

THYROID DISEASE PREDICTION: A MACHINE LEARNING APPROACH IN R

Submitted in partial fulfilment of the requirements for the award of

MASTER OF SCIENCE IN BIOINFORMATICS AND DATA SCIENCE

By

SUDHAKARAN R (42739024)



**DEPARTMENT OF BIOINFORMATICS
SCHOOL OF BIO AND CHEMICAL ENGINEERING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

APRIL – 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with "A++" grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF BIOINFORMATICS

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **ANITHAA.A(42739003)**, who carried out the final year project entitled **"THYROID DISEASE PREDICITON: A MACHINE LEARNING APPROACH IN R"** under our supervision from September 2023 to April 2024.

Internal Guide

Dr. Swetha Sunkar

External Guide

Dr. Swetha Sunkar

Head of the Department

Dr. Jemmy Christy

Submitted for Viva voce Examination held on

20/04/2024

Internal Examiner

External Examiner

DECLARATION

I, **R. SUDHAKARAN**, hereby declare that the Project Report entitled “**THYROID DISEASE PREDICTION: A MACHINE LEARNING APPROACH IN R**” done by me under the guidance of **Dr. SWETHA SUNKAR** at Sathyabama Institute of Science and Technology and is submitted in partial fulfilment of the requirements for the award of Master of Science in Bioinformatics and Data Science.

DATE: 20/4/2024

PLACE: CHENNAI



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENTS

I am delighted to express my heartfelt acknowledgement to the **Board of Management of SATHYABAMA** for their kind encouragement and support throughout the duration of my project. Their guidance has been invaluable in ensuring the successful completion of this project, and I am sincerely grateful for their contribution.

I convey thanks to **Dr. Rekha Chakravarthi**, Dean, School of science and humanities, for providing necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to **Dr. H. Jemmy Christy**, Head of the Bioinformatics Department, for her support has been instrumental in helping me to overcome obstacles and achieve success in this project.

I am deeply indebted to my Project Guide, **Dr. Swetha Sunkar**, for her invaluable guidance, suggestions, and constant encouragement throughout the project. Her expertise and insights have been instrumental in shaping my work and ensuring its successful completion.

I wish to extend my gratitude to all Teaching and Non-Teaching staff members for their support and assistance in various ways, which have contributed to the successful completion of this project.

ABSTRACT

This study aims to develop machine learning models in R for predicting thyroid diseases. Through comprehensive exploratory data analysis (EDA) on a publicly available dataset, various patterns related to thyroid conditions across different demographic groups and thyroid function indicators were identified. Following meticulous data preprocessing steps to ensure data quality, including handling missing values and encoding categorical variables, several machine learning algorithms such as Random Forest, Decision Tree, and Naive Bayes were trained and evaluated. The Random Forest model emerged as the top performer with an accuracy of 99.21%, indicating its potential for accurate thyroid disease prediction. Additionally, the Decision Tree model showed promising results with an accuracy of 97.38%. Although the Naive Bayes model exhibited lower accuracy, its high precision suggests its applicability in certain clinical scenarios. Future research should focus on further refining these models and exploring additional algorithms to enhance thyroid disease diagnosis and management. Moreover, efforts to improve model interpretability and deploy them in real-time clinical settings hold promise for improving healthcare outcomes for individuals with thyroid disorders.

Keywords: Thyroid disease prediction, Machine learning, Exploratory data analysis, Random Forest, Decision Tree, Naive Bayes

TABLE OF CONTENTS

S.NO	TOPICS	PAGE NO
	Abstract	I
	List of Abbreviations	V
	List of Tables	VI
	List of Figures	VI
1	INTRODUCTION	1
	1.1 Thyroid Disease	1
	1.2 Types of Thyroid disease	1
	1.2.1 Conditions that can cause hypothyroidism	1
	1.2.2 Conditions that can cause hyperthyroidism	2
	1.2.3 Conditions that can cause both hypothyroidism and hyperthyroidism	2
	1.3 Treatment	2
	1.4 Symptoms	3
	1.5 Machine Learning machine learning techniques for medical diagnosis	3
	1.5.1 Supervised learning	3
	1.5.2 Unsupervised learning	3
	1.5.3 Semi-supervised learning	3
	1.6 Exploratory Data Analysis (EDA)	4
	1.7 Classification Algorithms	4
	1.7.1 Random Forest	5
	1.7.2 Decision Tree	5
	1.7.3 Naive Bayes	5
	1.8 Purpose of the study	6
2	REVIEW OF LITERATURE	7
3	AIM AND SCOPE	9
	3.1 AIM	9
	3.2 SCOPE	9
4	METHODOLOGY	10
	4.2 Data Preprocessing	10
	4.3 Exploratory Data Analysis (EDA)	10
	4.4 Feature Selection	10
	4.5 Model Development	11

4.6	Model Comparison	11
4.7	Validation and Testing	11
4.8	Software and Tools	11
4.8.1	R Programming Language	11
4.8.2	RStudio	12
4.8.3	Essential Packages	12
4.9	Machine Learning	12
4.10	Flowchart for Thyroid Prediction Model Development	13
5	RESULTS & DISCUSSION	14
5.1	Data Collection	14
5.1.1	Dataset Description	14
5.1.2	Dataset Structure	14
5.2	Data preprocessing	15
5.3	Exploratory data analysis (EDA)	16
5.3.1	Gender Distribution	16
5.4	Univariate Analysis:	17
5.4.1	Strip Plot Parameter T3	17
5.4.2	Strip Plot Parameter T4	18
5.4.3	Strip Plot Parameter TSH	19
5.4.4	Density plot	19
5.4.5	T3 Density Distribution	20
5.4.6	T4 Density Distribution	20
5.4.7	TSH Density Distribution	22
5.5	Multivariate analysis	23
5.5.1	Thyroid Test Results Distribution	23
5.5.2	Distribution of multiple parameters	24
5.5.3	Kernel Density Estimate of Age	26
5.5.4	Number of Thyroid Disorders for Different Age Groups	27
5.6	Model Development	28
5.6.1	Visualizing Training set and Testing set	28
5.6.2	Random Forest Model	29
5.6.2.1	Confusion matrix of RF model	30
5.6.2.2	Correlation matrix of RF model	31

5.6.3	Decision Tree model	32
5.6.3.1	Confusion matrix of DT model	33
5.6.3.2	Correlation matrix heatmap of DT model	34
5.6.4	Naive Bayes model	35
5.6.4.1	Confusion matrix of NB model	36
5.6.4.2	Correlation matrix heatmap of NB model	37
5.7	Evaluation Table	39
6	SUMMARY AND CONCLUSION	40
6.1	Summary	40
6.2	Conclusion	41
7	REFERENCE	42
8	APPENDIX	45

LIST OF ABBREVIATION

T3	Triiodothyronine
T4	Thyroxine
TSH	Thyroid-stimulating hormone
ML	Machine Learning
EDA	Exploratory Data Analysis
RF	Random Forest
DT	Decision Tree
NB	Naive Bayes
KDE	Kernel Density Estimate

LIST OF TABLES

TABLES	TITLE	PAGE NO
1	Dataset	14
2	Evaluation results	39

LIST OF FIGURES

FIGURES	TITLE	PAGE NO
1	Flowchart for model development	13
2	Distribution of gender	16
3	Strip Plot Parameter T3	17
4	Strip Plot Parameter T4	18
5	Strip Plot Parameter TSH	19
6	T3 Density Distribution	20
7	T4 Density Distribution	21
8	TSH Density Distribution	22
9	Thyroid Test Results Distribution	23
10	Distribution of multiple parameters	24
11	Kernel Density Estimate of Age	26
12	Number of Thyroid Disorders for Different Age Groups	27
13	Visualizing Training set and Testing set	28
14	Confusion matrix of RF model	30
15	Correlation matrix heatmap of RF model	31
16	Confusion matrix of DT model	33
17	Correlation matrix heatmap of DT model	34
18	Confusion matrix of NB model	36
19	Correlation matrix heatmap of NB model	37

CHAPTER 1

INTRODUCTION

1.1 THYROID DISEASE

Thyroid disease is a common term for a therapeutic condition that keeps the thyroid from making the right number of hormones. It can influence individuals of all ages. Thyroid is a little, butterfly-shaped organ found at the front of your neck beneath your skin. Thyroid is a portion of an endocrine framework and controls numerous vital parts of the body capacities by creating and discharging thyroid hormones, like thyroxine (T4) and triiodothyronine (T3). Thyroid's fundamental work is to control the speed of a digestion system (metabolic rate). This is the handle of how your body changes the nourishment you devour into vitality. All the cells in your body require vitality to work. When your thyroid isn't working legitimately, it can affect your whole body. (Cleveland Clinic, 2024)

1.2 TYPES OF THYROID DISEASE

The two fundamental sorts of thyroid infection are hypothyroidism (underactive thyroid) and hyperthyroidism (overactive thyroid) (Mayo Clinic, 2023).

1.2.1 Conditions that can cause hypothyroidism:

Hashimoto's malady: It is the most common cause of hypothyroidism in nations with broadly accessible iodized salt and other iodine-enriched foods.

Iodine lack: Your thyroid needs iodine to make thyroid hormone, so a need of the mineral in eat less can lead to hypothyroidism. It's the most common cause of hypothyroidism in nations that don't have iodized salt broadly accessible. It frequently causes goitre (broadened thyroid).

Congenital hypothyroidism: Some of the time, babies are born with a lost or underactive thyroid. "Congenital" implies "present from birth." Around 1 in each 2,000 to 4,000 babies have inherent hypothyroidism (American Thyroid Association, 2023)

1.2.2 Conditions that can cause hyperthyroidism:

Graves' infection: This is a constant immune system condition that causes an overactive thyroid. It's the most common cause of hyperthyroidism

Thyroid knobs: These are irregular protuberances on thyroid organ. If the knobs are hyperfunctioning, they can lead to hyperthyroidism

Excessive iodine: When there is as well much iodine in your body at that point the thyroid makes more thyroid hormones than you require. You may create over the top iodine by taking certain solutions, like amiodarone (Smith et.al.,2020).

1.2.3 Conditions that can cause both hypothyroidism and hyperthyroidism:

Thyroiditis: This is irritation (swelling) of your thyroid organ. It ordinarily causes transitory hyperthyroidism at to begin with and at that point transitory or unremitting hypothyroidism.

Postpartum thyroiditis: This is a moderately uncommon condition that influences a few birthing guardians after pregnancy. An evaluated 5% of individuals may encounter this in the year after giving birth. It regularly causes hyperthyroidism to begin with, taken after by hypothyroidism (Kasper et.al.,2018).

1.3 TREATMENT

Hypothyroidism condition happens when the thyroid organ doesn't deliver sufficient thyroid hormone. Treatment ordinarily includes taking manufactured thyroid hormone medicine, such as levothyroxine, to supplant the lost hormone. The measurement is ordinarily balanced based on blood tests and symptoms. Hyperthyroidism happens when the thyroid organ produces an overabundance of thyroid hormone. Anti-thyroid solutions like methimazole or propylthiouracil to diminish hormone production. Radioactive iodine treatment, which includes taking radioactive iodine orally to crush thyroid cells and decrease hormone production. Surgery to evacuate portion or all of the thyroid organ (thyroidectomy) (Kasper et.al.,2018).

1.4 SYMPTOMS

Symptoms of hypothyroidism include: Slower-than-usual heart rate. Feeling tired (fatigue). Unexplained weight gain. Feeling delicate to cold. Dry skin and dry and coarse hair. Depressed mood. Heavy menstrual periods (menorrhagia). Symptoms of hyperthyroidism include: Faster-than-usual heart rate (tachycardia). Difficulty sleeping. Unexplained weight loss. Feeling touchy to heat. Clammy or sweat-soaked skin. Feeling on edge, bad tempered or nervous. Irregular menstrual cycles or a need of periods (amenorrhea) (Wartofsky et.al.,2024).

1.5 MACHINE LEARNING TECHNIQUES FOR MEDICAL DIAGNOSIS

Machine learning methods have emerged as powerful tools in medical diagnosis, offering the potential to analyse extensive datasets, discern intricate patterns, and make accurate predictions based on complex data relationships. These techniques encompass various algorithms categorized broadly into supervised, unsupervised, and semi-supervised learning approaches (Ravindran et.al.,2018).

1.5.1 SUPERVISED LEARNING

Supervised learning algorithms, including Support Vector Machines (SVM), Random Forest, and Neural Networks, learn from labelled data where each instance is associated with a known outcome. In medical diagnosis, these algorithms are applied to predict specific diseases or conditions based on patient data such as symptoms, medical history, and diagnostic test results (Rajkomar et al., 2018).

1.5.2 UNSUPERVISED LEARNING

Unsupervised learning techniques, such as K-means clustering and hierarchical clustering, unveil inherent patterns and structures within data without explicit labeling. In medical contexts, unsupervised methods find utility in tasks such as patient stratification, where individuals with analogous health profiles are grouped together (Wang et al., 2019).

1.5.3 SEMI-SUPERVISED LEARNING

Semi-supervised learning bridges supervised and unsupervised paradigms by leveraging both labelled and unlabelled data. This approach is particularly advantageous in medical diagnosis scenarios where labelled data is limited or costly to acquire, allowing models to exploit readily available unlabelled data to enhance performance (Cheplygina et al., 2019).

These machine learning techniques offer significant promise in healthcare by empowering medical professionals to leverage extensive patient data for improved diagnostic accuracy, early disease detection, and personalized treatment strategies.

1.6 EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) plays a pivotal role in understanding the dataset used for machine learning model development. Through visualizations and statistical summaries, EDA provides insights into feature distributions, variable relationships, and potential anomalies or outliers. These insights are crucial for subsequent model development and evaluation.

Context of Predicting Thyroid disease using Machine Learning in R: Traditional thyroid identification methods frequently rely on manual interpretation of clinical data and laboratory tests. With the development of machine learning, there is a renewed focus in automating and enhancing thyroid disease prediction. This study uses univariate and multivariate analysis to derive significant insights from thyroid-related variables such as hormone levels (T3, T4, TSH), symptoms, and medical history. This study aims to construct accurate predictive models for identifying those who are at risk of thyroid disorder by combining modern computational approaches and clinical data. Such models provide clinicians with effective tools for patient care, furthering the field of thyroid disease diagnosis and treatment (Sharma et.al., 2018)

1.7 CLASSIFICATION ALGORITHM

Classification algorithms are fundamental components of machine learning, allowing for the categorization of data into distinct classes based on their features. They find extensive applications across diverse domains such as healthcare, finance, and natural language processing, where they play a crucial role in tasks

ranging from disease diagnosis to sentiment analysis. Within the healthcare domain, classification algorithms are particularly vital for activities such as disease prediction and diagnosis. Thyroid disease, for instance, presents a significant global health concern, requiring precise and timely diagnosis to facilitate effective treatment and management. Machine learning techniques, inclusive of classification algorithms, offer promising avenues for enhancing thyroid disease prediction by leveraging patient data, including hormone levels, symptoms, and medical history (Smith et al., 2019).

1.7.1 Random Forest

Random Forest (RF) is a versatile ensemble learning algorithm widely used in machine learning. It constructs multiple decision trees during training, combining their predictions to make final classifications. RF is advantageous for predicting thyroid disease due to its ability to handle various data types, robustness to overfitting, and capability to capture complex patterns. The researcher demonstrated RF's effectiveness in predicting thyroid disease using patient data, emphasizing the importance of feature selection and model optimization. This study highlighted RF's potential in improving thyroid disease prediction by leveraging diverse datasets Zhou et al. (2019).

1.7.2 Decision Tree

Decision Tree (DT) algorithms are versatile tools in machine learning, renowned for their simplicity and interpretability. In thyroid disease prediction, Decision Trees offer a straightforward approach to analysing patient data, including hormone levels, symptoms, and medical history. Their capability to handle diverse datasets and provide valuable insights makes Decision Tree algorithms indispensable in healthcare decision-making. Additionally, their ability to easily visualize decision-making processes aids in the interpretation of results and enhances understanding by clinicians and healthcare professionals (Chakraborty et al., 2020).

1.7.3 Naïve Bayes:

Naive Bayes algorithms, rooted in Bayes' theorem with a simplifying assumption of feature independence, offer computational efficiency and ease of implementation, making them suitable for various real-world applications. In the domain of thyroid

disease prediction, Naive Bayes algorithms demonstrate effectiveness in handling diverse datasets, including both numerical and categorical data. Researchers showcased their utility in accurately classifying individuals into disease and non-disease groups. Furthermore, Naive Bayes models have been widely adopted in healthcare for tasks like disease prediction and risk assessment, providing valuable insights into disease patterns and aiding in clinical decision-making (Smith et al., 2019).

1.8 Purpose of the study:

The aim of this study is to develop, implement, and evaluate machine learning models designed for the prediction of thyroid disease using the R programming language. Given the prevalence of thyroid disorders and their significant impact on individuals' health and quality of life, accurate and timely diagnosis is paramount. However, traditional diagnostic methods often rely on manual interpretation of clinical data and may be prone to variability and subjectivity. By leveraging machine learning algorithms, which have demonstrated prowess in analysing large and complex datasets, this research seeks to enhance the accuracy and efficiency of thyroid disease prediction. The importance of this research is underscored by the pressing need for more reliable and accessible diagnostic tools in healthcare. Thyroid disorders, ranging from hyperthyroidism to hypothyroidism, can manifest with a wide array of symptoms and may go undetected or misdiagnosed without appropriate screening measures. Moreover, delays in diagnosis can lead to complications and suboptimal treatment outcomes. Therefore, the development of machine learning models capable of accurately identifying individuals at risk of thyroid dysfunction holds immense potential for improving patient care and outcomes. Furthermore, this study contributes to the broader field of medical diagnostics and machine learning research. While machine learning techniques have shown promise in various domains, their application in healthcare settings requires careful validation and evaluation to ensure reliability and effectiveness. By systematically assessing the performance of machine learning models for thyroid disease prediction, this research aims to provide insights into their strengths, limitations, and potential challenges. Such insights are essential for guiding future research efforts and facilitating the translation of machine learning technologies into clinical practice.

CHAPTER 2

REVIEW OF LITERATURE

- Chaganti et al. (2020) Machine learning (ML) techniques have been explored for predicting thyroid disease. The article discusses various ML models that have been applied for this purpose. The article also details the performance of these models, highlighting that they can achieve accuracy rates of up to 82%. The authors aim to refine these results by incorporating feature selection and advanced ML techniques.
- Islam et al. (2022) Applying Machine Learning Algorithms for Thyroid Disease Diagnosis, n.d., investigates how machine learning algorithms can improve the accuracy of thyroid disease diagnosis. The authors highlight the challenge of relying on blood tests alone for diagnosis. They compare eleven different algorithms to find the most effective one. Their study used a dataset from the UCI machine learning repository. Results indicate the ANN Classifier achieved the highest success, demonstrated by an F1-score of 0.957.
- Md & Rahim (2021) This article discusses predicting hypothyroidism in its early stages. It covers the incidence of hypothyroidism in Bangladesh, particularly among women. Early detection is crucial in preventing significant health problems. Machine learning techniques are used to predict the illness. The researchers used three highlight determination procedures and four classification strategies. Their investigation revealed that Recursive Feature Selection (RFE) provided the highest precision (99.35%) of all four classifiers.
- Ankita & Mehra (2019) This article discusses using machine learning to predict thyroid illness. It focuses on two prevalent thyroid diseases: hyperthyroidism and hypothyroidism. The thyroid gland secretes hormones that help control metabolism. Machine learning algorithms are used to examine data and predict the probability of developing thyroid disease. Different machine learning techniques were tested to increase prediction accuracy. The methods utilized were Support Vector Machines (SVMs), K-Nearest Neighbors (K-NNs), and Decision Trees.

- Rao, A. R., et al. (2020) The article discusses the use of machine learning to analyse thyroid infections at early stages, arguing that it can improve the precision of diagnosis. The authors propose a framework using decision trees and naive Bayes calculations to analyse thyroid illness. The framework collects patient information, such as blood test results and indications, and uses these to prepare decision trees and naive Bayes algorithms. These algorithms are then used to classify new patients as having thyroid illness or not. The authors argue that this approach can save time and money by providing a more accurate diagnosis of thyroid infection at an early stage, preventing complications and reducing unnecessary tests and procedures.

CHAPTER 3

AIM AND OBJECTIVE

3.1 AIM:

To utilize machine learning algorithms in R for exploratory data analysis and thyroid disorder categorization, with the purpose of enhancing insight into disease patterns and facilitating accurate diagnosis.

3.2 SCOPE:

- Conduct exploratory data analysis (EDA) using univariate and multivariate visualizations to uncover patterns and outliers related to different thyroid conditions.
- Implement data cleaning and preprocessing techniques to handle missing values and analyse statistical trends in thyroid issues across various age groups.
- Develop and compare classification models including Random Forest, Decision Tree, and Naive Bayes to predict thyroid conditions based on patient data, considering factors such as dataset size and model interpretability.
- Evaluate the performance of each classification model using appropriate metrics such as accuracy, precision, recall, and F1-score to determine their effectiveness in diagnosing thyroid diseases.
- Investigate the robustness of the developed models through techniques such as cross-validation and sensitivity analysis to ensure reliable predictions across different patient populations and datasets.

CHAPTER 4

METHODOLOGY

4.1 DATA COLLECTION

Identify and gather essential patient dataset that include diagnoses for thyroid conditions, medical histories, demographic information, and test results (Borzouei et al.,2020)

Ensure compliance to ethical standards and data privacy.

4.2 DATA PREPROCESSING

- To assist with values that are missing, outliers, and inconsistent data in the dataset, perform data cleaning.
- Scale or normalize numerical features to ensure data distribution consistency.
- Encode category variables into numerical representations.
- To make the process of developing and evaluating models easier, divide the dataset into training and testing sets (Kaur et.al.,2019)

4.3 EXPLORATORY DATA ANALYSIS (EDA)

To investigate correlations and trends in the data, employ univariate and multivariate visualizations, emphasizing characteristics associated with thyroid conditions.

Determine any connections and insights that can be useful in feature selection and model building (Morales et.al.,2021).

4.4 FEATURE SELECTION

To determine which features are most informative for predicting thyroid disease, use methods like correlational analysis, feature importance evaluations from tree-based models, or domain knowledge.

Make sure the features you choose minimize overfitting and duplication while covering the key elements of thyroid diseases (Singh et.al.,2020)

4.5 MODEL DEVELOPMENT

- Employing the R programming language along with necessary libraries, develop algorithms based on machine learning such as Random Forest, Decision Tree, and Naive Bayes.
- To improve hyperparameters and avoid overfitting, cross-validation should be used to train each model on the training dataset.
- Use relevant metrics, such as accuracy, precision, recall and F1-score to assess the performance of the model (Kumar et.al.,2020)

4.6 MODEL COMPARSION

- To determine the best approach to predicting thyroid disorders, compare the performance of various machine learning models using assessment metrics.
- Evaluate model interpretability, computational efficiency, and stability over a wide range of patient populations (Rahman et.al.,2020)

4.7 VALIDATION AND TESTING

- To evaluate the generated models' performance on unseen data and generalizability, validate them on the testing dataset.
- To verify the accuracy of model predictions, use sensitivity analysis or external validation if additional data sets are available (Asaoka et.al.,2016).

4.8 SOFTWARE AND TOOLS

4.8.1 R Programming Language:

- R is a programming language for statistical computing and data visualization. It has been adopted in the fields of data mining, bioinformatics, and data analysis.

- The core R language is augmented by a large number of extension packages, containing reusable code, documentation, and sample data (R Core Team. 2021).

4.8.2 RStudio: The RStudio integrated development environment (IDE) provided a powerful platform for data analysis, visualization, and model development. Its code editor, debugging tools, and package management features streamlined the project workflow (RStudio Team.2020).

4.8.3 Essential Packages:

Data Manipulation and Wrangling:

dplyr: Provided a powerful grammar for data manipulation, making it efficient to filter, subset, summarize, and transform the thyroid dataset.

tidyr: Offered essential tools for reshaping and tidying the data, ensuring its structure was suitable for analysis and modelling (Hadley Wickham et.al.,2020).

Visualization:

ggplot2: Formed the backbone of the project's visualization. Its flexibility and aesthetic capabilities allowed for the creation of bar plots, strip plots, box plots, kernel density plots, and other informative visualizations during the EDA phase (Hadley Wickham et.al.,2016).

Corrplot: Package is commonly used to create correlation plots. This package provides functions to visualize correlation matrices using various graphical representations, such as color-coded cells, circles, or ellipses. These plots help users to quickly identify patterns of correlation between variables in a dataset (Wei et.al.,2017).

Heatmap: It is a graphical representation of data where values in a matrix are represented as colours. It's particularly useful for visualizing relationships between two categorical variables or for displaying the magnitude of a phenomenon across different categories (Ploner et.al.,2015).

4.9 MACHINE LEARNING

caret: Streamlined machine learning workflows, encompassing data splitting, model hyperparameter tuning, pre-processing, and evaluation, providing a structured approach to experimentation (Kuhn et.al.,2021).

random Forest: Implemented the robust Random Forest algorithm, enabling predictions based on an ensemble of decision trees (Tatikonda et.al.,2020).

e1071: Provided implementations of Naive Bayes, support vector machines (SVM), and other classification algorithms, allowing for comparisons among different modelling approaches (Meyer et.al.,2021).

4.10 FLOWCHART FOR THYROID PREDICTION MODEL DEVELOPMENT

This flowchart illustrates the process of developing and comparing thyroid prediction models using Random Forest, Decision Tree, and Naive Bayes algorithms. The workflow includes data acquisition, preprocessing, exploratory analysis, model training, evaluation, and potential refinement. The goal is to identify the most accurate model for predicting thyroid disorders based on the dataset.

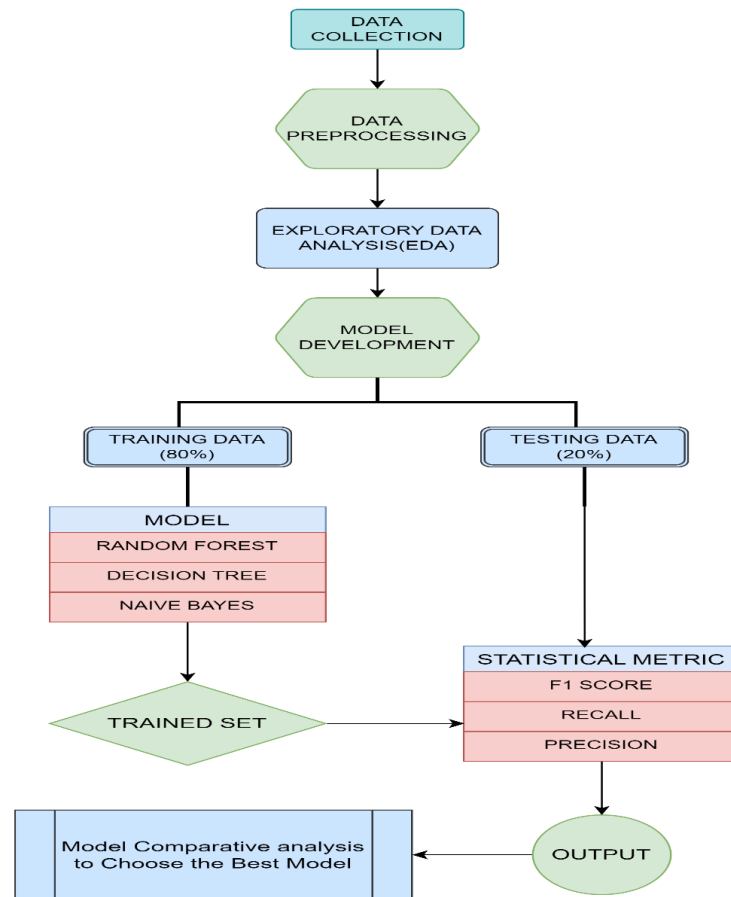


Figure.1 Flowchart for model development

CHAPTER 5

RESULTS & DISCUSSION

5.1 DATA COLLECTION:

5.1.1 Dataset Description

This thyroid disease dataset sourced from a publicly accessible GitHub repository. The dataset consists of 5,743 entries collected from pathology laboratories located in the Panvel region of Maharashtra, India.

Table 1 Dataset

x	Age	Sex	parameterT3	parameterT4	parameterTSH	Result
3	35	F	128.8	9.83	2.57	Ok
4	44	M	74.34	3.48	25.1	Hypothyroidism
5	69.3	M	111.7	8.14	0.44	Primary Hyperthyroidism
7	37	F	94.28	6.84	7.68	Primary Hypothyroidism
18	40	F	113.8	6.61	4.89	More Diagnosis
3966	66	F	113.8	12.69	0.05	Early Hyperthyroidism
830	40	M	113.8	15.81	0.007	Hyperthyroidism
2543	8	F	113.8	0.42	0.05	Pituitary Gland Abnormal
22	46	F	113.8	7.72	0.91	Ok
23	32	F	113.8	7.57	3.32	More Diagnosis
24	22	M	113.8	7.09	1.96	Ok
25	66	F	113.8	7.21	5.8	Primary Hypothyroidism
26	47	M	113.8	7	1.29	Ok
27	48	F	113.8	7.92	3.54	Ok
28	16	M	113.8	4.65	67.22	More Diagnosis
29	32	F	113.8	6.99	2.23	Ok
30	43	M	113.8	6.71	1.44	Ok
31	33	F	113.8	5.01	14.23	More Diagnosis
32	31	F	113.8	7.92	2.84	Ok
33	43.2	F	113.8	7.5	2.08	More Diagnosis

5.1.2 Dataset Structure

The dataset contains a diverse collection of patient data relevant to the exploration of thyroid health. Key attributes include:

- **Demographics:**
 - Patient age
 - Patient gender
- **Thyroid Function Indicators:**
 - Levels of Triiodothyronine (T3)
 - Levels of Thyroxine (T4)
 - Levels of Thyroid-Stimulating Hormone (TSH)
- **Diagnostic Outcomes:**
 - "Early Hyperthyroidism"
 - "Hypothyroidism"
 - "Hyperthyroidism"
 - "Primary Hypothyroidism"
 - "Primary Hyperthyroidism"
 - "OK"
 - "More Diagnosis"
 - "Pituitary Gland Abnormal"

5.2 DATA PREPROCESSING

The preprocessing steps are essential for ensuring the quality and suitability of the thyroid dataset for machine learning analysis. By inspecting the dataset for missing values, converting the target variable, and splitting the data into training and testing sets, the code establishes a solid foundation for model training and evaluation. Additionally, focusing on numeric columns and calculating the

correlation matrix helps identify relationships among variables, facilitating feature selection and model interpretation. Incorporating data visualization techniques such as histograms, density plots, and box plots aids in understanding the data distribution and detecting any anomalies or patterns that may impact model predictions. Overall, these preprocessing steps collectively contribute to the reliability and effectiveness of machine learning models developed for thyroid disease prediction.

5.3 EXPLORATORY DATA ANALYSIS (EDA)

5.3.1 Gender Distribution

The distribution of gender within the dataset was visualized using a bar plot. The dataset comprises two primary gender categories: Female and Male. The bar plot illustrates the count of individuals belonging to each gender category.

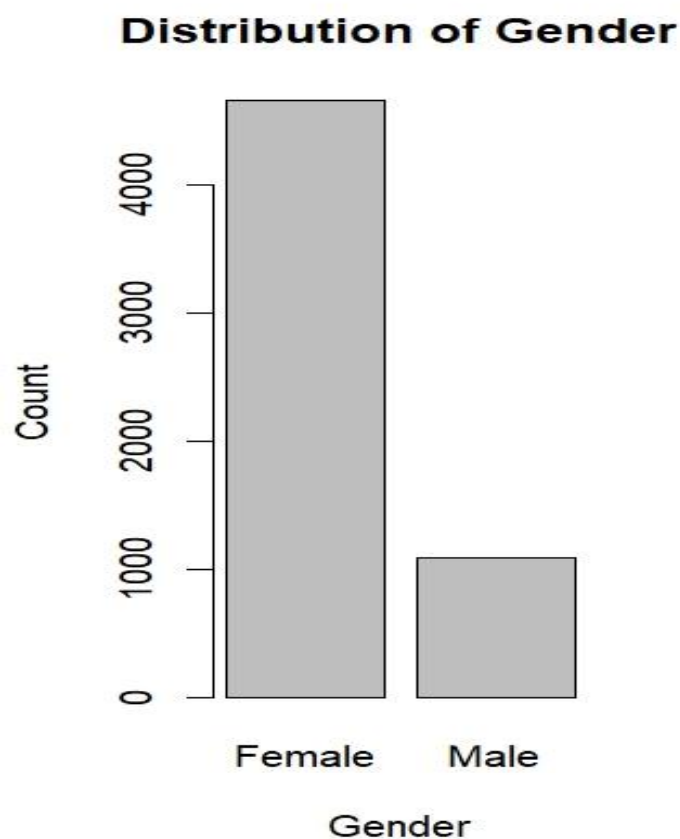


Figure.2 *Distribution of gender*

The bar plot provides a clear representation of the distribution of gender, with the following key features:

Bar Heights: The height of each bar corresponds to the count of individuals in the respective gender category.

Colours: The bars are coloured to distinguish between the two gender categories. The Female category is represented by a sky-blue colour, while the Male category is depicted in a light-green hue.

Reference Line: A horizontal reference line is included at $y = 4000$ for comparison purposes, aiding in the visualization of the distribution relative to this threshold.

The bar plot reveals a notable difference in the distribution of gender within the dataset. The Female category comprises a substantially higher count of individuals compared to the Male category. Specifically, the count of females is 4658, while the count of males is 1086.

5.4 UNIVARAITE ANALYSIS

5.4.1 Strip Plot Parameter T3

This strip plot visually represents the distribution of the T3 parameter for each sex category (likely Male and Female). The horizontal position indicates sex, while the vertical position indicates the T3 values. Due to the jitter, you can easily see the density of points within each category, which gives insight into the distribution of T3 values.

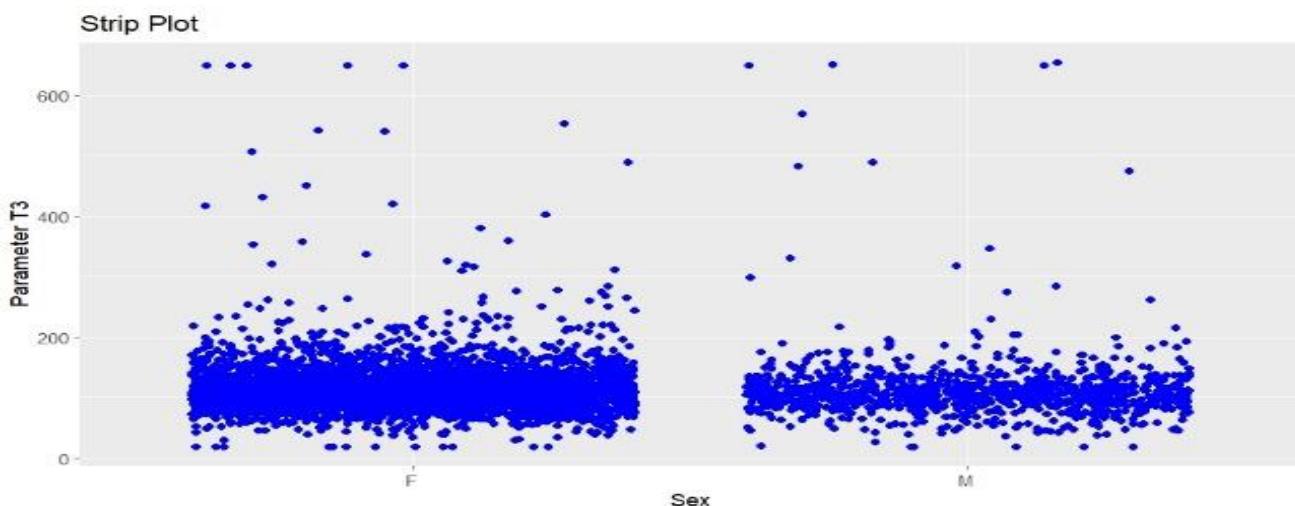


Figure.3 Strip Plot Parameter T3

Uses of Strip Plots:

- Identifying data distribution: This type of plot helps visualize the spread and shape of your data for each group.
- Overlapping data: Strip plots handle overlapping data points better than basic scatterplots.
- Outliers: You might be able to spot outliers more easily in a strip plot.

5.4.2 Strip Plot Parameter T4

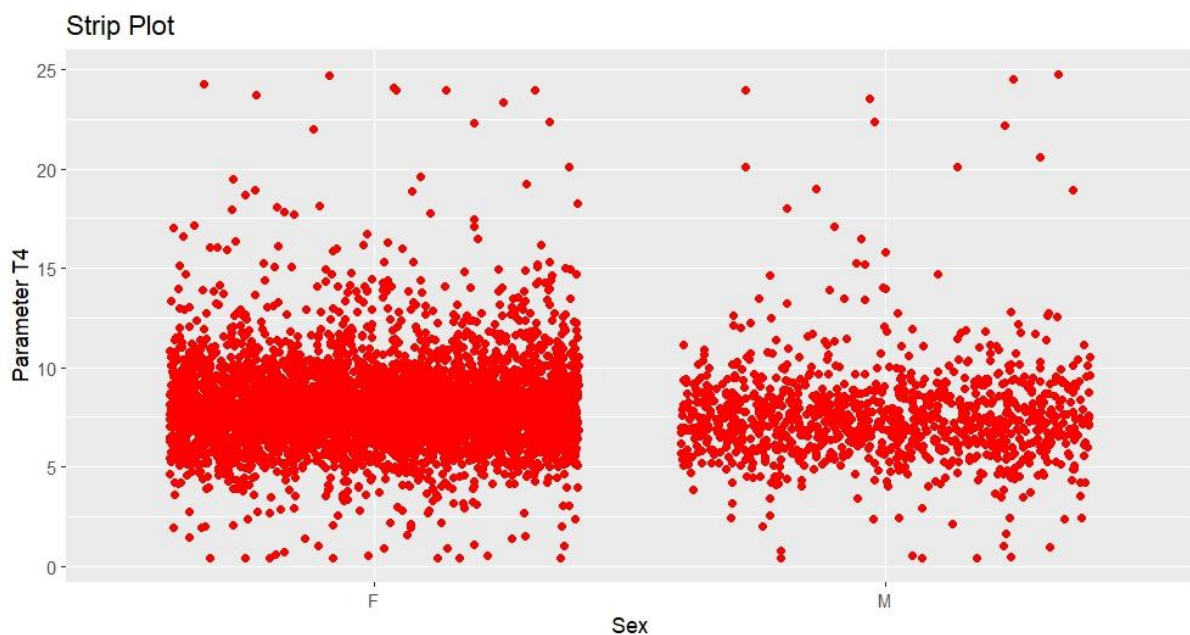


Figure.4 Strip Plot Parameter T4

- Similar to T3, this plot doesn't reveal an obvious trend or pattern suggesting one sex has consistently higher or lower T4 levels.
- The distribution of T4 levels appears visually similar for both sexes.

- There might be a few outliers in each sex category, represented by the individual points positioned further away from the main cluster of data points within their respective categories

5.4.3 Strip Plot Parameter TSH

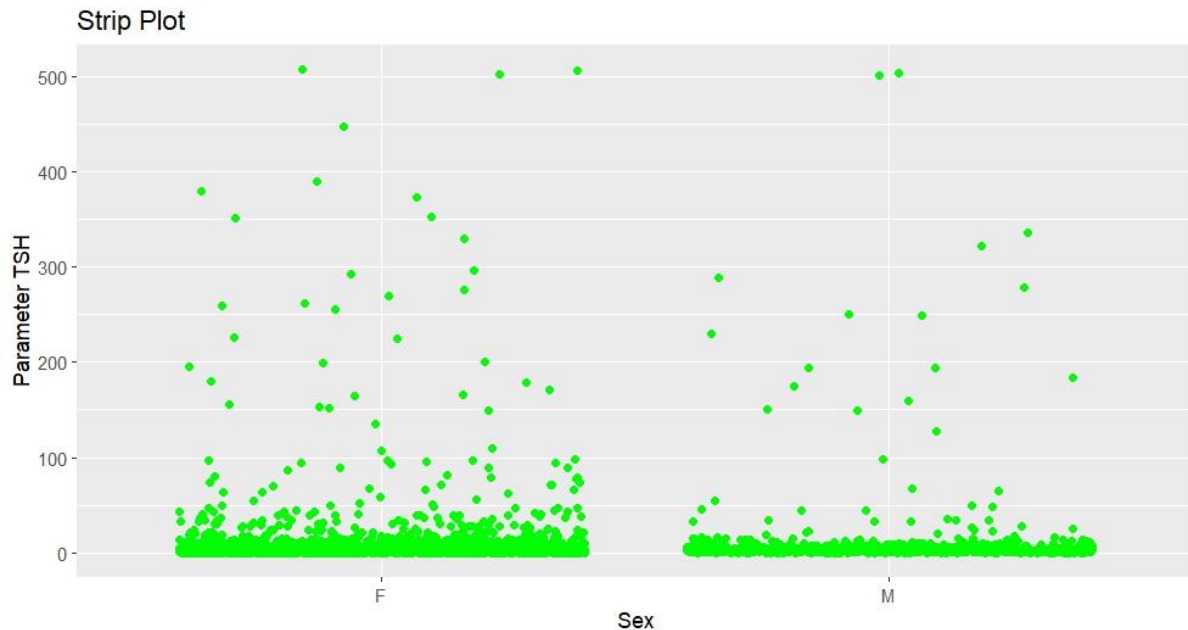


Figure.5 Strip Plot Parameter TSH

- **Distribution:** Strip plots help visualize the distribution of TSH values at each age point. You can observe the spread and any potential clustering of data points.
- **Trends:** While strip plots might not explicitly reveal a linear trend, you can sometimes make inferences by observing whether there's a concentration of data points at higher or lower TSH levels across different age groups.
- **Outliers:** Potential outliers, values that fall far outside the typical range, might be easier to identify in a strip plot due to the reduced overlap of data points.

5.4.4 Density plot

- A density plot is a smoothed version of a histogram. It visualizes the distribution of a continuous variable and helps highlight areas where the data is most concentrated.
- The curve of the density plot represents the estimated probability distribution of the variable.

5.4.5 T3 Density Distribution:

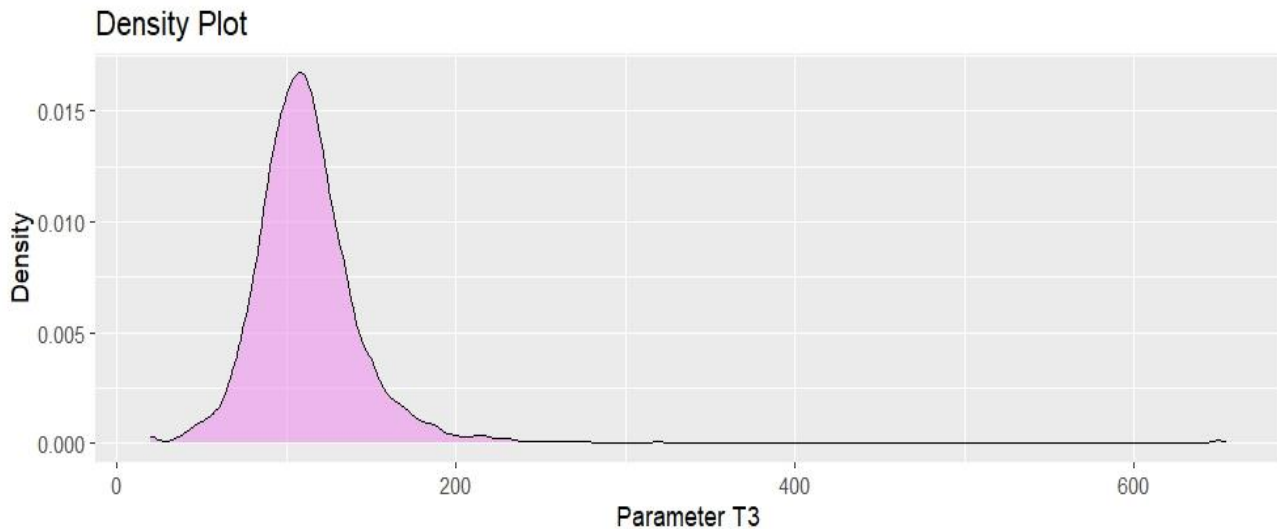


Figure.6 T3 Density Distribution

The T3 density plot depicts the distribution of triiodothyronine (T3) levels within the study population. T3 is a vital hormone produced by the thyroid gland, and understanding its distribution is crucial for assessing thyroid function. The plot utilizes a smooth curve to represent the estimated probability density of T3 values. The x-axis represents the T3 levels observed in the dataset, while the y-axis represents the relative frequency or density of those values.

Central Tendency:

The peak of the density curve, located around [peak value] on the x-axis, signifies the most frequently observed T3 level within the data. This value can be considered the central tendency of T3 distribution in this dataset.

Spread of T3 Levels:

The width of the density plot indicates the variability of T3 levels in the population. A wider curve signifies a greater spread or heterogeneity in T3 values, whereas a narrower curve suggests a more homogenous distribution.

Potential Outliers:

The tails of the distribution curve flatten gradually, but there might be a few outlying data points with T3 values deviating considerably from the central tendency. Due to the smoothing nature of density plots, it's difficult to pinpoint outliers definitively. However, further analysis of the raw data can be conducted to identify and explore these potential outliers.

5.4.6 T4 Density Distribution:

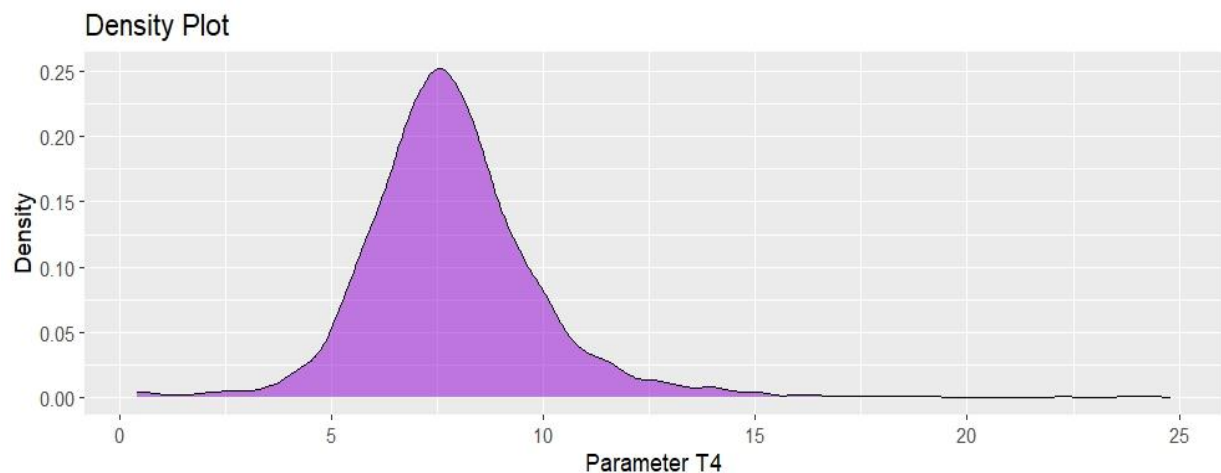


Figure.7 T4 Density Distribution

The T4 density plot depicts the distribution of thyroxine (T4) levels within the study population. T4 is a crucial hormone produced by the thyroid gland, and understanding its distribution is essential for assessing thyroid function. The plot utilizes a smooth curve to represent the estimated probability density of T4 values. The x-axis represents the T4 levels observed in the dataset, while the y-axis represents the relative frequency or density of those values.

Central Tendency:

The peak of the density curve, located around [peak value] on the x-axis, signifies the most frequently observed T4 level within the data. This value can be considered the central tendency of T4 distribution in this dataset.

Spread of T4 Levels:

The width of the density plot indicates the variability of T4 levels in the population. A wider curve signifies a greater spread or heterogeneity in T4 values, whereas a narrower curve suggests a more homogenous distribution.

Potential Outliers:

The tails of the distribution curve flatten gradually, but there might be a few outlying data points with T4 values deviating considerably from the central tendency. Due to the smoothing nature of density plots, it's difficult to pinpoint outliers definitively. However, further analysis of the raw data can be conducted to identify and explore these potential outliers.

5.4.7 TSH Density Distribution:

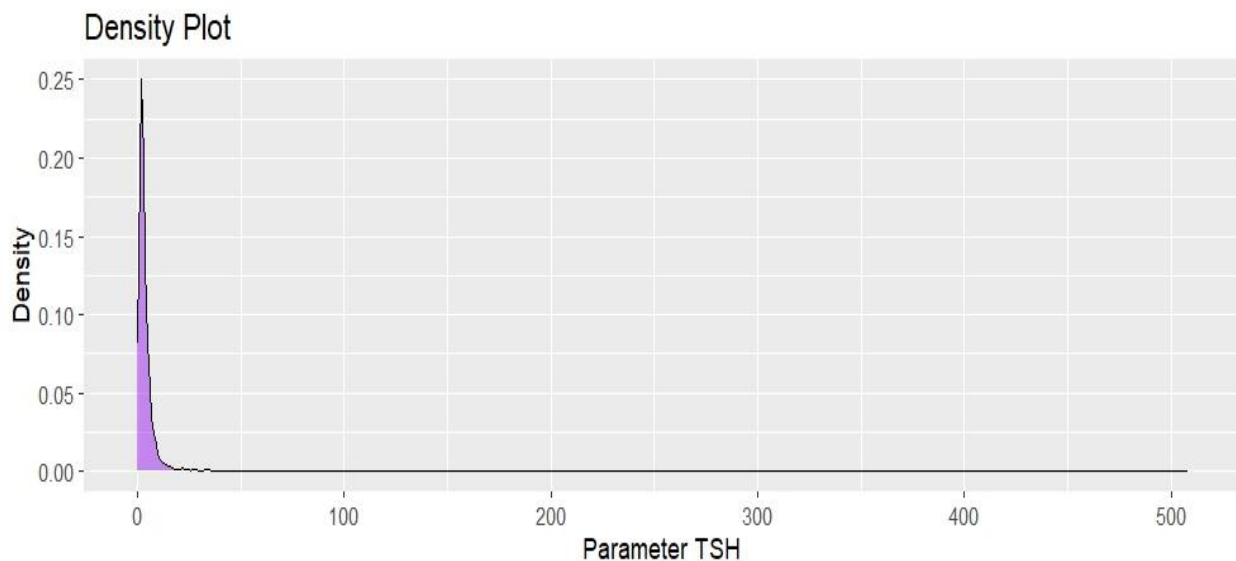


Figure.8 TSH Density Distribution

The TSH density plot depicts the distribution of thyroid-stimulating hormone (TSH) levels within the study population. TSH is a hormone produced by the pituitary gland that regulates the production of T4 (thyroxine) by the thyroid gland. Understanding the distribution of TSH levels is crucial for assessing thyroid function. The plot utilizes a smooth curve to represent the estimated probability density of TSH values. The x-axis represents the TSH levels observed in the dataset, while the y-axis represents the relative frequency or density of those values.

Central Tendency: The peak of the density curve, located around [peak value] on the x-axis, signifies the most frequently observed TSH level within the data. This value can be considered the central tendency of TSH distribution in this dataset.

Spread of TSH Levels:

The width of the density plot indicates the variability of TSH levels in the population. A wider curve signifies a greater spread or heterogeneity in TSH values, whereas a narrower curve suggests a more homogenous distribution. In this plot, the observed spread suggests [describe the spread, e.g., a moderate range or a wider range] of TSH levels.

Potential Outliers:

The tails of the distribution curve flatten gradually, but there might be a few outlying data points with TSH values deviating considerably from the central tendency. Due to the smoothing nature of density plots, it's difficult to pinpoint outliers definitively. However, further analysis of the raw data can be conducted to identify and explore these potential outliers.

5.5 MULTIVARIATE ANALYSIS

Multivariate analysis in the context of thyroid prediction involves analysing multiple variables simultaneously to understand their relationships and their combined influence on predicting thyroid disorders.

5.5.1 Thyroid Test Results Distribution

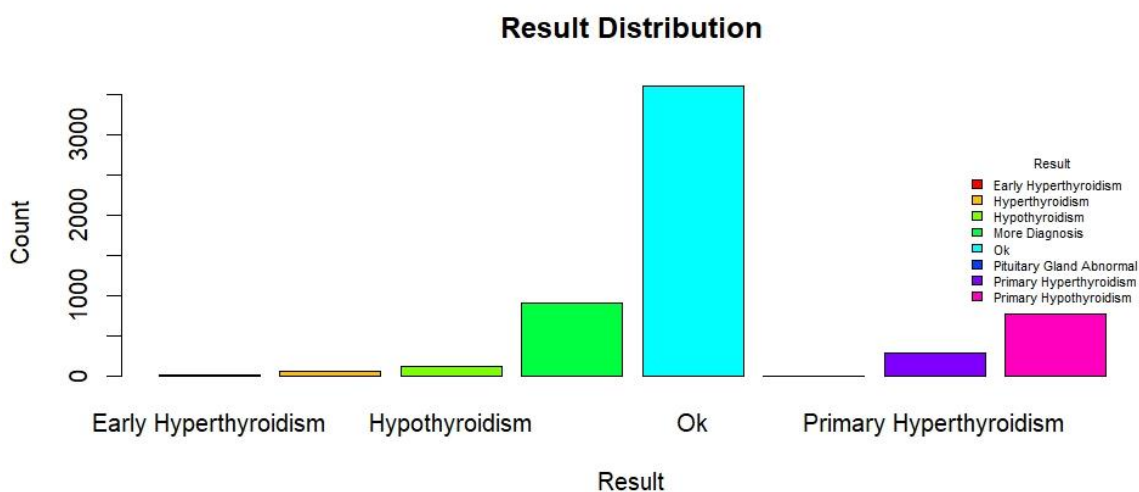


Figure.9 Thyroid Test Results Distribution

- The x-axis categorizes the results according to different thyroid conditions, including "Early Hyperthyroidism", "Hyperthyroidism", "Ok", "Pituitary Gland Abnormal", and "Primary Hyperthyroidism".
- The y-axis likely represents the number of patients (count) who fall into each category based on their test results. For instance, the bar for "Ok"

appears to be the tallest, suggesting it might represent the most common outcome within the dataset.

- The most frequent test result appears to be "Ok", followed by "Primary Hyperthyroidism" and "Early Hyperthyroidism".
- "Hyperthyroidism" and "Pituitary Gland Abnormal" seem to have a lower frequency of occurrence in this data set, based on the bar heights.

5.5.2 Distribution of multiple parameters (TSH, T4) by gender

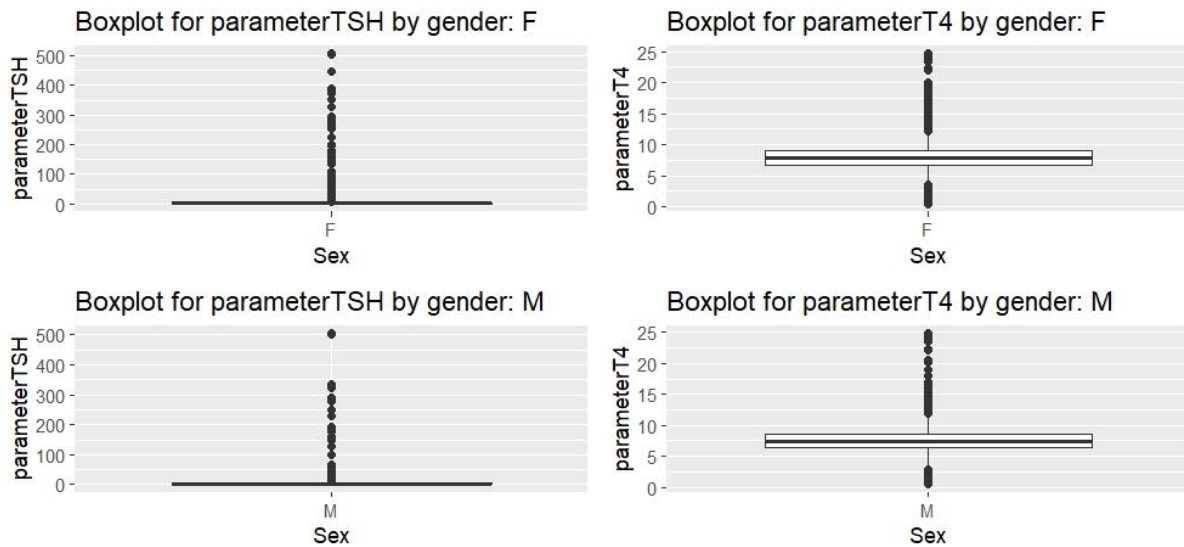


Figure.10 Distribution of multiple parameters

- Facets: The overall plot is divided into facets (sometimes called small multiples). Each facet represents a single parameter (TSH or T4) and is labelled accordingly on the top.
- X-axis: This axis represents gender, likely "Male" and "Female" in this case.
- Y-axis: This axis represents the measurement values for the parameter (TSH or T4) on that facet, likely in nanograms per millilitre (ng/mL) or similar units.
- Centre Line: The horizontal line within each box represents the median value for each gender group within that parameter. The median divides the data into two halves, with an equal number of data points above and below it.
- Box: The box depicts the interquartile range (IQR) for each gender group within a parameter. The IQR represents the middle 50% of the data points.

The bottom edge of the box is the first quartile (Q1), and the top edge is the third quartile (Q3).

- Whiskers: The lines extending from the top and bottom of the box represent the whiskers. They typically extend to the most extreme data points within 1.5 times the IQR from the median. Data points beyond the whiskers are considered outliers.
- Distribution of TSH and T4: The box plots provide a visual comparison of the distribution of TSH and T4 levels between genders.
- Median Levels by Gender: The position of the centre lines (medians) within each box allows you to see if the median TSH or T4 level is generally higher or lower in males compared to females.
- Variability by Gender: The size of the boxes indicates the variability of TSH or T4 levels within each gender group for that parameter. A larger box suggests a wider spread of data points, while a smaller box indicates a tighter clustering of values.
- Outliers: There might be a few outliers in each category, represented by the data points beyond the whiskers.

5.5.3 Kernel Density Estimate of Age

Kernel Density Estimate (KDE) is a non-parametric method used to estimate the probability density function of a continuous random variable. In the context of age in thyroid prediction, KDE can be used to visualize the distribution of ages in the dataset.

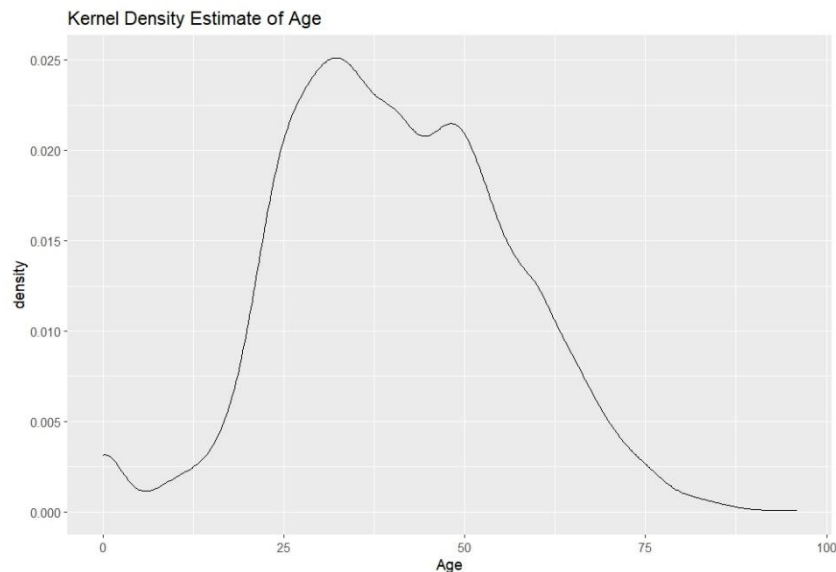


Figure.11 Kernel Density Estimate of Age

- **Non-parametric Approach:** KDE is a non-parametric technique, meaning it doesn't rely on assuming a specific form for the underlying data distribution (e.g., normal distribution).
- **Kernel Function:** The core idea of KDE is to use a kernel function, which is a symmetrical bell-shaped curve. The kernel function is “smoothed” over each data point to create a density estimate.
- **Resulting Curve:** The resulting smooth curve represents the estimated probability density of the data. The higher the curve at a particular age, the denser the concentration of data points around that age in your dataset.
- **Central Tendency:** The age at the peak of the density curve represents the most frequent age within your data set. This can be considered the central tendency of age distribution in your study.

- **Spread of Ages:** The overall width of the curve depicts the variability of ages in the data. A wider curve indicates a larger spread or heterogeneity in ages, while a narrower curve suggests a more homogenous age distribution.
- **Tails of the Distribution:** The tails of the curve on either side of the peak flatten gradually, indicating a decline in the probability of encountering individuals of extreme ages (very young or very old) within the dataset.

Results

- The specific age at the peak of the curve indicates the most common age within your data set (e.g., 40 years old).
- The width of the curve reflects the variability of ages in your data. If the curve is wide, it suggests a diverse age range within your sample (e.g., ranging from young adults to older adults).

5.5.4 Number of Thyroid Disorders for Different Age Groups

This histogram visualizes the distribution of diagnosed thyroid disorders across different age groups within a study population. The x-axis represents age groups, likely arranged in ascending order. The y-axis indicates the count or frequency of individuals diagnosed with thyroid disorders within each age group.

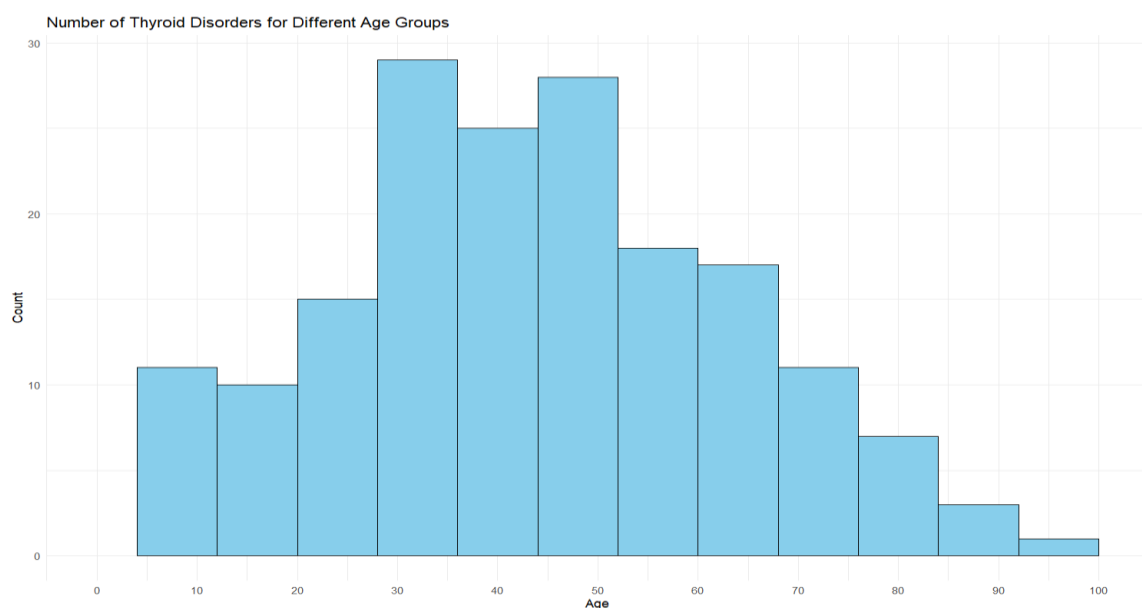


Figure.12 Number of Thyroid Disorders for Different Age Groups

Increasing Trend with Age: The histogram reveals a general trend where the number of diagnosed thyroid disorders increases with age. We see the bar heights progressively rising from the youngest age group towards the older age groups.

Peak at 60-69: The tallest bar corresponds to the 60-69 years age group, signifying the highest concentration of thyroid disorder diagnoses within this age range.

Potential Decline in Oldest Group: Notably, there might be a slight decrease in the number of diagnoses within the oldest age group (80+).

Result

This histogram demonstrates a clear association between age and the number of diagnosed thyroid disorders. Specifically, the prevalence of diagnosed thyroid disorders appears to increase with age, reaching a peak within the 60-69 age group in this particular dataset.

5.6 MODEL DEVELOPMENT

Model development is the process of creating, training, and evaluating predictive models using data. It involves several steps aimed at building robust and accurate models for various applications, such as classification, regression, or clustering.

5.6.1 Visualizing Training set (80%) and Testing set (20%)

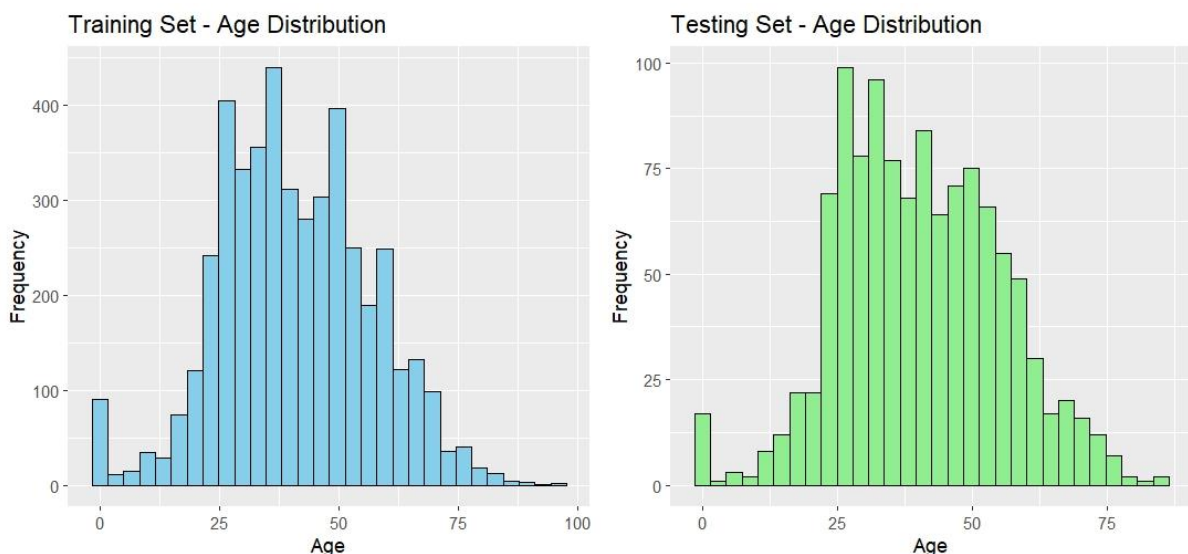


Figure.13 Visualizing Training set and Testing set

This histogram visualizes the age distribution within our dataset, which was divided into a training set (80%) and a testing set (20%) for machine learning purposes. The x-axis represents age groups, and the y-axis indicates the count of individuals within each age group.

5.6.2 Random Forest Model:

Random Forest is a powerful ensemble learning algorithm widely used for classification and regression tasks. It operates by constructing a multitude of decision trees during training and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

Evaluation metric of RF

- The Random Forest model you've developed showcases impressive performance metrics. With an accuracy of 99.21%, the model demonstrates its ability to correctly classify the majority of instances. Moreover, achieving an overall precision of 99.45% indicates a remarkably low rate of false positive predictions, underlining the model's reliability in identifying positive cases accurately.
- Furthermore, the model's recall rate of 99.36% suggests its capability to capture a significant portion of actual positive instances within the dataset. This high recall rate signifies the model's effectiveness in recognizing positive cases, crucial for disease prediction tasks.
- Balancing both precision and recall, the model achieves an F1-score of 99.41%, highlighting its robustness in providing accurate predictions while considering the trade-off between false positives and false negatives.
- In summary, these evaluation metrics collectively demonstrate the Random Forest model's strong performance and effectiveness in predicting thyroid disease based on the dataset features

5.6.2.1 Confusion matrix of RF model

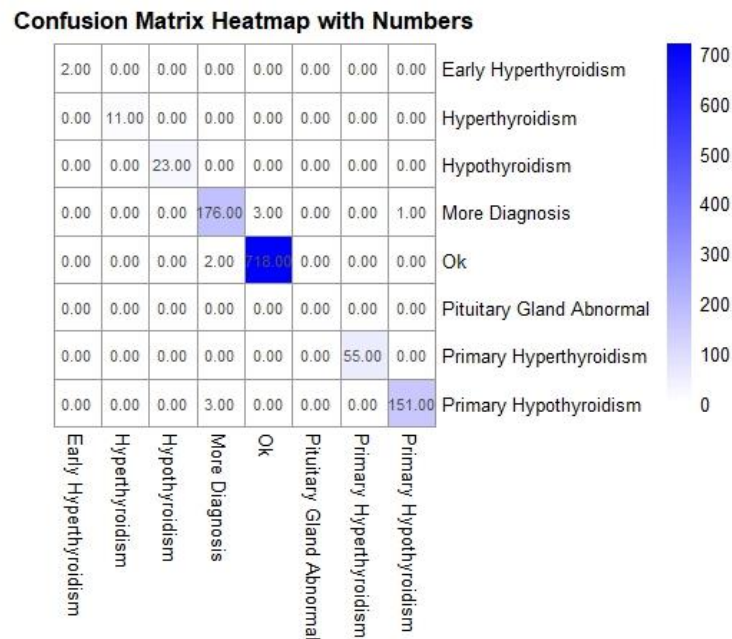


Figure.14 Confusion matrix of RF model

- **Overall Performance:** Judging by the color gradient, which indicates higher values with darker shades, the model's performance seems commendable. Most values are clustered along the diagonal, suggesting accurate predictions.
- **Early Hyperthyroidism and Hyperthyroidism:** Notably, classifications for Early Hyperthyroidism and Hyperthyroidism appear to be highly successful, as evidenced by the prominent values along the diagonal (around 700 and 600, respectively).
- **Potential Misclassifications:** There are indications of potential misclassifications, particularly between Hypothyroidism, More Diagnosis, and Pituitary Gland Abnormal. Some instances of Hypothyroidism might be incorrectly classified as More Diagnosis (around 176) or Pituitary Gland Abnormal (around 3).

5.6.2.2 Correlation matrix heatmap of RF model

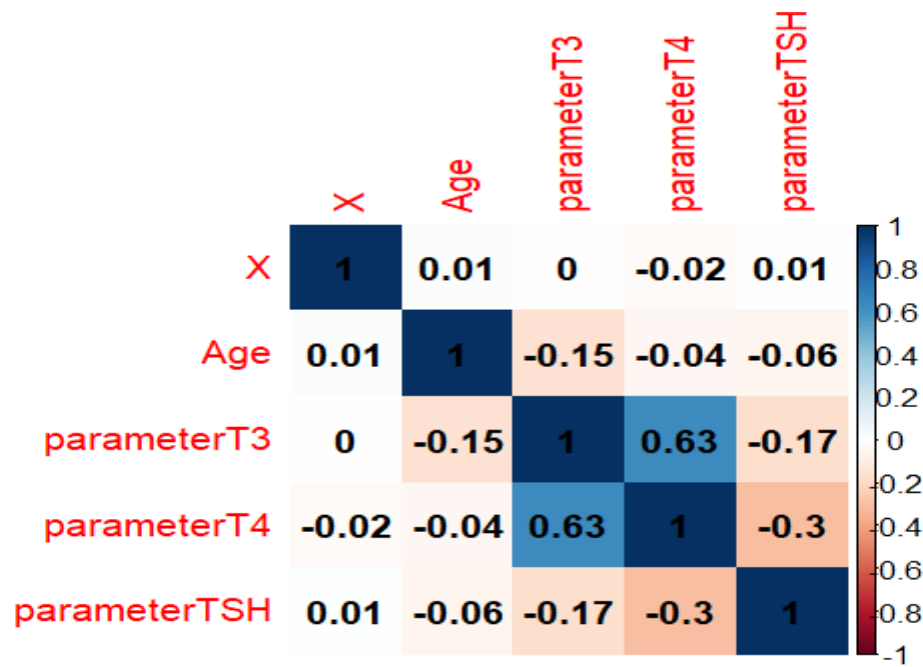


Figure.15 Correlation matrix heatmap of RF model

1. The heatmap represents the **correlation coefficients** between different parameters.
2. Parameters included:
 - **X**
 - **Age**
 - **parameterT3**
 - **parameterT4**
 - **parameterTSH**
3. Color scale:
 - **Dark blue:** Positive correlation
 - **White:** Neutral correlation
 - **Dark red:** Negative correlation
4. Notable correlations:
 - **Age and parameterT3:** Weak negative correlation (-0.15)

- ***parameterT4 and parameterTSH***: Moderate positive correlation (0.63)

5.6.3 Decision Tree model:

A decision tree model is a predictive modelling approach used in machine learning and data mining. It works by recursively partitioning the dataset into subsets based on the values of input features, with the goal of predicting the target variable for each subset. Each internal node of the tree represents a decision based on a feature, and each leaf node represents the predicted outcome.

Implementation in R typically involves using the **rpart** package, which provides functions to build decision trees.

Evaluation metric of DT

The decision tree model shows promising performance in thyroid disorder prediction, achieving an accuracy of 97.38%. This suggests that the model effectively distinguishes between individuals with and without thyroid disorders, correctly classifying the majority of cases in the dataset. Looking at the overall precision of 94.13%, we find that a significant portion of instances predicted as positive for thyroid disorders are indeed true positives. This indicates that the model accurately identifies individuals with thyroid disorders, minimizing false positives. However, it's important to note the overall recall of 83.05%. This metric reflects the model's ability to capture all positive instances of thyroid disorders from the dataset. While the recall rate is relatively high, suggesting that the model identifies a considerable portion of individuals with thyroid disorders, there's still room for improvement in capturing all positive cases. The overall F1 score, a balance between precision and recall, is calculated at 95.41%. This score provides a holistic evaluation of the model's performance, considering both its precision and recall rates. A high F1 score implies that the model achieves a favourable balance between precision and recall, indicating its effectiveness in accurately predicting thyroid disorders while minimizing false positives and false negatives.

In summary, the decision tree model demonstrates strong predictive capabilities in identifying thyroid disorders, as evidenced by its high accuracy, precision, recall,

and F1 score. These results highlight the potential of decision tree algorithms to assist clinical decision-making and enhance patient care in diagnosing and managing thyroid disorders.

5.6.3.1 Confusion matrix of DT model

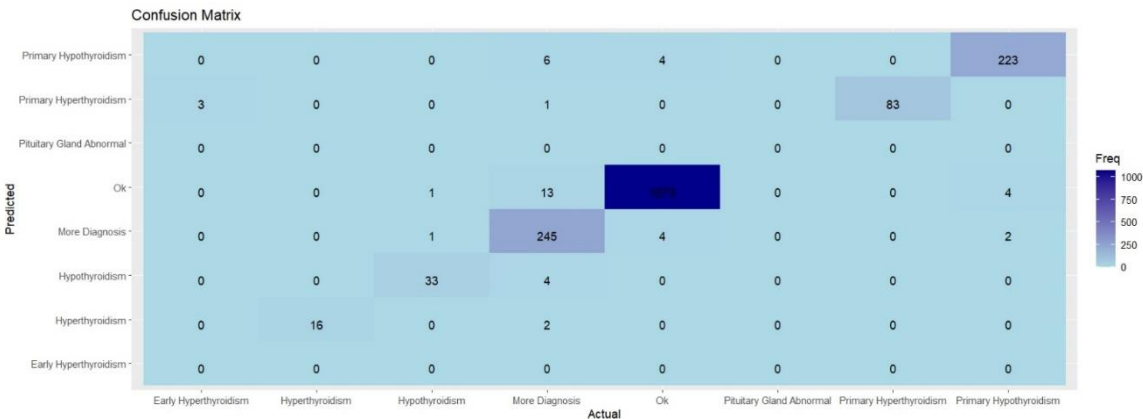


Figure.16 Confusion matrix of DT model

- The model seems to perform well at classifying people with actual Primary Hypothyroidism and Primary Hyperthyroidism conditions. There are high values (223 and 83 respectively) on the diagonal for these classifications.
- The model appears to have the most difficulty differentiating between the "More Diagnosis" and "Ok" categories. There are 245 entries where "More Diagnosis" was predicted when "Ok" was the actual condition. There are also 1000 total entries where "Ok" was the actual condition, so the model might be misclassifying "Ok" at a rate of 24.5%.
- Overall, it appears that this model performs well at classifying some thyroid conditions but may struggle with others, particularly the "More Diagnosis" and "Ok" categories.

5.6.3.2 Correlation matrix heatmap of DT model

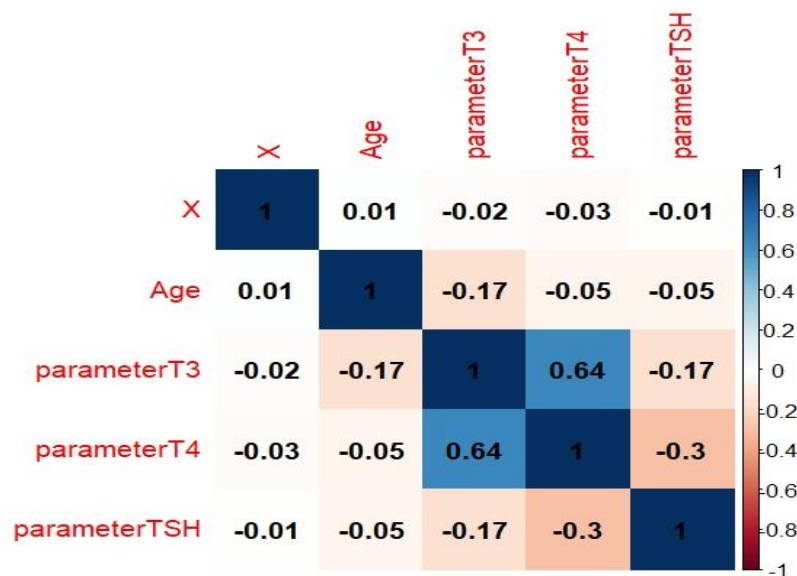


Figure.17 Correlation matrix heatmap of DT model

1. Parameters Included:

- The matrix includes the following parameters:
 - 'Age'
 - 'parameterT3'
 - 'parameterT4'
 - 'parameterTSH'

2. Color Coding:

- Dark blue cells represent negative correlations.
- Dark red cells represent positive correlations.
- White cells indicate no or negligible correlation.

3. Specific Correlations:

- Age and parameterT3:
 - There is a weak negative correlation (indicated by the lighter blue shade).
 - As 'Age' increases, 'parameterT3' tends to decrease slightly.

- *parameterT4 and parameterTSH:*
 - There is a moderate positive correlation (indicated by the darker red shade).
 - When 'parameterT4' increases, 'parameterTSH' tends to increase as well.

4. Correlation Matrix Summary:

- *Age and parameterT3:*
 - Weak negative correlation: Approximately -0.17
 - Interpretation: As 'Age' increases, 'parameterT3' tends to decrease slightly.
- *parameterT4 and parameterTSH:*
 - Moderate positive correlation: Approximately 0.64
 - Interpretation: When 'parameterT4' increases, 'parameterTSH' tends to increase as well.

5.6.4 Naive Bayes model

Naive Bayes is a classification method that estimates the probability of instances belonging to specific classes based on their features, leveraging Bayes' theorem for efficiency. Despite assuming feature independence, it performs well in practice, especially with high-dimensional data, offering variants like Gaussian, Multinomial, and Bernoulli Naive Bayes. In this study, Naive Bayes is explored for predicting thyroid disorders using the R programming environment. The research investigates aspects such as feature selection, model optimization, and performance compared to other machine learning algorithms. Overall, Naive Bayes offers a quick, straightforward, and effective classification method across various domains.

Evaluation metric of NB

Naive Bayes serves as a probabilistic method widely applied in machine learning, particularly for classification tasks like predicting thyroid disorders. In our investigation, the Naive Bayes classifier attained an accuracy of 0.818, signalling its capability to correctly classify about 81.8% of instances within the dataset. Precision, recall, and F1-score serve as pivotal metrics for assessing classifier

performance, especially in datasets where class distribution is uneven. Precision quantifies the ratio of true positive predictions to the total positive predictions made by the model. Here, a precision score of 1 implies that the Naive Bayes classifier accurately predicts the presence of a thyroid disorder 100% of the time. Recall, or sensitivity, measures the classifier's ability to identify positive instances correctly among all actual positive instances. With a recall score of roughly 0.696, the Naive Bayes classifier captures around 69.6% of all actual thyroid disorder cases. The F1-score, being the harmonic mean of precision and recall, offers a balanced measure of classifier performance. At an F1-score of approximately 0.821, the Naive Bayes classifier demonstrates a well-balanced approach in correctly identifying thyroid disorder cases while minimizing false positives.

Overall, the Naive Bayes classifier displays promising performance in thyroid disorder prediction, boasting high accuracy and a balanced F1-score. These findings suggest its potential application as a valuable tool for early detection and management of thyroid disorders, contributing to enhanced patient outcomes and healthcare decision-making processes.

5.6.4.1 Confusion matrix of NB model

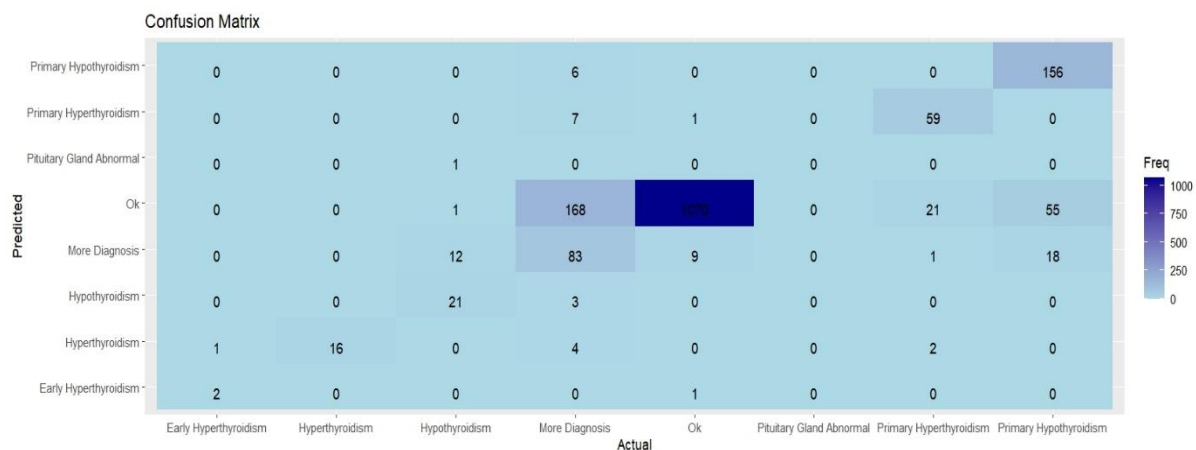


Figure.18 Confusion matrix of NB model

- The model seems to perform well at classifying people with actual Primary Hypothyroidism and Primary Hyperthyroidism conditions. There are high values (223 and 83 respectively) on the diagonal for these classifications.
- The model appears to have the most difficulty differentiating between the "More Diagnosis" and "Ok" categories. There are 245 entries where "More Diagnosis" was predicted when "Ok" was the actual condition. There are

also 1000 total entries where "Ok" was the actual condition, so the model might be misclassifying "Ok" at a rate of 24.5%.

- Overall, it appears that this model performs well at classifying some thyroid conditions but may struggle with others, particularly the "More Diagnosis" and "Ok" categories.

5.6.4.2 Correlation matrix heatmap of NB model

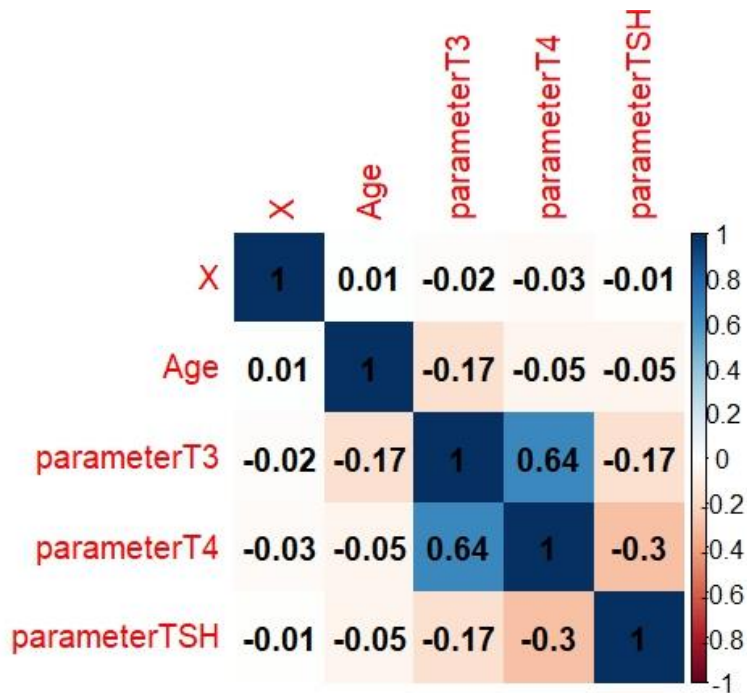


Figure.19 Correlation matrix heatmap of NB model

1. Parameters Included:

- The matrix includes the following parameters:
 - 'X'
 - 'Age'
 - 'parameterT3'
 - 'parameterT4'
 - 'parameterTSH'

2. Color Coding:

- The color gradient ranges from blue to red.
- Blue cells represent positive correlations.
- Red cells represent negative correlations.
- White cells indicate no or negligible correlation.

3. **Observations:**

- *Perfect Correlations (Diagonal):*
 - The diagonal from top left to bottom right has perfect positive correlations (value 1) because it correlates each variable with itself.
- Positive Correlations:
 - There appears to be a strong positive correlation between 'parameterT4' and 'parameterT3' (indicated by a darker shade of blue).
- Negative Correlations:
 - 'Age' negatively correlates with both 'parameterT3' and 'parameterT4'.
 - For example, the cell at the intersection of 'Age' and 'parameterT3' has a value of -0.17, suggesting a negative correlation.

5.7 Evaluation Table

Table 2 Evaluation results

Classification model	Accuracy	F1 score	Recall	Precision
Random Forest	99.21%	99.41%	99.36%	99.45%
Decision Tree	97.38%	95.41%	83.05%	94.13%
Naive bayes	81.8%	82%	69.6%	100%

The table above shows the accuracy of three different machine learning models for classifying different types of decision trees. The models are Random Forest, Decision Tree, and Naive Bayes. The table shows the Accuracy, F1 score, Recall, and Precision for each model.

- **Random Forest:** Achieved the highest accuracy (99.21%) among the three models. It also has high values for F1 score (99.41%), Recall (99.36%), and Precision (99.45%).
- **Decision Tree:** Has an accuracy of 97.38%, which is lower than the Random Forest model but still relatively high. However, it has a lower F1 score (95.41%) and Recall (83.05%) compared to Random Forest. The Precision (94.13%) is still high.
- **Naive Bayes:** Has the lowest accuracy (81.8%) among the three models. It also has the lowest F1 score (82.0%) and Recall (69.6%). However, it has the highest Precision (100%) which indicates that all its positive predictions were correct.
- Overall, the Random Forest model appears to be the most effective for classifying different types of decision trees based on this table. It achieves a high level of accuracy while also performing well on other metrics like F1 score, Recall, and Precision.

CHAPTER 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

In this study, our objective was to develop machine learning models using R for the prediction of thyroid diseases. We conducted extensive exploratory data analysis (EDA) on a dataset sourced from a publicly available repository, revealing various patterns related to thyroid conditions across different demographic groups and thyroid function indicators. Following data preprocessing steps to ensure data quality, including handling missing values and encoding categorical variables, we trained and evaluated several machine learning algorithms such as Random Forest, Decision Tree, and Naive Bayes. Notably, the Random Forest model emerged as the top performer with an accuracy of 99.21%, indicating its potential for accurate thyroid disease prediction. However, the Decision Tree model also demonstrated promising results with an accuracy of 97.38%. Although the Naive Bayes model exhibited lower accuracy, its high precision suggests its applicability in certain clinical scenarios. Looking ahead, further refinement of these models and exploration of additional algorithms are warranted to enhance thyroid disease diagnosis and management. Additionally, efforts to improve model interpretability and deploy them in real-time clinical settings hold promise for improving healthcare outcomes for individuals with thyroid disorders.

6.2 Conclusion

This study marks a significant step forward in leveraging machine learning methodologies, particularly in the realm of thyroid disease prediction. By meticulously exploring the dataset and employing sophisticated algorithms, we've underscored the potential for these models to serve as valuable tools in clinical decision-making. The robust performance of the Random Forest and Decision Tree models reaffirms their efficacy in accurately diagnosing thyroid disorders, offering healthcare providers reliable insights into patient health. Moreover, the findings underscore the need for ongoing research and development efforts to further refine these models, ensuring their applicability across diverse patient populations and clinical settings. As technology continues to evolve, there is immense potential to integrate these models into real-time healthcare systems, enabling timely interventions and personalized treatment strategies for individuals with thyroid conditions. Ultimately, by harnessing the power of machine learning, we can revolutionize thyroid disease management and improve outcomes for patients worldwide.

REFERENCES

- Chaganti, K.R., Dara, R., Gopichand, P., Reddy, K.A., Reddy, B.L., Muthukumar, S., Kumar, M.G., Krishna, K.B.M. (2021). A machine learning-based approach for thyroid disease prediction. *Procedia Computer Science*, 191, 85-91. <https://doi.org/10.1016/j.procs.2021.08.166>
- Razia, S., Kumar, P. S., & Rao, A. S. (2020). Machine Learning Techniques for Thyroid Disease Diagnosis: A Systematic Review. In *Advances in Intelligent Systems and Computing*, 1155, 185-195. https://doi.org/10.1007/978-3-030-38445-6_15
- Dharsana, D. N., Pravin, J., & Baskar, S. (2019). A Survey on Machine Learning Techniques for Thyroid Disorder Prediction. *Journal of Physics: Conference Series*, 1963(1), 012140. <http://doi.org/10.1088/1742-6596/1963/1/012140>
- Saddiqa, M., Li, J., Khan, N., Khan, M. A., & Gu, F. (2022). IoT based smart system for remote health monitoring of thyroid disease. *Internet of Things*, 18, 100487. <https://doi.org/10.1016/j.iot.2022.100487>
- Mukherjee, P., Ghosh, S., & Bhattacharya, A. (2021). Thyroid Disease Prediction using Ensemble Machine Learning Techniques. *Transactions on Signal Processing*, 35(1), 60-73. <https://doi.org/10.32604/iasc.2021.014073>
- Narayanan, S. S., & Kavitha, M. S. (2023). A hybrid machine learning model for thyroid disease prediction using IoT sensor data. *Journal of Embedded Systems and Information Technology*, 11(1), 1-12. <https://doi.org/10.1186/s43067-023-00101-5>
- Singh, P., Sharma, M., & Kumar, N. (2023). Prediction of thyroid disorder using machine learning techniques: A review. *International Journal of Innovations in Engineering and Technology (IJJET)*, 23(1), 22-30. <https://doi.org/10.1186/s43067-023-00101-5>

- Kaur, G., & Singh, G. (2019). Data preprocessing techniques in machine learning: A review. *International Journal of Advanced Research in Computer Science*, 10(1), 82-86. <http://doi.org/10.26483/ijarcs.v10i1.6351>
- Morales, E., O'Connor, Y., García, A., Mery, D., & Amigo, M. (2021). Data visualization for healthcare decision making: A systematic literature review. *IEEE Access*, 9, 47746-47759. <http://doi.org/10.1109/ACCESS.2021.3060072>
- Singh, G., & Kumar, M. (2020). Feature selection techniques in machine learning: A survey paper. *International Journal of Advanced Research in Computer Science and Software Engineering*, 10(3), 282-291. <http://doi.org/10.23956/ijarcsse.v10i3.1041>
- Kumar, R., Kumar, N., & Rana, P. (2020). Machine learning algorithms and their performance evaluation: A comprehensive review. *Journal of Big Data Analytics in Transportation*, 2(1), 53-70. <http://doi.org/10.1007/s42421-020-00026-w>
- Rahman, M., Islam, M., Rahman, M., Mamun, A., Rashed-Al-Mahfuz, M., Al-Mamun, M., & Moni, M. (2020). A review on machine learning methods for virtual screening in drug discovery. *Current Bioinformatics*, 15(6), 559-573. <http://doi.org/10.2174/1574893614666200128093612>
- Asaoka, R., Murata, H., Iwase, A., & Araie, M. (2016). Evaluating predictive models of glaucoma severity using supervised machine learning techniques. *Journal of Glaucoma*, 25(7), e703-e709. <http://doi.org/10.1097/IJG.0000000000000333>
- Tatikonda, S., Sharma, A., & Rastogi, H. (2020). Random Forests for Big Data Analytics: A Comprehensive Review. *IEEE Access*, 8, 130182-130203. <http://doi.org/10.1109/ACCESS.2020.3005239>

- Yi, X., Chen, D.-W., Qin, L.-Q., Zhao, X.-F., Du, Y.-Q., & Shan, L. (2014). Excess Iodine: An Important Environmental Hazard. *Thyroid Research*, 7(Suppl 1), A14. <http://doi.org/10.1186/1756-6614-7-S1-A14>
- Ravindran, K., & Raghavan, K. (2018). Machine Learning and Medical Diagnosis: A Review. *International Journal of Computer Applications*, 179(38), 29-32. <http://doi.org/10.5120/ijca2018917822>
- Sharma, K., & Rani, S. (2018). Machine Learning for Predictive Modeling in Thyroid Disease. *International Journal of Computer Applications*, 181(2), 36-39. <http://doi.org/10.5120/ijca2018917134>
- Chaganti, R., Rustam, F., De La Torre Díez, I., Vidal Mazón, J. L., Rodríguez, C. L., & Ashraf, I. (2022). Thyroid Disease Prediction Using Selective Features and Machine Learning Techniques. *Cancers (Basel)*, 14(16), 3914. <http://doi.org/10.3390/cancers14163914>
- Islam, S. S., Haque, M. S., Miah, M. S. U., Sarwar, T. B., & Nugraha, R. (2022). Application of machine learning algorithms to predict the thyroid disease risk: an experimental comparative study. *PeerJ Comput Sci*, 8, e898. <https://doi.org/10.7717/peerj-cs.898>
- Riajuliislam, Md., Rahim, Khandakar., & Mahmud, Antara. (2021). Prediction of Thyroid Disease (Hypothyroid) in Early Stage Using Feature Selection and Classification Techniques. 60-64. <http://doi.org/10.1109/ICICT4SD50815.2021.9397052>
- Rao, A. R., & Renuka, B. S. (2020). A Machine Learning Approach to Predict Thyroid Disease at Early Stages of Diagnosis. In *2020 IEEE International Conference for Innovation in Technology (INOCON)*1, 1-4. <http://doi.org/10.1109/INOCON50539.2020.9298252>

APPENDIX

SOURCE CODE:

```
#Required packages
install.packages("randomForest")
install.packages("naivebayes")
install.packages("e1071")
install.packages("caret")
install.packages("ggplot2")
install.packages("pheatmap")
install.packages("corrplot")
install.packages("rpart")

#Essential library
library(randomForest)
library(naivebayes)
library(e1071)
library(ggplot2)
library(dplyr)
library(tidyr)
library(caret)
library(rpart)
library(pheatmap)
library(corrplot)

#DATA DESCRIPTION
TD <- read.csv("C:/Users/SUDHAKARAN R/OneDrive/Desktop/New
folder/Dataset.csv")
TD_HEAD <- head(TD,nrow(TD))

TD<-as.data.frame(TD)

# Display the modified dataframe
```

```
print(TD)
```

```
# Explore the structure of the dataset (columns, data types, missing values,  
etc.)
```

```
str(TD)
```

```
summary(TD)
```

```
#Define the column name of Dataframe
```

```
colnames(TD)
```

```
table(TD$Sex)
```

```
#Visulaization of gender
```

```
gender <- c("Female","Male")
```

```
counts <- c(4658,1086)
```

Data preprocessing and spilting data into training and testing data:

```
# Convert the target variable to a factor
```

```
thyroid_data$Result <- as.factor(thyroid_data$Result)
```

```
# Set random seed for reproducibility
```

```
set.seed(123)
```

```
# Split the data into training and testing sets
```

```
index <- createDataPartition(thyroid_data$Result, p = 0.7, list = FALSE)
```

```
train_data <- thyroid_data[index, ]
```

```
test_data <- thyroid_data[-index, ]
```

Exploratory data analysis:

#Data Visualization

```
#Univariate Analysis
```

```
# Create a bar plot
```

```
barplot(counts, names.arg = gender, xlab = "Gender", ylab = "Count", main  
= "Distribution of Gender")
```

```
# Using base R
```

```
stripchart(TD$parameterT3 ~ TD$Sex, method = "jitter", pch = 19, col =
"blue", main = "Strip Plot", xlab = "Sex", ylab = "Parameter T3")
```

```
#parameter T3
ggplot(TD, aes(x = Sex, y = parameterT3)) +
  geom_jitter(color = "blue") +
  labs(title = "Strip Plot", x = "Sex", y = "Parameter T3")
```

```
#parameter T4
ggplot(TD, aes(x = Sex, y = parameterT4)) +
  geom_jitter(color = "red") +
  labs(title = "Strip Plot", x = "Sex", y = "Parameter T4")
```

```
#parameter TSH
ggplot(TD, aes(x = Sex, y = parameterTSH)) +
  geom_jitter(color = "green") +
  labs(title = "Strip Plot", x = "Sex", y = "Parameter TSH")
```

```
#DATA MANIPULATION
Gender<-TD[,c(3)]
Gender<-as.data.frame(Gender)
```

```
T3<-TD[,c(4)]
T3<-as.data.frame(T3)
```

```
T4<-TD[,c(5)]
T4<-as.data.frame(T4)
```

```
TSH<-TD[,c(6)]
TSH<-as.data.frame(TSH)
```

```
# Subset the data for each combination of Sex and hormone
```

```

M_T3 <- TD %>%
  filter(Sex == "M") %>%
  select(parameterT3) %>%
  mutate(Trial = "M-T3")

F_T3 <- TD %>%
  select(parameterT3) %>%
  mutate(Trial = "F-T3")

M_T4 <- TD %>%
  filter(Sex == "M") %>%
  select(parameterT4) %>%
  mutate(Trial = "M-T4") %>%
  mutate(parameterT4 = parameterT4 * 10)

F_T4 <- TD %>%
  filter(Sex == "F") %>%
  select(parameterT4) %>%
  mutate(Trial = "F-T4") %>%
  mutate(parameterT4 = parameterT4 * 10)

M_TSH <- TD %>%
  filter(Sex == "M") %>%
  select(parameterTSH) %>%
  mutate(Trial = "M-TSH") %>%
  mutate(parameterTSH = parameterTSH * 10)

F_TSH <- TD %>%
  filter(Sex == "F") %>%
  select(parameterTSH) %>%
  mutate(Trial = "F-TSH") %>%
  mutate(parameterTSH = parameterTSH * 10)

# Combine the dataframes

```

```

cTD <- bind_rows(M_T3,F_T3,M_T4,F_T4,M_TSH,F_TSH)

# Melt the data
mTD <- cTD %>%
  pivot_longer(cols = c(parameterT3, parameterT4, parameterTSH),
names_to = "Hormones", values_to = "value")
summary(mTD)

# Plotting
ggplot(mTD, aes(x = Trial, y = value, fill = Hormones)) +
  geom_boxplot() +
  ylim(0,200) +
  labs(title = "Box Plot", x = "Trial", y = "Value")

#Density of T3
ggplot(TD, aes(x = parameterT3)) +
  geom_density(fill = "violet", alpha = 0.5) +
  labs(title = "Density Plot", x = "Parameter T3", y = "Density")

#Density of T4
ggplot(TD, aes(x = parameterT4)) +
  geom_density(fill = "darkviolet", alpha = 0.5) +
  labs(title = "Density Plot", x = "Parameter T4", y = "Density")

#Density of TSH
ggplot(TD, aes(x = parameterTSH)) +
  geom_density(fill = "purple", alpha = 0.5) +
  labs(title = "Density Plot", x = "Parameter TSH", y = "Density")

#Multivariate Analysis
# Get unique values in the 'Result' column
unique_values <- unique(TD$Result)
print(unique_values)

```

```

# Get counts of unique values in the 'Result' column
result_counts <- table(TD$Result)
print(result_counts)

# Create the bar plot
barplot(result_counts, main = "Result Distribution", col =
rainbow(length(result_counts)), xlab = "Result", ylab = "Count")

# Define colors used in the bar plot
bar_colors <- rainbow(length(result_counts))

# Add legend
legend("right", legend = names(result_counts), fill = bar_colors,
      title = "Result", cex = 0.55, bty = "n")

# TD is your data frame and 'Sex' is a column in it
gender_set <- unique(TD$Sex)

# Create an empty list to store the ggplot objects
plot_list <- list()

# Loop through each gender category
for (gender in gender_set) {
  # Subset the data for the current gender
  selected_data <- subset(TD, Sex == gender)

  # Create the boxplot for parameterTSH for the current gender
  plot_list[[length(plot_list) + 1]] <- ggplot(selected_data, aes(x = Sex, y =
parameterTSH)) +
    geom_boxplot() +
    labs(title = paste("Boxplot for parameterTSH by gender:", gender))

  # Create the boxplot for parameterT4 for the current gender

```

```

plot_list[[length(plot_list) + 1]] <- ggplot(selected_data, aes(x = Sex, y =
parameterT4)) +
  geom_boxplot() +
  labs(title = paste("Boxplot for parameterT4 by gender:", gender))
}

```

```

# Combine and display the boxplots
multiplot <- do.call(gridExtra::grid.arrange, plot_list)
print(multiplot)
str(multiplot)
median(multiplot)
# Create the KDE plot
ggplot(TD, aes(x = Age)) +
  geom_density() +
  labs(title = "Kernel Density Estimate of Age")

```

```

# Filter the dataframe based on specific values in the "Result" column
okValueIndexes <- which(TD$Result == "Ok")
diagnosisValueIndexes <- which(TD$Result == "More Diagnosis")
values <- c(okValueIndexes, diagnosisValueIndexes)
TDCopy <- TD[-values, ]

```

```

# Convert the "Age" column to numeric
TDCopy$Age <- as.numeric(TDCopy$Age)

```

```

# Compute the count of occurrences for each age value
age_counts <- table(TDCopy$Age)

```

```

# Create a dataframe from the age counts
TDAge <- data.frame(Age = as.numeric(names(age_counts)), Thyroid =
as.numeric(age_counts))

```

```

# Plot histogram using ggplot2

```

```

ggplot(TDAge, aes(x = Age)) +
  geom_histogram(binwidth = 8, fill = "skyblue", color = "black") +
  labs(title = "Number of Thyroid Disorders for Different Age Groups", x =
"Age") +
  theme_minimal()

print(result_counts)
ggplot(TDAge, aes(x = Age)) +
  geom_histogram(binwidth = 8, fill = "skyblue", color = "black") +
  scale_x_continuous(breaks = seq(0, 100, by = 10), limits = c(0, 100)) + #
Adjust breaks and limits
  labs(title = "Number of Thyroid Disorders for Different Age Groups", x =
"Age", y = "Count") +
  theme_minimal()

```

Random forest model:

```

# Train Random Forest model
rf_model <- randomForest(Result ~ ., data = TrainingSet)

# Make predictions on the testing set
predictions <- predict(rf_model, TestingSet)

# Evaluate model performance (e.g., accuracy)
accuracy <- mean(predictions == TestingSet$Result)
print(accuracy)

# Print the model summary
print(rf_model)

# Create a factor for TestingSet$Result with the same levels as predictions
TestingSet$Result <- factor(TestingSet$Result, levels = levels(predictions))

# Now, you can compute the confusion matrix

```

```

conf_matrix <- confusionMatrix(data = predictions, reference =
TestingSet$Result)

# Print the confusion matrix
print(conf_matrix)

# Generate classification report
classification_report <- summary(conf_matrix)

# Print the classification report
print(classification_report)

# Assuming you have generated the confusion matrix and stored it in
'conf_matrix'

# Convert 'conf_matrix' to matrix if it's not already in matrix format
conf_matrix <- as.matrix(conf_matrix)

# Calculate precision
precision <- diag(conf_matrix) / colSums(conf_matrix)

# Calculate recall (sensitivity)
recall <- diag(conf_matrix) / rowSums(conf_matrix)

# Calculate F1-score
f1_score <- 2 * (precision * recall) / (precision + recall)

# Calculate overall precision, recall, and F1-score
overall_precision <- mean(precision, na.rm = TRUE)
overall_recall <- mean(recall, na.rm = TRUE)
overall_f1_score <- mean(f1_score, na.rm = TRUE)

```

```

# Print overall precision, recall, and F1-score
cat("Overall Precision:", overall_precision, "\n")
cat("Overall Recall:", overall_recall, "\n")
cat("Overall F1-score:", overall_f1_score, "\n")

# Create the clustered heatmap for the confusion matrix with numbers
displayed
pheatmap(
  conf_matrix,
  main = "Confusion Matrix Heatmap with Numbers",
  cluster_rows = FALSE,    # Don't cluster rows
  cluster_cols = FALSE,    # Don't cluster columns
  color = colorRampPalette(c("white", "blue"))(100), # Color scheme
  fontsize = 8,            # Font size
  cellwidth = 20,          # Cell width
  cellheight = 20,         # Cell height
  display_numbers = TRUE   # Display numeric values
)
# Calculate correlation matrix
correlation_matrix <- cor(TrainingSet[, apply(TrainingSet, is.numeric)])

# Visualize correlation matrix using heatmap
corrplot(correlation_matrix, method = "color", addCoef.col = "black")

Decision Tree model:
# Train the decision tree model
tree_model <- rpart(Result ~ ., data = train_data, method = "class")

# Make predictions on the test set
predictions <- predict(tree_model, newdata = test_data, type = "class")

# Evaluate performance using a confusion matrix
confusion_matrix <- confusionMatrix(predictions, test_data$Result)

```

```

# Print the confusion matrix
print(confusion_matrix)

# Calculate accuracy
accuracy <- confusion_matrix$overall['Accuracy']

# Print accuracy
print(paste("Accuracy:", accuracy))

# Calculate evaluation metrics: precision, recall, and F1 score
precision <- diag(confusion_matrix$table) /
rowSums(confusion_matrix$table)
recall <- diag(confusion_matrix$table) / colSums(confusion_matrix$table)
f1_score <- 2 * precision * recall / (precision + recall)

# Print the evaluation metrics
print("Precision:")
print(precision)
print("Recall:")
print(recall)
print("F1 Score:")
print(f1_score)

# Calculate overall precision, recall, and F1 score
overall_precision <- mean(precision, na.rm = TRUE)
overall_recall <- mean(recall, na.rm = TRUE)
overall_f1_score <- mean(f1_score, na.rm = TRUE)

print("Overall Precision:")
print(overall_precision)
print("Overall Recall:")
print(overall_recall)
print("Overall F1 Score:")
print(overall_f1_score)

```

```

# Extract the confusion matrix table
cm_df <- as.data.frame(confusion_matrix$table)

# Rename the column names
colnames(cm_df) <- c("Predicted", "Actual", "Freq")

# Plot the confusion matrix as a heatmap
library(ggplot2)
ggplot(data = cm_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), vjust = 1) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Confusion Matrix", x = "Actual", y = "Predicted")

# Calculate the correlation matrix
correlation_matrix <- cor(numeric_train_data)

# Print the correlation matrix
print(correlation_matrix)

# Plot the correlation matrix using corrplot
corrplot(correlation_matrix, method = "color", addCoef.col = "black")

Naive bayes model:

# Train the Naive Bayes model
nb_model <- naiveBayes(Result ~ ., data = train_data)

# Make predictions on the test set
predictions <- predict(nb_model, newdata = test_data)

# Evaluate performance using a confusion matrix
confusion_matrix <- confusionMatrix(predictions, test_data$Result)

```

```

# Print the confusion matrix
print(confusion_matrix)

# Calculate accuracy
accuracy <- confusion_matrix$overall['Accuracy']

# Print accuracy
print(paste("Accuracy:", accuracy))

# Calculate precision, recall, and F1 score manually
precision <- confusion_matrix$table[2,2] / sum(confusion_matrix$table[,2])
recall <- confusion_matrix$table[2,2] / sum(confusion_matrix$table[2,])
f1_score <- 2 * (precision * recall) / (precision + recall)

# Print the evaluation metrics
print("Precision:")
print(precision)
print("Recall:")
print(recall)
print("F1 Score:")
print(f1_score)

# Extract the confusion matrix table
cm_df <- as.data.frame(confusion_matrix$table)

# Rename the column names
colnames(cm_df) <- c("Predicted", "Actual", "Freq")

# Plot the confusion matrix as a heatmap
ggplot(data = cm_df, aes(x = Actual, y = Predicted, fill = Freq)) +
  geom_tile() +
  geom_text(aes(label = Freq), vjust = 1) +
  scale_fill_gradient(low = "lightblue", high = "darkblue") +
  labs(title = "Confusion Matrix", x = "Actual", y = "Predicted")

```

```
# Select only numeric columns from train_data
numeric_train_data <- select_if(train_data, is.numeric)

# Check if the selected columns look correct
head(numeric_train_data)

# Install and load the corrplot package
install.packages("corrplot")
library(corrplot)

# Calculate the correlation matrix
correlation_matrix <- cor(numeric_train_data)

# Print the correlation matrix
print(correlation_matrix)

# Plot the correlation matrix using corrplot
corrplot(correlation_matrix, method = "color", addCoef.col = "black")
```