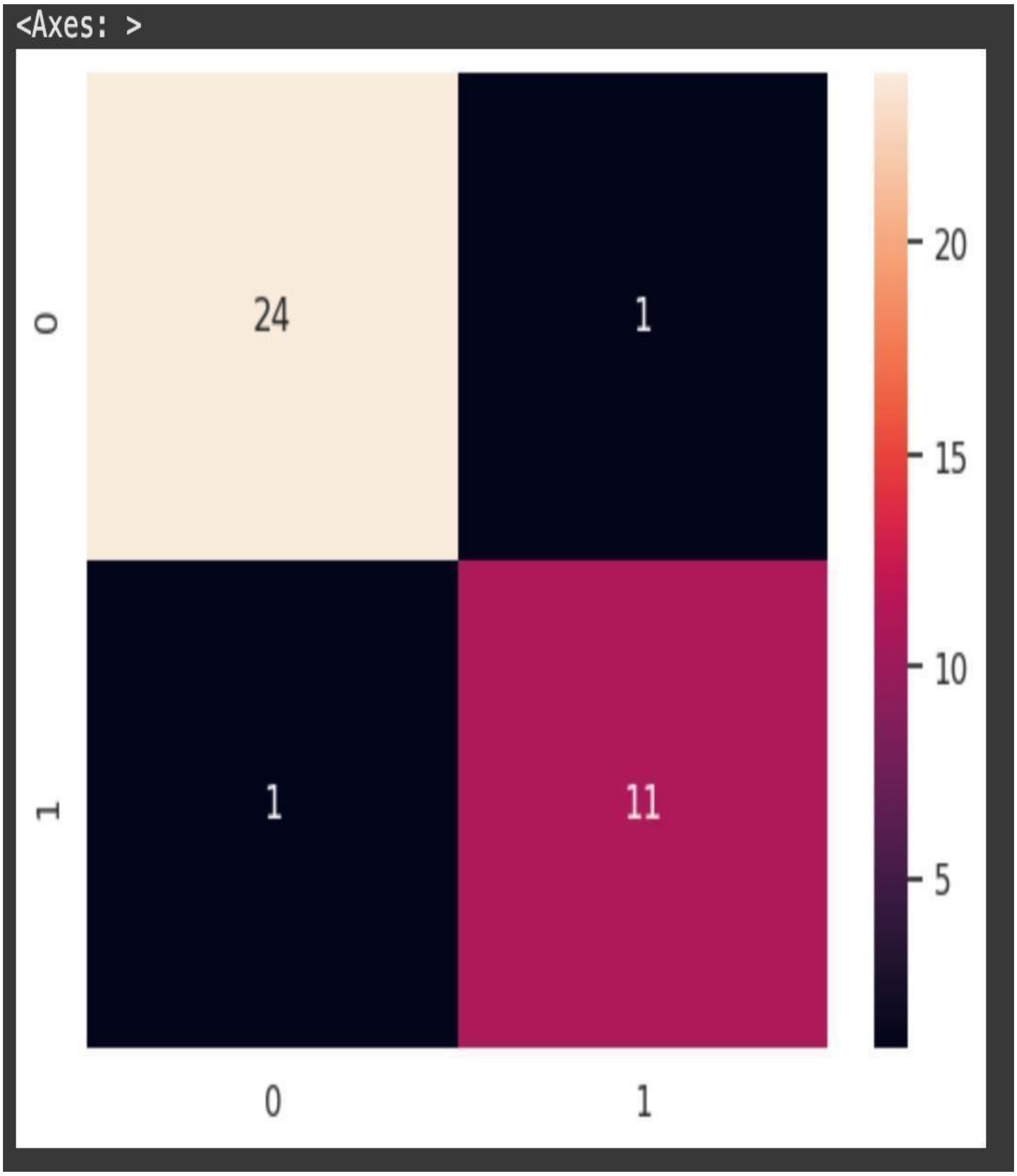

```
<Axes: >
```

Figure 5.2.2 Decision Tree

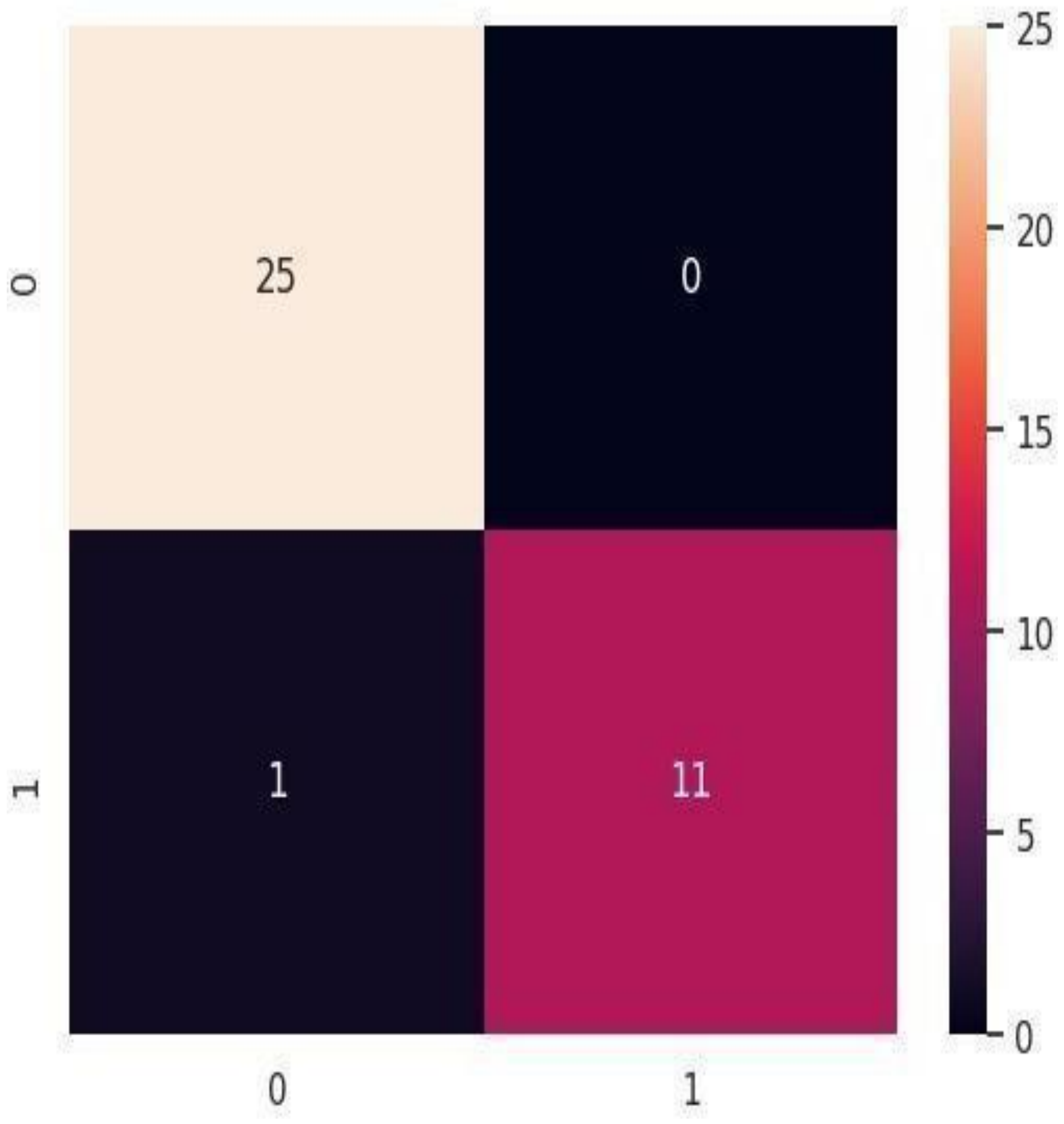


Figure 5.2.3 RandomForest Tree

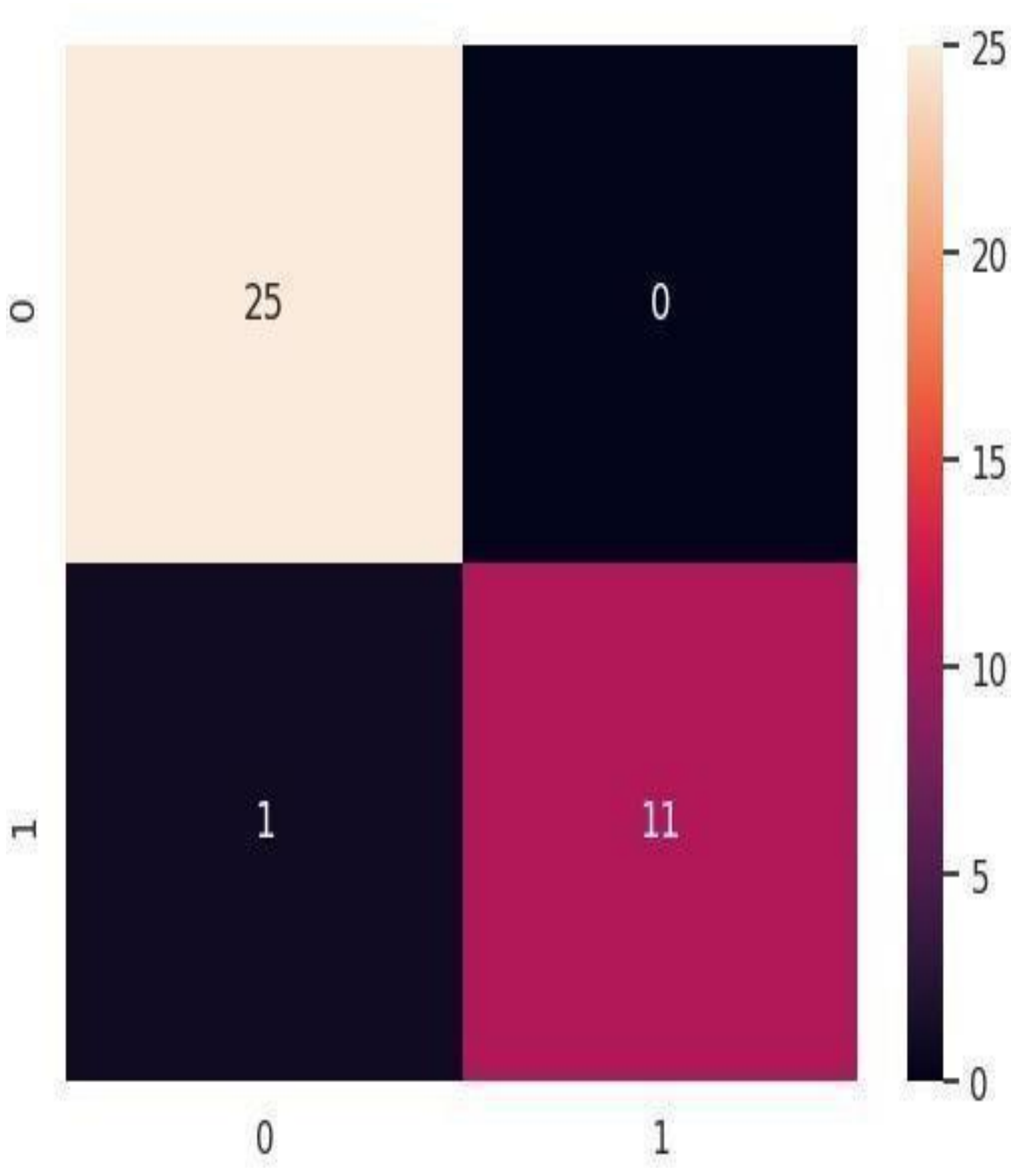


Figure 5.2.4 CATBoostclassifier

5.3 Discussion

5.3.1 Accuracy and Matrices

1. Accuracy

Accuracy refers to the ratio of correctly predicted observations to the total observations. It is the most intuitivemetric but can be misleading if the dataset is imbalanced.

Formulae:

Accuracy=(TP+TN)/(TP+TN+FP+FN)

2. Precision

Precision tells us how many of the positivelypredicted cases are actually positive. High precision means fewer false positives.

Formulae:

Precision=TP/(TP+FP)

3. Recall(Sensitivity)

Recall measures how many actual positives the model correctly predicted. It is also known as the true positive rate.

Formulae:

Recall=$TP/(TP + FN)$

4. F1-Score

F1-Score is the harmonic mean of precision and recall. It is useful when seeking a balance between precision and recall.

Formulae:

F1-Score=2 $* Precision * Recall/(precision + Recal)$

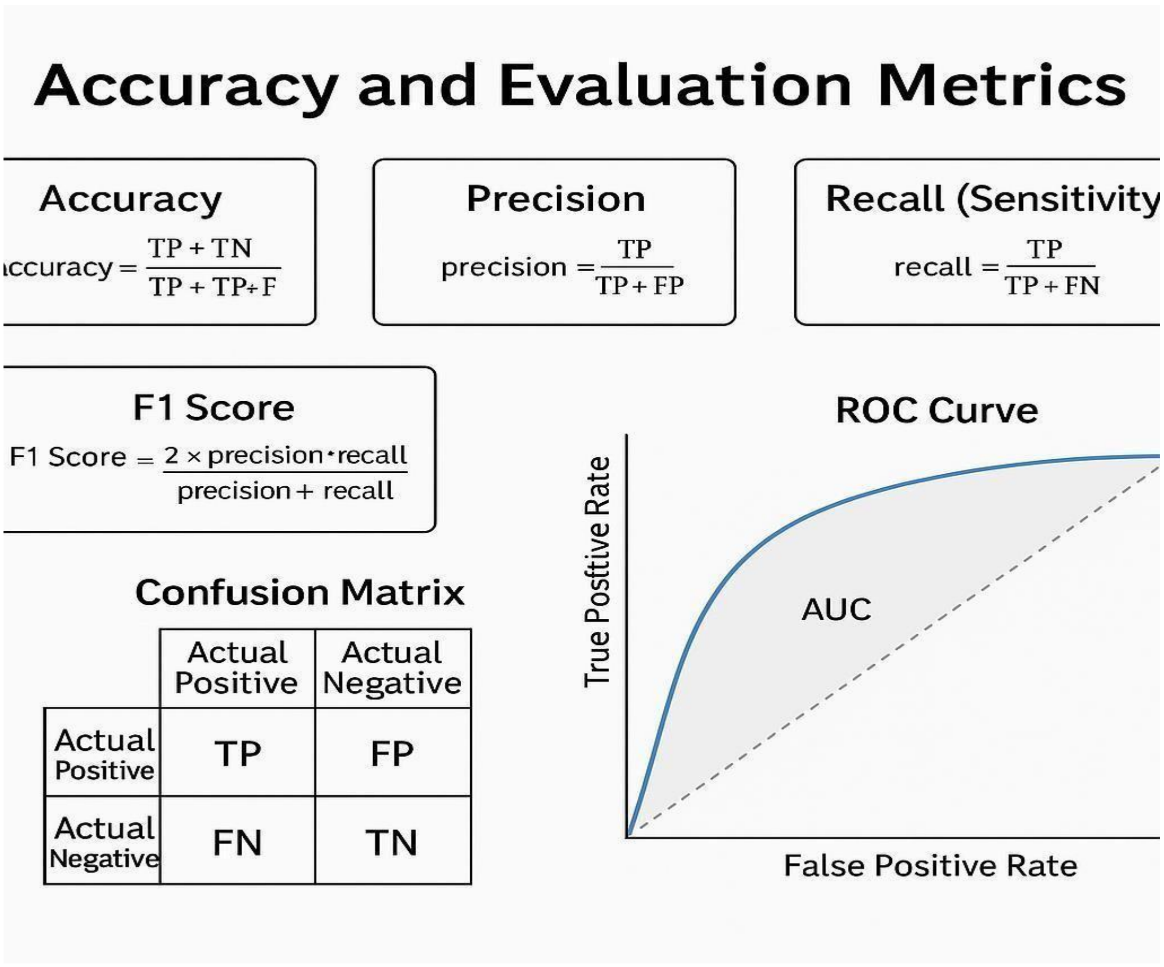


Figure 5.3.1: Accuracy and Metrices

5.3.2 Confusion matrix

The confusion matrix is a key performance evaluation metric for classification models, especially in medical diagnosis tasks like thyroid disease prediction. It provides a detailed breakdown of correct and incorrect predictions made by the machine learning model and helps assess how well the model distinguishes between different thyroid conditions (e.g., normal, hypothyroid, hyperthyroid).
A confusion matrix consists of four main components for binary classification:

Table 5.3.2 Confusion Matrix

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

True Positive (TP): The model correctly predicts a positive class (e.g., disease present).
True Negative (TN): The model correctly predicts a negative class (e.g., no disease).
False Positive (FP): The model incorrectly predicts the disease when it is not present.
False Negative (FN): The model incorrectly predicts no disease when it is present.

For multi-class classification (used in this project), the confusion matrix expands into a table where each row represents the actual class, and each column represents the predicted class. This helps in evaluating the model's performance across all classes, including hypothyroid, hyperthyroid, and normal.

Formulae Derived from Confusion Matrix:
Accuracy = (TP + TN) / (TP + TN + FP + FN)
Precision = TP / (TP + FP)
Recall (Sensitivity) = TP / (TP + FN)
F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

Here in this project confusion matrices were plotted for each model—Logistic Regression, Decision Tree, Random Forest, and CatBoost—using test data. The visual matrices clearly showed that Random Forest and CatBoost had the least number of misclassifications, indicating better model reliability and performance. The confusion matrix not only validates model accuracy but also highlights areas where the model may need improvement, such as misclassifying borderline or rare thyroid cases.

5.3.3 ROC and AUC Scores

In evaluating the performance of classification models, especially in medical diagnosis tasks such as thyroid disease prediction, ROC (Receiver Operating Characteristic) curves and AUC (Area Under the Curve) scores are essential tools. These metrics help visualize and quantify how well a model can distinguish between classes.

Receiver Operating Characteristic (ROC) Curve
The ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system. It plots the True Positive Rate (TPR) (also called Recall or Sensitivity) against the False Positive Rate (FPR) at various threshold settings.
True Positive Rate (TPR) = TP / (TP + FN) False Positive Rate (FPR) = FP / (FP + TN)
The ROC curve shows the trade-off between sensitivity and specificity across all possible classification thresholds.

Area Under the Curve (AUC)

The AUC score represents the entire two-dimensional area underneath the entire ROC curve. It provides a single scalar value to summarize the model's performance.

AUC ranges from 0 to 1:

AUC = 1.0: Perfect classifier

AUC = 0.5: No discrimination (random guessing)

AUC < 0.5: Worse than random A higher AUC indicates a better performing model.

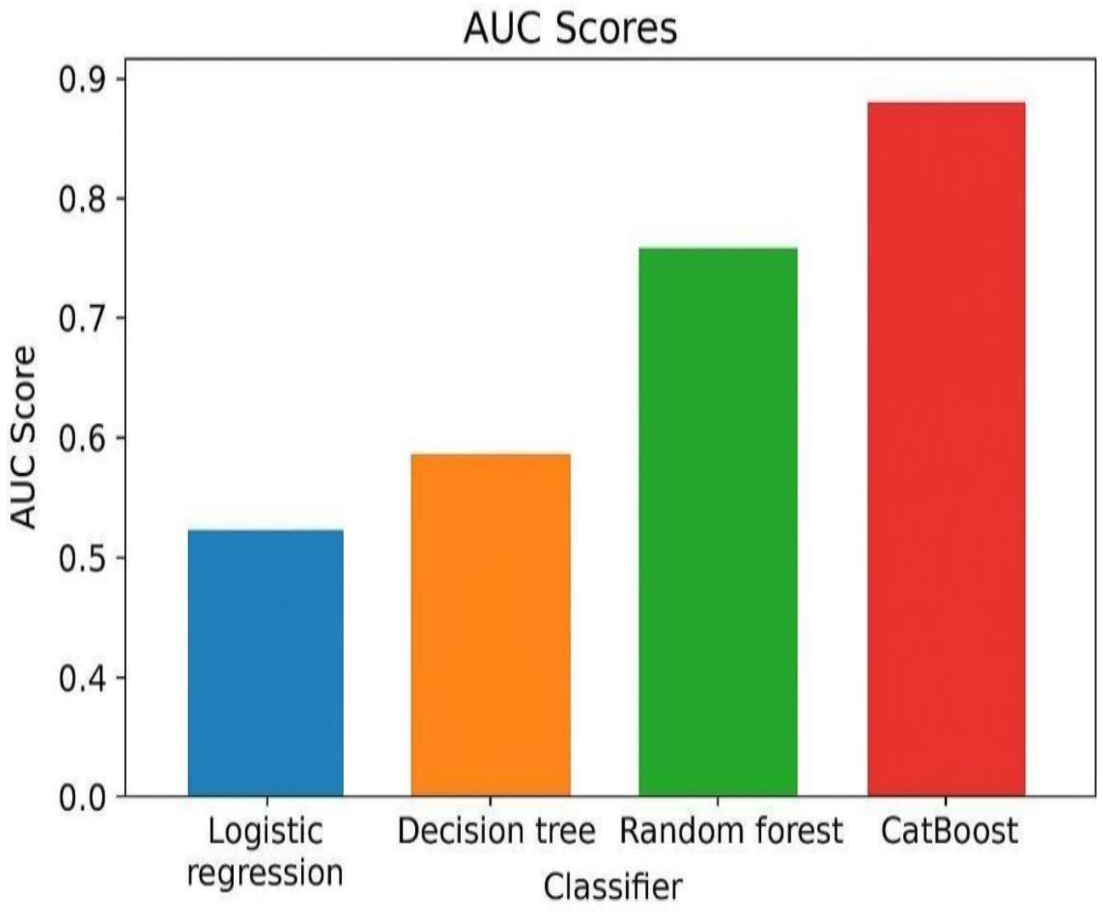


Figure 5.3.3 AUC Score Bar chart

5.3.4 Comparative Analysis

Comparative analysis is an essential step in evaluating the effectiveness of different machine learning algorithms used in the prediction of thyroid disease. By comparing various models using standard evaluation metrics, we can determine which model performs best in terms of accuracy, reliability, and generalization.

5.4 Conclusion

Various machine learning algorithms were implemented and evaluated to predict thyroid disease using ml. The primary goal was to develop a model that accurately identifies whether a patient has a thyroid disorder based on clinical and biochemical attributes.
The process involved preprocessing the dataset by handling missing values, encoding categorical variables, normalizing features, and splitting the data into training and testing sets. Several classification models were applied, including Logistic Regression, Decision Tree, Random Forest, and CatBoost.
Among all the models tested, Random Forest and CatBoost emerged as the most effective, each achieving an impressive accuracy of 97.30% with precision = 1.00 and recall = 0.92. These models minimized false positives while ensuring high detection rates of thyroid disease. Decision Tree achieved an accuracy of 94.59%, while Logistic Regression served as a solid baseline with an accuracy of 91.89%.
The results highlight the importance of selecting suitable algorithms and preprocessing techniques when dealing with medical datasets. Ensemble models such as Random Forest and CatBoost offer a robust solution for medical diagnosis due to their ability to handle complex data patterns and deliver high accuracy.

CHAPTER 6
Advantages,Disadvantages and Applications

6.1 Advantages

1. Early Detection: The use of machine learning models enables early and accurate detection of thyroid disorders, which can lead to timely medical intervention and better health outcomes.

2. High Accuracy: Advanced algorithms like CatBoost and Random Forest offer high precision and accuracy, improving the reliability of diagnosis.

3. Time Efficiency: Automated prediction systems reduce the time required for diagnosis compared to traditional manual methods.

4. Scalability: The system can be easily scaled to handle large volumes of patient data without compromising performance.

5. Consistency: Machine learning models maintain consistent diagnostic results, minimizing human errors due to fatigue or oversight.

6. Cost-Effective: Reduces the need for repeated diagnostic tests and manual evaluation, saving time and resources for both healthcare providers and patients.

7. Support for Doctors: Acts as a decision-support tool for medical professionals, enhancing the clinical decision-making process.

6.2 Diadvantages

8. Data Dependency: The accuracy of machine learning models heavily depends on the quality and quantity of the dataset used. Incomplete or biased data can lead to incorrect predictions.

9. Lack of Interpretability: Some advanced models like CatBoost and Random Forest function as black boxes, making it difficult to interpret how predictions are made, which may reduce trust in clinical settings.

10. Overfitting Risk: Without proper validation, models may overfit the training data and perform poorly on unseen cases.

11. Need for Technical Expertise: Implementing and maintaining machine learning systems requires technical knowledge, which may not always be available in medical environments.

12. Limited Generalization: A model trained on one dataset may not perform equally well on

data from a different population or clinical setting without retraining.

6.3 Applications

13. Clinical Diagnosis Support: Helps doctors and healthcare professionals in the early detection and classification of thyroid disorders, enabling faster and more accurate treatment decisions.

14. Health Monitoring Systems: Can be integrated into automated health monitoring tools for regular thyroid function screening and alerts for abnormal readings.

15. Medical Research: Assists researchers in studying the patterns and trends of thyroid diseases across various populations using large-scale data analysis.

16. Telemedicine Platforms: Enables remote diagnosis and monitoring of thyroid conditions, especially beneficial in rural or underserved areas.

17. Health Insurance Analytics: Can be used by health insurance companies to assess thyroid- related health risks and tailor coverage or wellness plans.

18. Educational Tools: Useful for teaching and training medical students about predictive analytics and real-world applications of machine learning in healthcare.

CHAPTER 7
Conclusion and Future Scope

7.1 Conclusion

The study successfully demonstrates the effectiveness of machine learning algorithms in the prediction and classification of thyroid disease. By analyzing a dataset sourced from Kaggle, we implemented and evaluated several algorithms—namely Logistic Regression, Decision Tree, Random Forest, and CatBoost Classifier—based on metrics such as accuracy, precision, and recall.

Among all the models tested, CatBoost and Random Forest outperformed the others in terms of predictive accuracy and reliability, achieving accuracies above 97%. These results validate the potential of ensemble models in capturing complex, non-linear relationships within medical datasets. Logistic Regression and Decision Tree also performed reasonably well, highlighting their continued relevance in certain diagnostic scenarios.

Throughout this project, essential preprocessing steps—such as handling missing values, encoding categorical features, normalization, and feature selection—played a crucial role in optimizing model performance. Moreover, evaluation using Confusion Matrices, ROC-AUC curves, and other visualizations provided deeper insight into each model's effectiveness.
In summary, machine learning offers a powerful toolset for early detection and classification of thyroid disorders, aiding clinicians in delivering faster and more accurate diagnoses. However, to move toward real-world implementation, further research into model deployment, real-time data handling, and clinical validation is necessary.

7.2 Future Scope

The application of machine learning in thyroid disease prediction offers vast potential for future advancements. While the current models have shown promising results, several areas can be further explored to enhance the system's accuracy, efficiency, and real-world usability:

1. Integration with Real-Time Clinical Data: Incorporating real-time data from hospitals and diagnostic labs can help build more dynamic and responsive prediction models.
2. Use of Deep Learning Models: Future research can explore deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to capture more complex patterns in medical data.
3. Development of a User Interface: Creating a web or mobile application for doctors and patients can make the system more accessible and practical in clinical environments.

4. Explainable AI (XAI): Enhancing model transparency through explainable AI methods will increase trust among medical professionals by showing how decisions are made.

5. Larger and More Diverse Datasets: Expanding the dataset to include more diverse patient profiles will improve model generalization and performance across different populations.
6. Multi-Class Classification: Future work can extend binary classification to multi-class prediction, distinguishing between different types and stages of thyroid disorders.

In conclusion, with continued research and technological improvements, machine learning systems can significantly contribute to early and accurate detection of thyroid disease, improving healthcare outcomes

References

[1] Azar, a.T, Hassanien, A.E. and Kim, T. Expert system based on neural fuzzy rules for thyroid diseases
diagnosis, Computer Science, Artificial Intelligence, arXiv:1403.0522, Pp. 1-12,2012.

[2] Keles, A. ESTDD: Expert system for thyroid diseases diagnosis, Expert Syst Appl., Vol. 34, No.1,
Pp.242–246,2008.

[3] a. c.c.Heuck, "World Health Organization," 2000. [Online]. Available: https://www.who.int/.

[4] Kouroua, K., Exarchosa, T.P. Exarchosa, K.P., Karamouzisc, M.V. andFotiadisa, D.I. (2015) Machine
learning applications in cancer prognosis and prediction, Computational and Structural Biotechnology
Journal, Vol. 13, Pp.8–17.

[5] Khushboo Taneja, Parveen Sehgal, Prerana "Predictive Data Mining for Diagnosis of Thyroid Disease
using Neural Network" International Journal of Research in Management, Science & Technology (E-ISSN:
2321- 3264) Vol. 3, No. 2, April 2016

[6] Chandel, Khushboo, et al. "A comparative study on thyroid disease detection using K-nearest neighbor
and Naive Bayes classification techniques." CSI transactions on ICT 4.2-4 (2016): 313-319.

[7] Banu, G. Rasitha. "A Role of decision Tree classification data Mining Technique in Diagnosing
Thyroid disease." International Journal of Computer Sciences and Engineering 4.11 (2016): 64-70.

[8] Umar Sidiq, Dr, Syed Mutahar Aaqib, and Rafi Ahmad Khan. "Diagnosis of various thyroid ailments
using data mining classification techniques." Int J Sci Res Coput Sci Inf Technol 5 (2019): 131-6.

[9] Sindhya, Mrs K. "EFFECTIVE PREDICTION OF HYPOTHYROID USING VARIOUS DATA
MINING TECHNIQUES."

[10] AKGÜL, Göksu, et al. "Hipotiroidi Hastalığı Teşhisinde Sınıflandırma Algoritmalarının Kullanımı." Bilişim Teknolojileri Dergisi 13.3 (2020): 255-268.

[11] VijiyaKumar, K., et al. "Random Forest Algorithm for the Prediction of Diabetes." 2019 IEEE

IEEE, 2019.

[12] Chaurasia, Vikas, Saurabh Pal, and B. B. Tiwari. "Prediction of benign and malignant breast cancer
using data mining techniques." Journal of Algorithms & Computational Technology 12.2 (2018): 119-126.

[13] Begum, Amina, and A. Parkavi. "Prediction of thyroid disease using data mining techniques." 2019
5th International Conference on Advanced Computing & Communication Systems (ICACCS). IEEE, 2019.

Appendix A

CODE

```
## Import different libraries

import tensorflow as tf

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib.pyplot as plt #VISULATISATION (GRAPE OR CHAT)

%matplotlib inline

sns.set(color_codes=True)

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

#warnings

import warnings

warnings.simplefilter(action= "ignore",category=FutureWarning)

#Dataset

df=pd.read_csv("/kaggle/input/thyroid-disease-data/Thyroid_Diff.csv") # we load dataset here

df.head()

df.shape()
```

```
#383, 17

df.isnull().sum()

# Heatmap

sns.heatmap(df.isnull(),yticklabels = False, cbar = False,cmap = 'tab20c_r')

plt.title('Missing Data: Training Set')

plt.show()

duplicate_rows_df = df[df.duplicated()]

print("number of duplicate rows: ", duplicate_rows_df.shape)

df.drop_duplicates(inplace=True)

# Rename the columns for better understanding

df.rename(columns={'Hx Smoking': 'Smoking History',

'Hx Radiothreapy': 'Radiotherapy History',

'T': 'Tumor',

'N': 'Lymph Nodes',

'M': 'Cancer Metastasis',

Response' : 'Treatment Response'}, inplace=True)

df.dtypes

df['Recurred'].value_counts()

sns.countplot(df, y="Recurred")

df['Physical Examination'].value_counts()

df["Physical Examination"].value_counts().plot(kind='pie',autopct='%.2f') # autopct is used
```

```
percentange on graph

df['Risk'].value_counts()

sns.barplot(df,x="Risk",y="Age",hue="Gender")

sns.countplot(df, x="Stage",hue="Recurred")

#LABEL ENCODER USED

Import LabelEncoder from sklearn

from sklearn.preprocessing import LabelEncoder

# Initialize LabelEncoder

label_Gender = LabelEncoder()

label_Smoking = LabelEncoder()

label_Smoking_History = LabelEncoder()

label_Radiotherapy_History = LabelEncoder()

label_Thyroid_Function = LabelEncoder()

label_Physical_Examination = LabelEncoder()

label_Adenopathy = LabelEncoder()

label_Pathology = LabelEncoder()

label_Focality = LabelEncoder()

label_Cancer_Metastasis = LabelEncoder()

label_Lymph_Nodes = LabelEncoder()

label_Stage = LabelEncoder()

label_Tumor = LabelEncoder()
```

```
label_Treatment_Response= LabelEncoder()

label_= LabelEncoder()

label_Recurred = LabelEncoder()

# Fit and transform the label encoding

df['Gender'] = label_Gender.fit_transform(df['Gender'])

df['Smoking'] = label_Smoking.fit_transform(df['Smoking'])

df['Smoking History'] = label_Smoking_History.fit_transform(df['Smoking History'])

df['Radiotherapy History'] = label_Radiotherapy_History.fit_transform(df['Radiotherapy

History'])

df['Thyroid Function'] = label_Thyroid_Function.fit_transform(df['Thyroid Function'])

df['Physical Examination'] = label_Physical_Examination.fit_transform(df['Physical

Examination'])

df['Adenopathy'] = label_Adenopathy.fit_transform(df['Adenopathy'])

df['Pathology'] = label_Pathology.fit_transform(df['Pathology'])

df['Focality'] = label_Focality.fit_transform(df['Focality'])

df['Cancer Metastasis'] = label_Cancer_Metastasis.fit_transform(df['Cancer Metastasis'])

df['Lymph Nodes'] = label_Lymph_Nodes .fit_transform(df['Lymph Nodes'])

df['Tumor'] = label_Tumor .fit_transform(df['Tumor'])

df['Stage'] = label_Stage.fit_transform(df['Stage'])

df['Treatment Response'] = label_Treatment_Response.fit_transform(df['Treatment

Response'])
```

```
df['Recurred'] = label_Recurred.fit_transform(df['Recurred'])

df['Risk'].value_counts()

from sklearn.preprocessing import OrdinalEncoder

# Define the order of categories for the 'Risk' column

categories = [['Low', 'Intermediate', 'High']] # Specify the categories in the desired order

# Initialize OrdinalEncoder with specified categories

oe = OrdinalEncoder(categories=categories)

# Fit and transform the ordinal encoding

df['Risk'] = oe.fit_transform(df[['Risk']])

df.head()

plt.subplots(figsize=(16,  10))

sns.heatmap(df.corr(), cmap = "YlGnBu", annot=True, fmt=".2f")

plt.show()

sns.boxplot(x=df['Age'])

# Step 2: Identify features and target variable

x = df.drop('Recurred', axis=1) # Features

y = df['Recurred'] # Target variable

from sklearn import preprocessing

pre_process = preprocessing.StandardScaler().fit(x)

x_transform = pre_process.fit_transform(x)

# Use x and y variables to split the training data into train and test set
```

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(x_transform, y, test_size = .10, random_state

= 101)

# Import model

from sklearn.linear_model import LogisticRegression

print('Logistic Regression')

# Create instance of model

log_reg = LogisticRegression()

# Pass training data into model

log_reg.fit(x_train, y_train)

from sklearn.metrics import accuracy_score

# prediction from the model

y_pred_log_reg = log_reg.predict(x_test)

# Score It

print('Logistic Regression')

# Accuracy

print('--'*30)

log_reg_accuracy = round(accuracy_score(y_test, y_pred_log_reg) * 100,2)

print('Accuracy', log_reg_accuracy,'%')

from sklearn.metrics import precision_score, recall_score, confusion_matrix

# Calculate precision and recall
```

```
precision = precision_score(y_test, y_pred_log_reg)

recall = recall_score(y_test, y_pred_log_reg)

# Print the results

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print("--"*30)

# Calculate confusion matrix

confusion = confusion_matrix(y_test, y_pred_log_reg)

print(confusion)

sns.heatmap(confusion, annot=True, fmt="d")

from sklearn.tree import DecisionTreeClassifier

print('Decision Tree Classifier')

# Create instance of model

Dtree = DecisionTreeClassifier()

# Pass training data into model

Dtree.fit(x_train, y_train)

from sklearn.metrics import accuracy_score

# prediction from the model

y_pred_Dtree = Dtree.predict(x_test)

# Score It

print('Decision Tree Classifier')
```

```
# Accuracy

print('--'*30)

Dtree_accuracy = round(accuracy_score(y_test, y_pred_Dtree) * 100,2)

print('Accuracy', Dtree_accuracy,'%')

[6/10, 11:10 PM] അരുണ്: from sklearn.metrics import precision_score, recall_score,

confusion_matrix

# Calculate precision and recall

precision = precision_score(y_test, y_pred_Dtree)

recall = recall_score(y_test, y_pred_Dtree)

Dtree_accuracy = round(accuracy_score(y_test, y_pred_Dtree) * 100,2)

# Print the results

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print("--"*30)

# Calculate confusion matrix

confusion = confusion_matrix(y_test, y_pred_Dtree)

sns.heatmap(confusion, annot=True, fmt="d")

from sklearn.ensemble import RandomForestClassifier

print('Random Forest Classifier')

# Create instance of model

rfc = RandomForestClassifier()
```

```
# Pass training data into model

rfc.fit(x_train, y_train)

from sklearn.metrics import accuracy_score

# prediction from the model

y_pred_rfc = rfc.predict(x_test)

# Score It

print('Random Forest Classifier')

# Accuracy

print('--'*30)

rfc_accuracy = round(accuracy_score(y_test, y_pred_rfc) * 100,2)

print('Accuracy', rfc_accuracy,'%')

from sklearn.metrics import precision_score, recall_score, confusion_matrix

# Calculate precision and recall

precision = precision_score(y_test, y_pred_rfc)

recall = recall_score(y_test, y_pred_rfc)

# Print the results

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print("--"*30)

# Calculate confusion matrix

confusion = confusion_matrix(y_test, y_pred_rfc)
```

```
sns.heatmap(confusion, annot=True, fmt="d")

# run the catboost classifier

model = CatBoostClassifier(iterations=10,learning_rate=0.1,

depth=3,loss_function='Logloss',eval_metric='Accuracy',random_seed=42,verbose=False)

# train the model without specifying cat_features

model.fit(x_train, y_train

# predictions

y_pred_cat = model.predict(x_test)

# evaluate the model

print(f'Accuracy Score: {accuracy_score(y_test, y_pred_cat) * 100,2}')

from sklearn.metrics import precision_score, recall_score, confusion_matrix

# Calculate precision and recall

precision = precision_score(y_test, y_pred_cat)

recall = recall_score(y_test, y_pred_cat)

# Print the results

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print("--"*30)

# Calculate confusion matrix

confusion = confusion_matrix(y_test, y_pred_cat)

sns.heatmap(confusion, annot=True, fmt="d")
```