DETECTION OF THYROID DISEASE USING MACHINE LEARNING MODELS

2023
MASTER THESIS
COMPUTER ENGINEERING

Muntadher Adnan Waheed ALSAADAWI

Thesis Advisor
Assist. Prof. Dr. Eftal ŞEHİRLİ

DETECTION OF THYROID DISEASE USING MACHINE LEARNING MODELS

Muntadher Adnan Waheed ALSAADAWI

Thesis Advisor

Assist. Prof. Dr. Eftal ŞEHİRLİ

T.C.

Karabuk University

Institute of Graduate Programs

Department of Computer Engineering

Prepared as

Master Thesis

KARABUK

January 2023

I certify that in my opinion the thesis submitted by Muntadher Adnan Waheed ALSAADAWI titled “DETECTION OF THYROID DISEASE USING MACHINE LEARNING MODELS ” is fully adequate in scope and in quality as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Eftal ŞEHİRLİ .....
Thesis Advisor, Department of Medical Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Computer Engineering as a Master of Science thesis. Jan 19,2023

Examining Committee Members (Institutions)	Signature
Chairman : Prof. Dr. İsmail Rakıp KARAŞ (KBU)	.....
Member : Assist. Prof. Dr. Birsen GÜLDEN ÖZDEMİR (DU)	.....
Member : Assist. Prof. Dr. Eftal ŞEHİRLİ (KBU)	.....

The degree of Master of Science by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Müslüm KUZU .....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Muntadher Adnan Waheed ALSAADAWI

ABSTRACT

M. Sc. Thesis

DETECTION OF THYROID DISEASE USING MACHINE LEARNING MODELS

Muntadher Adnan Waheed ALSAADAWI

Karabük University
Institute of Graduate Programs
The Department of Computer Engineering

Thesis Advisor:
Assist. Prof. Dr. Eftal ŞEHİRLİ
Jan 2023, 102 pages

Disease diagnosis and prognosis are among the most crucial uses of machine learning (ML) models. In recent years, ML models have played a crucial and persuasive role in disease diagnosis and classification. Thyroid disease is an issue for human health that needs attention since the thyroid gland regulates human metabolism and plays a crucial role in managing human health. This thesis presents a method for classifying thyroid disease using traditional ML models (K-nearest Neighbor (KNN), Support Vector Machine (SVM), Decision Tree (DT), Naive Bayes (NB), Logistics Regression (LR), and Multi-Layer Perceptron (MLP) and ensemble models (Random Forest (RF), XGBoost, Soft Vote, Stacking, and Bagging). The proposed method was trained and tested in two steps, first using all features of the dataset and then using the best-correlated features selected by the Recursive Feature Elimination (RFE) model. The highest accuracy (ACC) of traditional models with all features was found to be obtained by DT and MLP at

99.92% and 97.30%, respectively. Ensemble models obtained 100% of ACC in the XGboost and Bagging models. The RFE model was applied to the dataset and achieved 100% and 98.06% ACC in DT and NB, respectively. As for ensemble models, XGBoost and Bagging also achieved 100% of ACC, and the Stacking model achieved 99.53% of ACC. The proposed ensemble models outperformed the traditional models in terms of sensitivity, specificity, precision, F1 score, and Matthews Correlation Coefficient (MCC) as well as ACC. The proposed models were tested for overfitting using feature selection, cross-validation and comparison of training and test ACC. The time spent for training and prediction was found to be reasonable.

Key Words : Machine Learning, Ensemble models, Thyroid disease, Hypothyroidism, Hyperthyroidism.

Science Code : 92432

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENİMİ MODELLERİ KULLANILARAK TİROİD HASTALIĞININ TESPİTİ

Muntadher Adnan Waheed ALSAADAWI

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Tez Danışmanı:
Dr. Öğr. Üyesi Eftal ŞEHİRLİ
Ocak 2023, 102 sayfa

Hastalık teşhisi ve tahmini makine öğrenmesi modellerinin en önemli kullanım alanları arasında yer almaktadır. Son yıllarda, bu konuda makine öğrenmesi modelleri önemli ve ikna edici bir rol üstlenmiştir. Tiroid bezi insan metabolizmasını düzenlediği ve insan sağlığında önemli bir rol oynadığı için tiroid hastalığı insan sağlığı için dikkat edilmesi gereken bir sorundur. Bu tez, geleneksel makine öğrenmesi modelleri olan K-en Yakın Komşu, Destek Vektör Makinesi, Karar Ağacı, Naive Bayes, Lojistik Regresyon ve çok katmanlı perseptron ile topluluk öğrenme modelleri olan Rastgele Orman, XGBoost, Soft Vote, Stacking ve Bagging kullanarak tiroid hastalığını sınıflandırmak için bir yöntem sunmaktadır. Önerilen yöntem, önce veri kümesinin tüm özniteliklerini kullanarak ve ardından özyinelemeli öznitelik eleme yöntemi tarafından seçilen en iyi ilişkili özellikleri kullanarak iki adımda eğitilmiş ve test edilmiştir. Tüm özniteliklere sahip geleneksel modellerin en yüksek doğruluğu sırasıyla %99.92 ve %97.30 ile karar ağacı ve çok katmanlı

perseptron tarafından elde edilmiştir. Toplu modeller için XGboost ve Bagging modelleri %100 doğruluk elde etmiştir. Özyinelemeli öznitelik eleme yöntemi veri setine uygulanmış ve geleneksel makine öğrenmesi modellerinden karar ağacı ile Naive Bayes modelleri sırasıyla %100 ve %98.06 doğruluk elde etmiştir. Topluluk modellerinden XGBoost ve Bagging %100 doğruluk ve Stacking modeli %99.53 doğruluk elde etmiştir. Önerilen topluluk modelleri doğruluk parametresi ile birlikte duyarlılık, özgüllük, kesinlik, F1 puanı ve Matthews Korelasyon Katsayısı açısından geleneksel modellerden daha iyi performans göstermiştir. Önerilen modeller, öznitelik eleme ve çapraz doğrulamanın yanında eğitim ve test doğruluklarının karşılaştırılması kullanılarak aşırı uyum için test edilmiştir. Eğitim ve tahmin işlemleri için harcanan zaman makul olarak değerlendirilmiştir.

Anahtar Kelimeler : Makine öğrenmesi, Topluluk öğrenme, tiroid hastalığı, hipotiroidizm, hipertiroidizm.

Bilim Kodu : 92432

ACKNOWLEDGMENT

My most profound appreciation goes to Dr. Eftal ŞEHİRLİ my master's degree advisor and mentor, for his time, effort, and understanding in helping me succeed in my thesis, his vast wisdom and wealth of experience have inspired me throughout my thesis. I want to express my gratitude to everyone at Karabük University. Thanks to their generosity and encouragement, my time spent studying and living in Turkey has been truly rewarding. To conclude, I'd like to thank God, my parents, my brother, my sisters, and my friends. It would have been impossible to finish my studies without their unwavering support over the past few years.

CONTENTS

Page

Page

Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

Page

SYMBOLS AND ABBREVITIONS

ABBREVITIONS

ACC	:	Accuracy
AI	:	Artificial Intelligence
CAD	:	Computer aided diagnosis
DT	:	Decision Tree
FN	:	False Negative
FP	:	False Positive
KNN	:	K-Nearest Neighbor
LR	:	Logistic Regression
MCC	:	Matthews Correlation Coefficient
ML	:	Machine Learning
MLP	:	Multilayer Perceptron
NB	:	Naïve Bayes
RF	:	Random Forest
SMOTE	:	Synthetic Minority Oversampling Technique
SVM	:	Support Vector Machine
T3	:	Tri-iodothyronine
T4	:	L-thyroxine
TN	:	True Negative
TP	:	True Positive
TSH	:	Thyrotropin-Stimulating Hormone
WHO	:	World Health Organization

PART 1

INTRODUCTION

A thyroid disease occurs when the thyroid gland is unable to produce the typical quantities of hormones, which in turn causes problems with the body's ability to operate normally. The medical professionals are able to identify such a problem based on the findings of the physical investigation and the medical examination, then they may start the appropriate treatment course. The procedure for making a diagnosis is dependent on a battery of testing, which may include blood tests and urine tests [1]. Patients who have thyroid issues are cared for at the internal medicine department. As a consequence of this, there is a need for additional medical professionals or means of instant diagnosis as the human population continues to expand. Alternatively, a computer aided diagnosis (CAD) systems are worth to develop help medical professionals.

Endocrinology, which includes thyroid illness, is one of the medical specialties that is most often misunderstood and underdiagnosed [2]. The World Health Organization (WHO) reports that diabetes is the illness that affects more people throughout the world than any other endocrine condition, but thyroid disease is just behind it. The conditions known as hyperfunction, hyperthyroidism and hypothyroidism each afflict around 1% and 2% of people, respectively [3]. According to recent studies, women are 5 to 8 times more prone than males to suffer thyroid problems [30]. Both hyper and hypothyroidism may have a variety of root causes, including dysfunction of the thyroid gland, failure of the pituitary gland, or tertiary dysfunction of the hypothalamus [3]. In situations with severe iodine deficiency, the prevalence of goiter can reach 80% [4]. This can cause goiter to become more frequent. The thyroid gland is a potential site for the development of many distinct types of malignancies as well as a hazardous area in which endogenous antibodies wreak havoc [5].

The diagnosis of thyroid disease is a highly time-consuming and challenging process. Clinical examination and a number of blood tests are required in order to arrive at a diagnosis of thyroid illness using the conventional method. The primary challenge, however, is to accurately diagnose the illness in its earliest stages as a high proportion of the time. In the realm of medicine, Data Mining plays an important role in the process of illness diagnosis [6]. Data Mining offers a wide variety of categorization strategies that may improve the accuracy (ACC) of illness prediction. The examination of risk factors for a variety of illnesses may make use of the patient information that has been received from a variety of health care organizations [7].

Tri-iodothyronine (T3) and L-thyroxine (T4) are two hormones produced by the thyroid gland [8]. Thyroid hormones control several aspects of metabolism, including energy production, digestion, and thermogenesis. Hormones like T3 and T4 are synthesized by the pituitary gland. When the body needs more thyroid hormone, the pituitary gland secretes Thyrotropin-Stimulating Hormone (TSH) [9], which travels via the circulation to the thyroid gland. Next, the thyroid is stimulated by TSH to produce T3 and T4 hormones [8]. The pituitary glands feedback mechanism regulates thyroid hormone production [8]. When levels of T3 and T4 are high, the body produces less TSH. When levels of T3 and T4 are low, the produces more TSH [8].

Hypothyroidism has several complications such as thyroid surgery, exposure to ionizing radiation, persistent inflammation of the thyroid glands or auto-immune thyroid, iodine shortage, and decreased release of enzymes that produce thyroid hormones, The incidence of hypothyroidism ranges from 1% to 2% in areas that have an abundance of iodine. The condition is more prevalent in women over the age of 50 and occurs ten times more frequently in women than in males [4]. Thyroid inflammation and damage cause hypothyroidism. Some symptoms include obesity, a slower heart rate, prolonged exposure to extremely cold or hot temperatures, throat swelling, dry eyes, numb palms, hair loss, irregular menstruation cycles, and digestive difficulties. If treatment is not sought, the severity of these symptoms may increase [4].

Hyperthyroidism, also known as an overactive thyroid, may also be caused by the local physical state of the thyroid, the use of different drugs, and the lack of control over the release of thyroid hormones. Graves' illness is one of the most prevalent complications of hyperthyroidism. Graves' disease develops when the body continues to produce proteins that instruct the thyroid to secrete even more thyroid hormone. The problem of thyroid illness should never be underestimated by thyroid patients since it may lead to fatal diseases such as thyroid storm (a kind of severe hyperthyroidism) and myxedema (the last stage of untreated hypothyroidism) which might lead to death [10].

According to physicians, early diseases identification, diagnosis, and treatment are all very important in controlling the course of a disease and even avoiding death. Early detection and differential diagnosis improve the likelihood of successful therapy for several distinct kinds of abnormalities. In spite of the many experiments that have been conducted, medical diagnosis is often considered to be a challenging endeavor [11]. Data Mining is a technology that looks for patterns and connections in large databases using a semi-automated process [12]. One of the most effective solutions for the majority of difficult issues are algorithms that are used for machine learning (ML) [13]. Diseases like thyroid problems can be predicted and diagnosed with the help of data extraction approach and classification process. Thyroid illness classification is a good example of this point. Due to the importance, high performance and efficiency of ML algorithms for the diagnosis of thyroid disease [14], we undertook the aforementioned research and classification. The use of ML in healthcare has been there since the early days of the industry, but recently there has been a resurgence of interest in this area. [15]. As a consequence of this, experts believe that ML will quickly become standard practice in the medical field [16].

Accurate data analysis and use may improve service in several fields that are important to human life. Due to the significance of data, service providers in both the public and commercial sectors have shown interest in data collecting for future strategy planning. The purpose of data analysis is to anticipate the future status of a certain application by identifying the characteristics that contribute to future development or decline in business sectors. Consequently, significant progress has

been made lately in the field of Data Mining. There have been several kinds of Data Mining algorithms for effective knowledge extraction from so-called large data [17].

Occasionally, data are gathered in certain application by allowing users to manually input their comments. Ready-made data are also accessible for research reasons and may be used for the development of algorithms and optimization [18]. Data science is a discipline concerned with the development of tools and techniques for data analysis; it is primarily divided into three fields: classification, prediction, and clustering [19]. In recent years, it has been apparent that the complexity of life and changes in human dietary habits have led to a major rise in medical complications [16]. In addition, the expense of medical therapy is considered to be on the high side, particularly for compliance that may need surgical intervention [16]. Through the development of intelligent systems, data science and technology may be used to improve medical diagnostics.

1.1. PROBLEM STATEMENT

One of the most prevalent disorders is thyroid gland disease, which is a highly complicated infection caused by excessive levels of (TSH) and by difficulties with the thyroid organ itself [4]. Hashimoto's thyroid condition is the most well-known cause of hypothyroidism [4]. About a third of the global population resides in iodine-deficient regions. In locations where the daily iodine intake is below 50 μg, goiter is frequently prevalent, and congenital hypothyroidism is observed when the daily iodine intake falls below 25 μg. The prevalence of goiter in regions with considerable iodine shortage might reach up to 80% [20]. The majority of persons with thyroid abnormalities have an autoimmune condition, ranging from basic atrophic hypothyroidism to Hashimoto's thyroiditis to Graves' disease-induced thyrotoxicosis. In terms of goiter and thyroid nodules, the most prevalent thyroid illness in the general population is a common physiological goiter [21]. According to various studies, the incidence of diffuse goiter decreases with age; the maximum frequency is seen in pre-menopausal women. Hence, the ratio of women to males is at least 4:1. In contrast, the prevalence of thyroid antibodies and thyroid nodules

increases with age. 1.5% of men and 6.4% of women aged 60 or older in Massachusetts had clinically evident thyroid nodules [22].

Medical data analysis is essential for the development of novel medical theories and the prevention of specific illnesses. Previous studies done in a flurry of Data Mining have demonstrated that the quantity of data in the field that correlates human everyday activities, continues to grow physically. A substantial quantity of data from medical applications is generated every day [23]. Due to the lack of existing technology and recognized product for illness detection, as well as the inability of many nations throughout the globe to offer specialist physicians, In this study, the topic is discussed. Nevertheless, an examination of the literature reveals that ML algorithms exhibit varying degrees of performance. In applications like as medical applications, the precision of the acquired information is vital to the diagnostic method and, therefore, to the patients' lives. However, the medical applications of data mining are still under progress, and the following obstacles exist in this regard. Lifestyles, eating habits, and other environmental influences vary amongst individuals. Thus, therapy applications might vary from region to region. These factors make it challenging to build and implement a generic model. The ethics, storage policy, and digitization of medical data vary by location. The rate of population growth is faster than that of doctors. With the fast advancement of technology, new algorithms have been created.

1.2. PROPOSAL OF THE STUDY

This thesis recommends applying Ensemble models for disease prediction in order to extract information from dataset with a specific ACC level. This thesis employs six traditional models such as KNN, SVM, DT, NB, LR, and MLP. The dataset is used for all models, and the results are compared to Ensemble models that are Random Forest (RF), XGboost, Soft Voting, Stacking and Bagging thought to be the most modern and accurate for improving the average prediction performance over any productive member in the ensemble. The primary goal is to improve prediction ACC and the diagnostic procedure. Another goal is to provide hospitals with utilizing the proposed developed model.

1.3. CONTRIBUTIONS OF THESIS

- Obtaining private data containing 1,250 thyroid samples could increase the size and diversity of the dataset used for studying thyroid disease, which could allow for more robust and accurate analyses and conclusions.
- Using the SMOTE model to balance the dataset could help to address any imbalances in the data and ensure that the results of the study are not biased.
- Using the RFE technique to select the best features could help to identify the most important predictors of thyroid disease and improve the ACC of the ML models.
- Conducting knowledge discovery by comparing different traditional ML models and ensemble models could help to identify the best way to predict thyroid disease and improve the ACC of the models.
- Checking for overfitting using train-test splitting and cross-validation could help to ensure that the models are not overly specialized to the training data and have the ability to generalize to new data.
- Using six different performance metrics (ACC, sensitivity, specificity, precision, F1 score, and MCC) could provide a comprehensive evaluation of the models' performance and allow for a more nuanced understanding of their strengths and limitations.
- Finding the difference between training ACC and test ACC could provide insights into the model’s generalization ability and identify any potential issues with overfitting.
- Finding the training time to prediction time could provide information about the efficiency of the models and help to identify any potential bottlenecks in the process.

Overall, the contributions listed above could be significant in advancing the understanding and detection of thyroid disease and improving the ACC and efficiency of ML models for predicting the condition.

1.4. STRUCTURE OF THESIS

This thesis report is divided into six chapters that discuss the specifics of the work and the findings obtained. The chapter divisions of this thesis report are as follows:

- “Introduction” Part 1 provides an introduction of the thyroid disease and uses of ML in healthcare. The "problem statement" and the "proposal of this research" are also introduced.
- “Literature Reviews” section of Part 2 includes a thorough analysis of current research that have employed Data Mining and ML to diagnose thyroid illness.
- “Python and Essentials Packages” Part 3, Describe what Python is and what libraries are used in this thesis.
- “Machine Learning Techniques” Part 4, describes what ML techniques are used in this thesis.
- Part 5: “Materials” This Part details the data used and how they are collected. The platform used and what are its features.
- Part 6: “Methodology” provides all of the classification method's implementation procedures used in this study.
- Part 7, “Results & Discussion”, The comprehensive results and their discussion received after completing all project processes are included.
- Part 8, “Conclusion” summarizes the information after assessing and interpreting the findings of this study in light of research contributions and future possible development.

PART 2

LITERATURE REVIEWS

Predictions of thyroid illness have been made in the past. Thyroid disease may be predicted using a variety of classifiers, such as DT, NB, and SVM, among others. For thyroid illness prediction, data pre-processing, feature extraction, feature selection is critical. A comprehensive review of numerous features and procedures used to predict thyroid illness is offered.

In the study presented [24], two machine-learning methods known as SVM and RF are used to diagnose thyroid problems. During the course of the inquiry, the Thyroid Dataset that was provided by the University of Califurnia at Irvine (UCI) was used. Thyroid diseases benchmark dataset has 7200 samples with 21 selected features and target. Target labels are: Class 1 is normal, Class 2 hyperthyroidism, and Class 3 hypothyroidism. Both approaches were assessed based on a number of criteria, including ACC, sensitivity, F1-score, and precision. ACC of 91% and 89%, were respectively assigned to the SVM and RF models in this study. According to an ACC, SVM is superior than RF about the diagnosis of thyroid problems.

Nazari et. al. [25] used SVM classifier in order to identify cases of thyroid illness. This work analyzed and contrasted two thyroid datasets, one from UCI it has 215 patients and 5 features. The other from Imam Khomeini Hospital in Tehran, Iran, these 1538 patients have 21 features. Several methods, including Sequential forward selection (SFS) and sequential backward selection (SBS), and a Genetic algorithm (GA) were employed for features selection technique. In this particular scenario, genetic algorithm biased support vector machine (GASVM) demonstrated the highest classification ACC of all of the presented approaches, coming in a 98.62%.

Chaubey et. al. [26]. advocated conducting a study to evaluate and determine the level of ACC achieved by LR, DT, and KNN algorithms when it came to detect and assess thyroid condition. They used a dataset from the UCI ML repository have 215 patients and 5 features, and the dataset was split into three sections: training 70%, validation 15%, and test 15%. There were two different classes for the dataset such as class 0 and class 1. Class 0 indicates that a person has a thyroid, whereas class 1 indicates that they are normal. Only the two most important features, (T4) and (T3). The result of this study is a prediction of whether or not individuals have thyroid illness. Were used as features for the decision tree (DT) model. According to the findings of the research, the KNN technique was superior and achieved an ACC of 96.87%.

Geetha and Baboo presented a classification strategy for thyroid illness [27]. The two most prevalent thyroid disorders among the general population are hyperthyroidism and hypothyroidism. This research utilizes data collected from the UCI repository, which has been preprocessed. The nature of the preprocessed data is multivariate. the available 21 traits are reduced to 10 features using the Hybrid Differential Evolution Kernel-Based Naïve Based technique. The subset of data is now provided to the Kernel Based Naïve Based classification method to determine its fitness. This procedure is repeated 21 to 25 times until the mistakes are decreased or stabilized, at which point the samples is classified. The measured ACC of classification is 97.97%.

Along with a description of the condition and health recommendations, Aswathi and Antony [28] offered a technique for identifying and diagnosing a user's thyroid disease. Thyroid gland data collection with 21 features extracted from UCI ML Respiratory. SVM parameters were optimized using the particle swarm optimization method, and a SVM was employed for classification. Users are given a graphical user interface with a panel to write their own input information. While entering the data, there could be some values that are missing. The KNN technique is used to remove any missing values from user input. The research did not include any performance metrics.

Salman K. and Sonuç [29] A dataset collected from the Iraqi people to predict thyroid disease contains 1250 samples and 17 features into three groups, such as normal containing 957 samples, hypothyroidism containing 142 samples, and hyperthyroidism containing 151 samples, which is the same the dataset that we will use in this thesis. Through the class, it becomes clear that there is great confusion for the unbalanced data, traditional algorithms such as SVM, DT, RF, NB, LR, KNN, LDA and MLP were used. The model was built in the form of two models. In the first model, all features were taken, and the highest ACC of the MLP algorithm was 69.4%. In the second model, three features were removed based on a previous study, and the highest ACC was obtained by RF as 98.93%.

The algorithms DT, NB, SVM, and KNN were implemented by Sidiq et. al. [30]. The data set was taken from one of the well-known laboratories in Kashmir, which contains 807 samples and 6 features. classes contain 3 groups, the normal contains 553 samples, hypothyroidism contains 218 samples, and hyperthyroidism contains 36 samples. DT was determined to have the greatest ACC, reaching 98.89 percent, compared to other classification methods.

Thyroid Illness Prediction with Hybrid ML Methods was suggested by YasirIqbal Mir and Dr. Sonu Mittal [31]. They gathered extensive data from 1,464 Indian patients. This paper proposes an effective framework and employs many ML techniques. This study consisted of three portions: pathological observations, serological notes, and a combination of the two. For their study, they employed five common ML models. Pathological and serological parameters have been identified, and dataset have been collected. The employed dataset had 21 characteristics and one attribute with multiple classes. Bagging, Boosting, NB, J48, and SVM classifiers may all be used to this dataset with success. Results are compared with measures like the confusion matrix, ACC, specificity, sensitivity, precision, Recall, and ROC-Curve. In the first trial, bagging achieved a 98.56% ACC rate. In the second trial, SVM ACC was 99.08%. In the last part experiment, the J48 classifier achieved 92.07% ACC.

Solmaz, Alkan, and Gunay [32] are the authors. It has been suggested that carrying out this study would make it possible for people equipped with mobile devices to acquire up-to-date information about the illness or to seek medical treatment for any ailment in a setting other than a hospital. The Practical Thyroid Analysis System includes a mobile device, a software program that runs on android, a database that uses the SQL language, and a server (MATLAB based decision algorithms). A mobile device operating on the android platform may be used to diagnose functional thyroid illness if the system is used. It was determined that the Ensemble approach, which has a high achievement rate for diagnosing thyroid illness, would be the best classification algorithm to utilize in the system after other classification algorithms were investigated. The Ensemble classification method achieved a success percentage of 99.06% and 99.08%, respectively for the First and Second Data Groups, which were obtained from the University of California's Machine-Learning Database (UCI). The first dataset contains 215 records with 5 features. The second group of datasets contain 7200 patients with 21 features.

A study was steered out of [33] with the purpose of detecting hypothyroidism and hyperthyroidism, which are the two record frequent types of thyroid problems. Both multinomial logistic regression models and neural networks were used in the classification process. Both methods were successful. The study was carried out on 310 patient’s datasets taken from Imam Khomeini Hospital, and even in this instance the models took into consideration demographics as well as hormone features as inputs. In every instance, the mean ACC for the multinomial logistic regression method was 91.04%, while the ACC for the neural network model was 96.03%.

In Table 2.1 it shows the previous studies with the algorithms used and the research year for each research.

Table 2.1. Related literature review.

No	Authors	Reference	Year	Models
1	Shivastuti & Haneet Kour, et. al.	[24]	2021	SVM, RF
2	Salman. K & Sonuç	[29]	2021	RF, KNN, LR, SVM, NB, MLP, LDA, DT
3	Chaubey & Bisen, et. al.	[26]	2020	LR, DT, KNN
4	Yasir Iqbal Mir & Dr. Sonu Mittal	[31]	2020	J48, Boosting, Bagging, NB, SVM
5	Solmaz, R., Alkan, A., & Gunay, M.	[32]	2020	Ensemble classification
6	Shiva Borzouei, Hossein Mahjub, et. al.	[33]	2020	LR, Neural Networks Models
7	Sidiq U, Aaqib, et. al.	[30]	2019	KNN, SVM, DT, NB
8	Aswathi and A. Antony	[28]	2018	KNN and SVM using particle swarm optimization
9	K. Geetha & Baboo	[27]	2016	NB
10	Kousarrizi & F. Seiti	[25]	2012	SVM

PART 3

PYTHON AND ESSENTIALS PACKAGES

3.1. PYTHON

Python is a robust programming language that is simple to learn. It contains high-level data structures that are efficient and basic but effective approach to object-oriented programming. Python's beautiful syntax, dynamic typing, and interpreted nature make it an ideal language for scripting and quick application development across a wide range of platforms [79]. Various programming skills including such as scientific analyses, desktop applications, web - based applications, database programming, microcontroller communication, parallel programming, digital image and signal processing, network programming, and so on [80].

3.1.1. Advantages of Python

- Open-source
- Simple to use and to learn.
- Increased efficiency.
- Flexibility.
- Comprehensive library.
- It is very readable and straightforward to debug.
- Installing a Python application requires minimal modification to run on a wide variety of operating systems and platforms.
- It is a programming language created in a dynamic manner. Therefore, declaring the data type of variables is optional [79, 80].

3.1.2. Disadvantages of Python

- Due to the simplicity of its programming, users encounter problems while dealing with other computer languages.
- Processing time is slow.
- Many problems in the language's layout are only revealed at runtime
- Manually adding packages [79, 80].

3.2. PYTHON ESSENTIALS PACKAGES

Python packages offer a simple and effective approach to solve difficult issues in domains as diverse as scientific computing, data visualization, and data modeling. Complex jobs are better addressed incrementally, one sub task at a time. That’s why computer programmer develop and use components, which are collections of linked code kept in distinct files and designed to solve certain tasks. Python includes a variety of built-in packages and libraries. These programs offer unique characteristics and the ability to complete specific jobs. In addition to them, NumPy, Pandas, Scikit-Learn, Matplotlib, SciPy, seaborn and yellowbrick, packages were utilized [79,81].

3.2.1. NumPy

NumPy is the most important tool for doing computational tasks in the scientific community using Python. Python's adaptability and ease of use are combined with the rapidity of languages such as C and Fortran in this programming environment [81].

NumPy is applicable to:

- Superior array actions (e.g., add, multiply, slice, reshape, index).
- Complete scientific operations.
- Production of random numbers.
- The application of linear algebraic procedures [79, 80].

3.2.2. Pandas

This package can manipulate tabular, time series, and matrix data. It is well-recognized as a quick, effective, and user-friendly data analysis and manipulation tool. It operates with data frame objects; a data frame is a structure specialized to two-dimensional data. Data frames, like database tables and Excel spreadsheets, have rows and columns [79, 80].
Pandas can be used for a variety of purposes, including:

- Reading and writing data from some documents like CSV, Excel, and SQL databases.
- Reshaping and pivoting dataset
- Joining and merging datasets.
- Data collection and transformation.
- Datasets can be sliced, indexed, and subset [79, 80].

3.2.3. Matplotlib

The most widely used data exploration and visualization package is Matplotlib. It may be used to make simple graphs such as line plots, histograms, scatter plots, bar charts, pie charts and so on. This library may also be used to build animated and interactive visuals. Every other visualization library is built on Matplotlib [81].

Almost every property of Matplotlib may be customized, including figure size and DPI, line width, color and style, axes, axis, grid attributes, text, and so on. Furthermore, every action involves more coding, and developing a visually pleasing storyline may be a difficult and time-consuming process. It may be found that it is more productive to utilize a different visualization program depending on the work at hand [79, 80].

3.2.4. Scikit-Learn

Python's most helpful and robust package for ML is Scikit-learn (Sklearn). It provides a consistent Python interface to a variety of fast ML and statistical modeling methods, including classification, regression, clustering, and dimensionality reduction. This library, developed mostly in Python, is developed using NumPy, SciPy, and Matplotlib [34, 35].

3.2.5. SciPy

SciPy is a Python collection of numerical routines that offers the core building blocks for modeling and solving scientific issues. SciPy supports methods for optimization, integration, interpolation, eigenvalue problems, algebraic equations, and differential equations, as well as specific data structures such as sparse matrices and k-dimensional trees. SciPy is constructed on top of NumPy, which offers array data structures and rapid numerical functions. SciPy is the base upon which more advanced scientific libraries, such as scikit-learn and scikit-image, are constructed. SciPy is utilized by scientists, engineers, and others throughout the world [34, 35, 36].

3.2.6. Seaborn

Seaborn is a Python module for creating statistical visuals. It is built upon matplotlib and tightly interacts with pandas' data structures.

Seaborn assists you in exploring and comprehending your data. Its charting routines operate on data frames and arrays comprising whole datasets and conduct the required semantic mapping and statistical aggregation to generate useful charts. Its dataset-oriented, declarative API enables you to concentrate on the meaning of the various aspects of your plots rather than the specifics of how to render them [34, 35, 36].

3.2.7. Yellowbrick

Yellowbrick is a collection of diagnostic and visual analytic tools meant to enhance ML using scikit-learn. The package implements a new core API object, the visualizer, which is an information scikit-learn estimator. Similar to transformers or models, visualizers learn from data by visualizing the model selection process.

Visualizers allow users to drive the model selection procedure by fostering an intuitive understanding of feature engineering, algorithm selection, and hyperparameter tuning. For example, it can assist in the diagnosis of typical problems with model complexity and bias, heteroscedasticity, underfitting and overtraining, and class balance problems. By including visualizers into the workflow for model selection, yellowbrick helps users to lead predictive models toward more effective outcomes more quickly [34, 35, 36].

These are among the most popular visualizers:

- Formalized Report of Classification.
- Confusion Matrix.
- Precision-Recall Curve.
- ROCAUC.

PART 4

MACHINE LEARNING TECHNIQUES

One of the most central subfields of artificial intelligence (AI) and ML focuses on the development of algorithms that give computers the ability to train automatically [37]. The use of ML methods helps to eliminate the need to manually progresses each rule to reach a conclusion or isolate a particular pattern. This is accomplished by training it on a wide range of data sets, which enables it to comprehend both its notion and its structure. That is, the algorithms are taught on their own without human intervention [37]. The capacity of computers to make accurate predictions based on prior experiences is one of the important tasks of ML.

In recent years, ML has made great progress thanks to the fast expansion in computer storage space and processing power that have taken place. The capacity to detect links among enormous volumes of data is one of the many benefits that ML offers. In addition, the simple processing of data is image-based, which helps specialists make tough judgments. In addition to this, it can facilitate the speedy processing of massive volumes of data, something that the human brain could never achieve in such a short length of time [38].

The use of ML methods is widespread across several industries, including the medical profession. Because of the difficulties and expenses associated with clinical data analysis, ML-based approaches have been developed for the healthcare industry [39]. ML is a viable alternative to conventional approaches [40] when developmental time and costs are the most important factors, or when a problem seems too complex to be explored in its totality. In addition, ML may be used when a subject seems to be too complicated to be fully investigated.

There are many other kinds of ML, Figure 4.1 depicts the three most frequent types of learning, which are supervised learning, unsupervised learning, and reinforcement learning respectively.

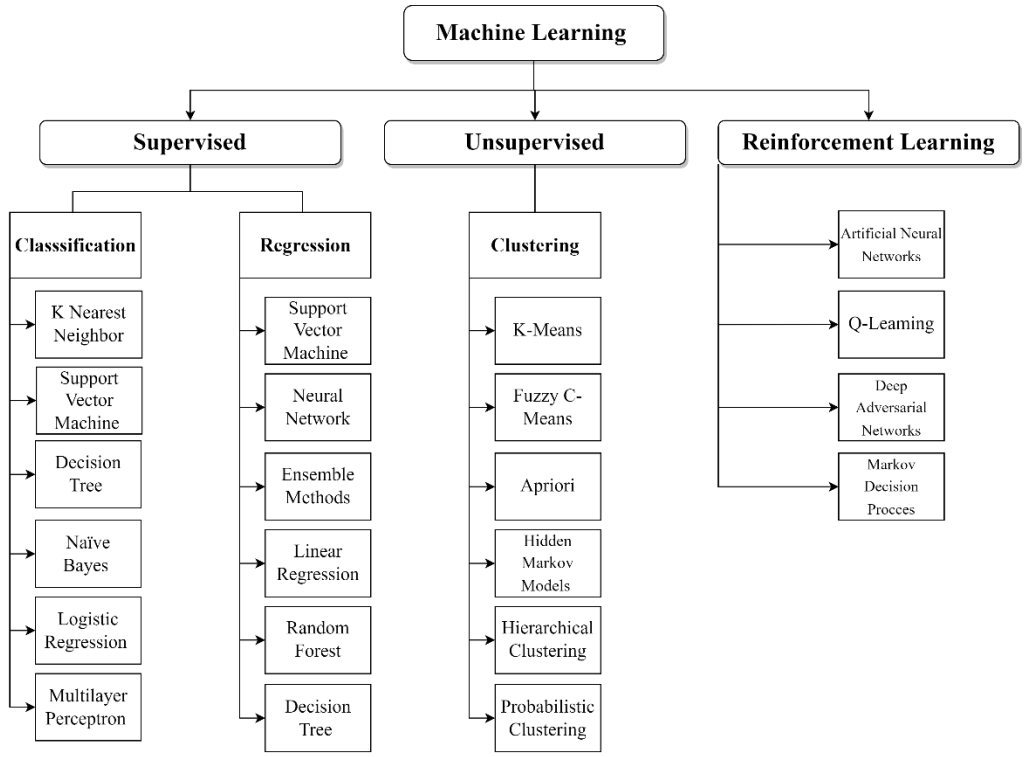

Figure 4.1. There are three main categories of ML methods.

In the first sort of learning for ML models, known as supervised learning, the models are trained on data whose outputs have already been defined. This means that the model is trained on both inputs (features) and outputs (targeting) so that it can predict future outputs from future inputs [37].

The second sort of learning is known as unsupervised learning, and it involves training algorithms by providing them with data without defining the outputs (targeting). In the process of being trained, models construct the linkages and patterns that they will later utilize to generate predictions based on new data [37].

The third sort of learning, may be broken down into three separate categories: observational learning, learning via modeling, and learning through reinforcement, while some book accept that it should be broken down into 3 categories and some

book not accept [38]. In this model, an agent investigates its surroundings in order to accomplish a predetermined objective. While investigating its environment, it comes to certain conclusions and takes some judgments. If the agent's choice brings closer to his objective, then the agent will earn a positive reward. Otherwise, the agent will receive a negative recompense. To put it another technique, this strategy might be seen of as one that relies on trial and error [41].

Supervised learning that was examined in this study was classification, with the goal of predicting the probability of thyroid disease. Thyroid status (normal, hyperthyroid, hypothyroid) is the classification target for the ML models.

4.1. K-NEAREST-NEIGHBOR MODEL

The KNN method is one of the most well-known classification algorithms. It is used to predict the class of a sample with an unknown class based on the classes of the records that are situated in its immediate neighborhood. This is done by taking into account the proximity of the records to the sample. The algorithm consists of the three stages listed below [42]:

- Determining the distance traveled by the input record using all of the training records.
- Organizing the training record according to the distance, and choosing the KNN for each record.
- We select as the one to make use of the class that possesses the greatest proportion of the resources held by its k closest neighbors (this method considers the class as the class of input record which is observed more than all the other classes among the KNN) as shown in Figure 4.2. The class label of the new record may be predicted by using a distance criterion in this space like the Euclidean distance, Manhattan and so on, in conjunction with the class labels of the records that are neighboring it, if those records have *n* features. The formula of the Euclidian distance is shown in Eq. 4.1 [38].

$$d(x, x') = \sqrt{(x_1 - x_1')^2 + (x_2 - x_2')^2} \quad (4.1)$$

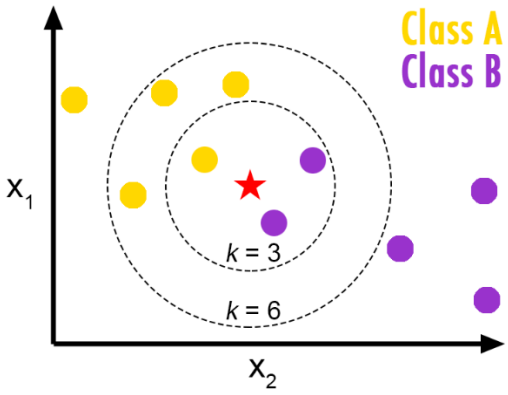

Figure 4.2. Select best K of KNN method [43].

In general, KNN looks for similar records among the set of training records in order to make a prediction about a new record class. This is how it determines which records are most likely to be similar to the new record. The classifier uses the records' relative proximity to one another as a measure of how close they are, and it chooses the records that are the most similar to one another.

4.1.1. Advantages of KNN Model

- It is a basic method to understand.
- It is a versatile instrument that may be used for classification and regression.
- It has a high ACC level.
- There is no need to formulate any additional data assumptions, hone down on a number of parameters, or develop a model.
- It does not need a significant amount of time to put into effect, which makes it particularly useful when working with nonlinear data [44,45].

4.1.2. Disadvantages of KNN Model

- The reliability of the data has a significant role in its outcome.
- If the data set is extensive, it might take a significant amount of time.
- Sensitive to data volume and irrelevant aspects.
- Because it is necessary to retain all of the training data, it requires a huge memory.

- Because it saves every training instance, it requires a significant amount of processing power [44,45].

4.2. SUPPORT VECTOR MACHINE MODEL

It was created in middle of the 1990s by Vapnik [46], and it is one of the most successful algorithms for supervised ML. The statistical learning theory provided the basis for its creation. The approach known as SVM may be used for both classification and prediction purposes. Classification is the application that makes the most use of it since it is one of the ML classification strategies that is the most effective [47]. In order to finish the classification, the input space of the dataset is linearly or non-linearly partitioned [48]. As illustrated in Figure 4.3, this is accomplished via defining the hyperplane in a vector space that has N-dimensions and making a distinction between two different categories of items. There is a possibility that there is more than one hyperplane dividing the two classes; nonetheless, in this scenario, the hyperplane with the greatest margin distance is selected. This is due to the fact that larger margins result in more accurate test sample predictions. The locations that are geographically closest to the decision border are identified as support vector, and the position of the decision boundary may be affected by these vectors [48].

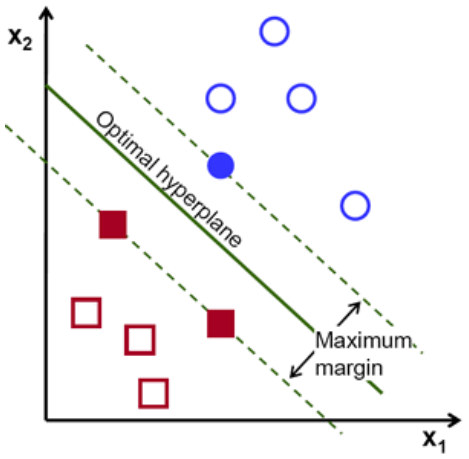

Figure 4.3. Optimal Hyperplane of SVM [49].

4.2.1. Advantages of SVM Model

- It works well in spaces with many dimensions.
- The method can still be used even when there are more dimensions than samples.
- It utilizes less memory since just a subclass of the training points, known as support vectors, are utilized by the decision function [47,48].

4.2.2. Disadvantages of SVM Model

- Overfitting must be avoided at all costs when selecting kernel functions and regularization terms if the number of attributes is much more than the number of examples.
- SVM can usually be highly time consuming [47,48].

4.3. DECISION TREE MODEL

For mutually classification and regression tasks, it is a crucial supervised ML algorithm. As can be seen in Figure 4.4, the DT takes the form of a tree-like flowchart in which the data is split constantly in accordance with a certain parameter. Refers to two units, the decision nodes, used to test a certain feature, and the papers that refer to the outcome of this test, as well known as the "root" node at the upper of the decision tree [50]. Different DT algorithms such as ID3, C4.5, C5, CART use different mathematical methods to partition training data for classification and regression [51].

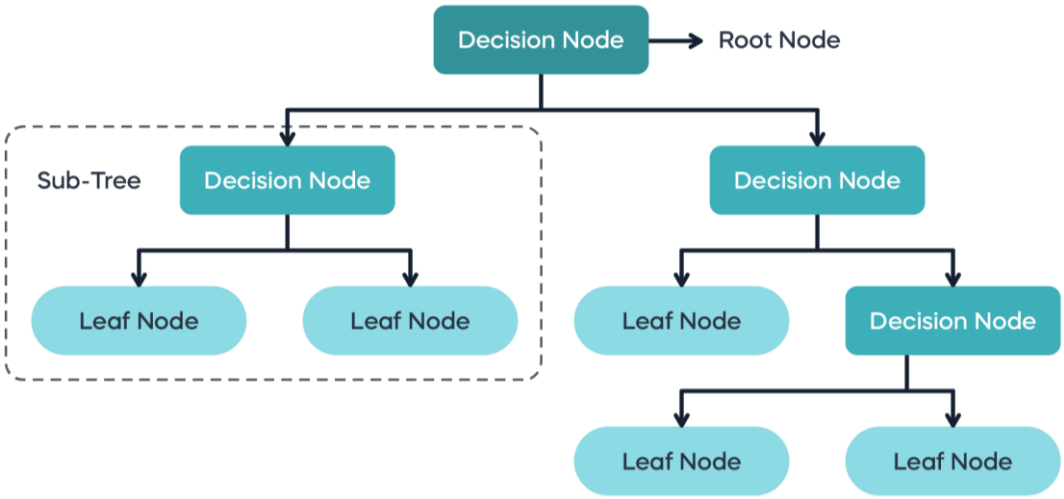

Figure 4.4. The structure of the Decision Tree Model [52].

The CART technique is applicable to both regression and classification issues. Its method produces one of these trees based on the type of reliable variable. A classification tree is constructed if a mutable is categorical. a regression tree is constructed if a variable is numeric [51].

The trees that we produce are trees of classification [53]. These are the steps that make up the CART algorithm: A CART Algorithm is built from the top down, with the partitioning initial at the root node and moving down through the levels as measurements are used to determine the best possible division. Gini impurity, information gain, variance reduction, and other similar measures are used to compare the effectiveness of various DT algorithms. The CART method employs the Gini Impurity. How many times an element in a subset has been incorrectly identified is tallied (for example, if someone suffering from hypothyroid is labelled as a normal thyroid). The randomness of the label's dispersion is taken into account throughout this tagging procedure [54]. The following Eq. 4.2 can be used to evaluate its effectiveness:

$$G = 1 - \sum_{j=1}^{a} (Pi)^2 \tag{4.2}$$

Where, G is Gini impurity metrics, j is an integer between one and a, and Pi is the proportion of items in class j.

4.3.1. Advantages of DT Model

- The DT approach does not require an excessive cost to construct a tree.
- It is compatible with numeric and category data.
- DT method does not need considerable data preprocessing.
- It functions effectively with binary and multiple predictions.
- The performance of the algorithm may be evaluated using statistical measures [52,53,54].

4.3.2. Disadvantages of DT Model

- Overfitting is a common concern in DT. Pruning, computing the minimum number of specimens needed in a leaf node, and measuring the depth of the tree could reduce this challenge.
- Outliers make decision trees unstable. Using ensemble DT solves this problem.
- DT predictions are neither smooth or continuous.
- XOR and equivalency difficulties are difficult to describe in DT.
- Unbalanced data categories might lead to biased trees [52,53,54].

4.4. NAÏVE BAYES MODEL

In the field of data mining, NB is among the most well-known classification algorithms [55]. Using the class as a starting point, it concludes the likelihood that a new instance fit into the class by treating each property as independent of the others [56]. The requirement for estimation multivariate possibilities from training data drives this assumption. Most possible permutations of feature values are either not present or insufficiently represented in the data used for training. Because of this, it is clear that attempting to directly estimate each relevant multi-variate probability is doomed to fail. The conditional independence assumption of NB allows it to avoid this problem. However, despite this strong independence condition, NB is a highly

effective classifier in many practical contexts [57]. In Figure 4.5 and Eq. 4.3 the Gaussian-NB algorithm is illustrated.

$$P\ A|B\ = \frac{P(B|A)P(A)}{P(B)} \tag{4.3}$$

Where P A|B is the conditional probability's final probability, P(A) is the class's prior probability, P(B|A) is the probability of the predictor's assumed class, and P(B) is the predictor's prior probability.

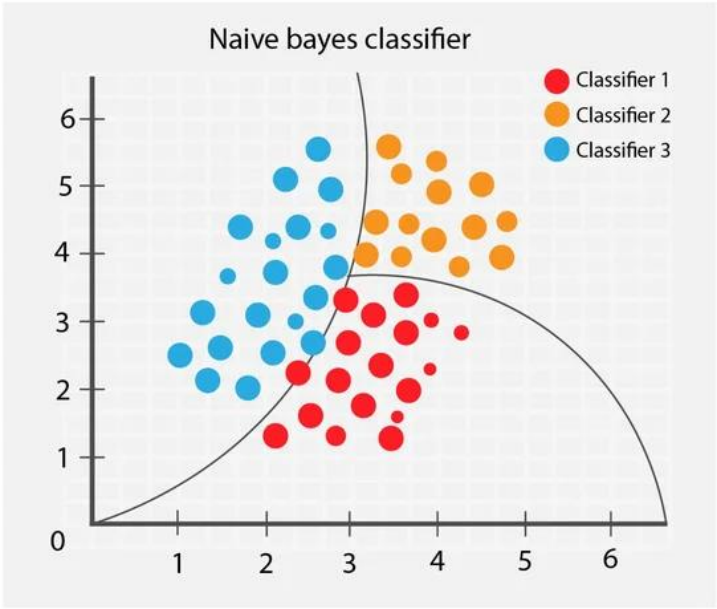

Figure 4.5. Gaussian NB algorithm [58].

4.4.1. Advantages of NB Model

- It can be implemented quickly.
- It is a quick and easy technique that yields good results.
- It is Easily trainable with minimal input data.
- It can manage a large dataset with ease.
- It is simple to grasp and construct [55-57].

4.4.2. Disadvantages of NB Model

- When no training tuples exist for a particular class, the posterior probability is 0. In this case, the model is incapable of making any predictions. This problem is called the Zero Probability.
- It is practically impossible to get a set of completely independent predictors in practice [55-57].

4.5. LOGISTIC REGRESSION MODEL

One of the most popular ML models is LR, which is applied in Supervised Learning [59]. In statistics, it refers to a technique for calculating an unknown categorical dependent variable based on known explanatory variables [60]. Predicting the impact of a categorical dependent variable can be done with the help of logical regression. Consequently, the final result must be a single absolute number. As shown in Figure 4.6, it returns probabilistic values between 0 and 1 rather than labels like yes or no, 0 or 1, true or false, etc. LR and Linear Regression are comparable in their use. LR is used to handle classification difficulties, while linear regression is utilized to tackle regression problems [61]. The Eq. 4.4 explain logistic regression equation.

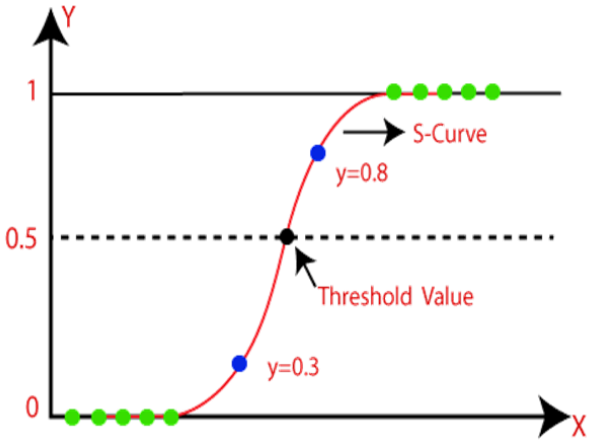

Figure 4.6. Logistic Regression curve [62].

$$p(X) = 1/1 + e^{-(a+bX)} \tag{4.4}$$

Where, P is the probability, e is the natural log base (approximately 2.718) the values of the model parameters are a and b. The value of a determines P when X is equal to

zero. The value of b modifies the rate at which the probability varies in response to a change of one unit in X. (similar to regular linear regression, LR allows for both standardized and unstandardized b weights to be used).

4.5.1. Advantages of LR Model

- It requires no resizing of features. Each observation is assigned a likelihood score via LR.
- Due to its simplicity and efficiency, it requires little computational resources.
- It is easy to execute and analyze, and is frequently utilized by data analysts and scientists [60,63].

4.5.2. Disadvantages of LR Model

- It cannot tackle nonlinear problems; hence, nonlinear characteristics must be converted.
- Independent variables that are correlated or very similar to one another but unrelated to the target variable will not perform well in LR.
- It cannot manage several category characteristics.
- It is susceptible to being overfitted [60,63].

4.6. MULTILAYER PERCEPTRON MODEL

The term "MLP" refers to a type of forward artificial neural network that is constructed using a sequence of outputs in addition to a number of inputs (MLP). A vector graph is created by multiple layers of input nodes that are inserted in between each of the MLP inputs and outputs layer as shown in Figure 4.7. When training the network, MLP makes advantage of backpropagation to do so. The multilayer perceptual notion is a neural network that links multiple layer in the form of a focused graph [64]. The signal only travels in one path between the nodes of the network. All of the nodes, with the allowance of the input’s node, have a nonlinear activation function. MLP make use of a supervised learning method called

backpropagation in their training. MLP is a method that is applied frequently and can be found being used in supervised learning difficulties, computational environmental science, and similar distributed processing research. Some of the applications include automatic translation, recognition of both speech and images, and recognition of text [64].

The term "MLP" comes from a single classifier that is known as a "perceptron." A perceptron is made up of a solo neuron and linearly classifies the input. The following Eq. 4.5 will describe how the bias is applied to the input, which is a vector that has a particular weight multiplied on it [64].

$$Y = (X * W) + B \tag{4.5}$$

Where, *Y* represents the output, *W* characterizes the weight, *X* signifies the input, and *B* represents the bias.

The main constraint of the perceptron model that relates to linear classification is overcome by the MLP algorithm, which also results in more complicated functions. After being vectorized, the input data is next sent into the primary layer. There, it is multiplied through weights that have been initially initialized at random, and then some biases are added to the product. At long last, an activation function is used on the whole thing to produce the desired effect. The output is then sent on to the following layer, which then repeats the process, with the exception of the first layer, which receives its input data from the layer directly below it. After the final layer is reached, the loss function can be computed, as demonstrated by the Eq. 4.6, which can be found below [66].

$$Loss(\hat{y}, y, W) = -yln\ \hat{y} - (1 - y)ln\ (1 - \hat{y}) + \alpha||W||_2^2 \tag{4.6}$$

Where, $\alpha||W||_2^2$ is a term for the regularization of L2, α controls the severity of the penalty and is a non-negative hyperparameter that is used to determine this, W is the

weights of the layers that come before and after the hidden layer, respectively, $\hat{y}$ is the intended destination of the sample, y is target of predicting.

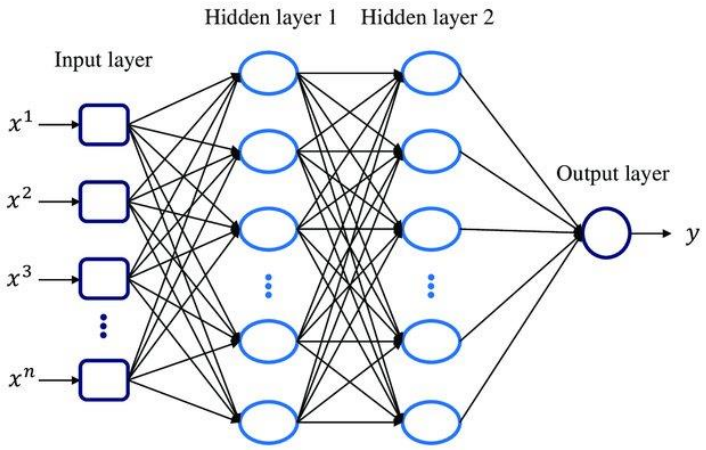

Figure 4.7. Layers of Multilayer Perceptron technique [65].

4.6.1. Advantages of MLP Model:

- Can be utilized in the solution of difficult non-linear issues.
- Offers prompt predictions after training has been completed.
- Even with fewer data points, it is possible to achieve the same ACC ratio [64,66].

4.6.2. Disadvantages of MLP Model:

- The effectiveness of the model is straight correlated to the quality of the training.
- When utilizing MLP as hyperparameters, it is necessary to adjust the number of hiding neurons, layers, and iterations.
- It's unclear how much influence the dependent variable has on each independent variable. Computations are complex and time-consuming [64,66].

4.7. RANDOM FOREST MODEL

is a supervised ML model, and can be used for mutually classification and regression. RF is an ensemble learner since it grows many decision trees instead of just one. This increases the number of trees, leading to a more robust classifier [67]. To categorize a new object, RF generates a large number of classifiers and then averages their output. The split is determined by RF's search through a random selection of input variables, which is followed by the production of multiple classification and regression (CART) trees, every one of which is learned on a bootstrap sample of the original datasets. To construct CARTs, data in the root node (which comprises the full learning sample) is repeatedly separated into child nodes [68]. As shown in Figure 4.8, the output of the classifier is decided by a majority vote from each tree in RF, with each tree casting a vote for input x.

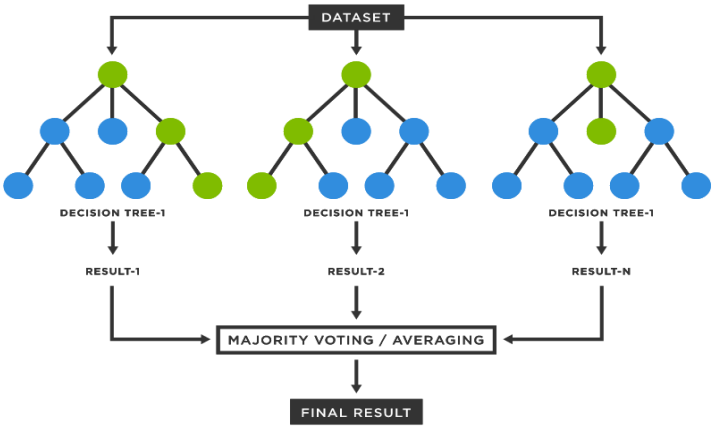

Figure 4.8. The foundation of the RF method [69].

The Gini index, or the formula in Eq. 4.7 used to choose how nodes on a decision tree branch, is often used when Random Forests are run based on classification data.

$$Gini = 1 - \sum_{i=1}^{C}(p_i)^2 \tag{4.7}$$

This formula habits the class and the probability to figure out the Gini of each branch on a node. This tells us which branch is more likely to happen. Here, pi shows how often the class you are looking at shows up in the dataset, and C shows how many classes there are.

Also use entropy in Eq. 4.8, to figure out how the nodes in a decision tree branch off from each other.

$$Entropy = \sum_{i=1}^{C} -p_i * \log_2(p_i) \quad (4.8)$$

Entropy looks at the chances of a certain result to decide which way the node should branch. The logarithmic function used to figure it out makes it harder to figure out than the Gini index.

4.7.1. Advantages of RF Model

- Highly adaptable and accurate to a surprising degree. Also, it continues to be accurate even if a lot of the data are missing.
- The quantity and magnitude of data sets it can process.
- Reduces the chance of running into a classifier that doesn't work well because of how the training and test data are related.
- It has a good way to fix mistakes in the datasets where the classes are not equal.
- It can figure out how much each classification feature is worth [67,68].

4.7.2. Disadvantages of RF Model

- Visually hard to understand and interpret.
- It is also more expensive when there are a lot of decision trees in the forest.
- Needs a lot of calculations, and the algorithm itself is not as heuristic.
- Much harder and takes more time to build than a decision tree [67,68].

4.8. EXTREME GRADIENT BOOSTING MODEL

Extreme Gradient Boosting, sometimes known as "XGBoost" for its shortened form, is a method of ML that makes use of both gradient boosting and decision tree approaches. Friedman designed the first iteration of the XGBoost algorithm [70] in the year 2002.

After that, two researchers at the University of Washington, Tianqi Chen and Carlos Guestrin, presented it as an article at (Special Interest Group Association for Information Discovery of Computing Machines and Data Mining) 2016 conference. The article got a lot of attention in the ML world [71]. XGBoost is a very popular algorithm, and in Kaggle competitions, it is usually the one that wins. Things like energy, money, health, etc. It has found a place in the field where it can be used, and compared to other algorithms, it is much faster and better at what it does. Also, XGBoost is 10 times faster than other algorithms and is good at making predictions. Also, XGBoost has a number of regularizations that make the system work better overall and stop it from overfitting or overlearning.

Gradient boosting is an ensemble method that uses boosting to create a strong classifier from a group of weak classifiers. Start with a basic learner and work your way up to a strong learner, as shown in Figure 4.9. The idea behind both gradient boosting and XGBoost is the same. The biggest difference between them is how they are put into action. In Eq. 4.9, XGBoost controls the complexity of trees by using different regularization techniques [72].

$$obj(0) = \sum_i^n l(y_i - \hat{y}_i) + \sum_{j=1}^{j} \Omega(f_j) \tag{4.9}$$

The goal aim of XGBoost is the entirety of a loss function that is applied to all predictions and a regularization function that is applied to all predictors. Together, these two functions are known as the regularization function (j trees). In the Eq. 4.10, fj stands for a guess from the *jth*-tree. Log loss is a popular metric that XGBoost uses. It is a probability-based metric that is used to measure how well a classification model works.

$$-\frac{1}{N}\sum_{i=1}^{N} y_i . \log\big(p(y_i)\big) + (1 - y_i) . \log(1 - p(y_i)) \tag{4.10}$$

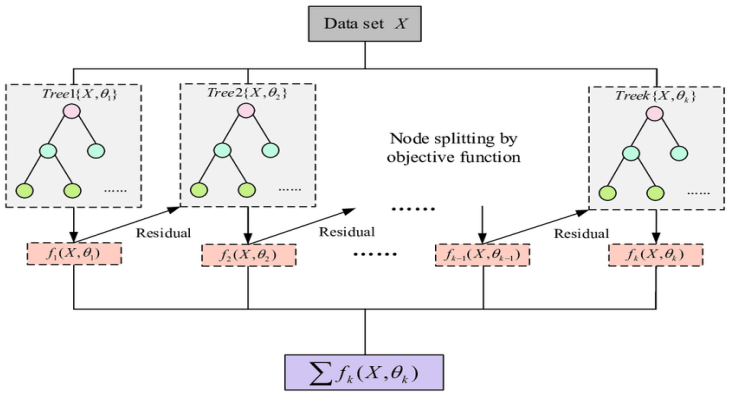

Figure 4.9. The flow chart of XGBoost method [73].

4.8.1. Advantages of XGBoost Model

- If the dataset is clean, it can stop overfitting.
- Can deal with the missing values.
- After each iteration, the user can run cross-validation.
- It Performs Acceptably in Cases Involving Small to Medium-Sized Datasets [70-72].

4.8.2. Disadvantages of XGBoost Model

- When used with dataset that is both sparse and unstructured, XGBoost does not perform very well.
- Gradient Boosting is very sensitive to outliers because each classifier has to fix the mistakes made by the learners that came before it.
- More complicated than other linear algorithms to understand [70-73].

4.9. MAJORITY OF VOTING MODEL

The term "majority voting" refers to the process by which we choose the class label that has been predicted by a greater number of classifiers, or that has obtained further than fifty percent of the votes. To be more precise, the term "majority vote" can only be used in contexts with binary class settings. However, it is not difficult to simplify the notion of majority voting to settings with many classes; this type of voting is

referred to as plurality voting. At this point, we choose the label for the class that garnered the most votes (mode). We begin by training m distinct classifiers, using the training set as our dataset source (C_1..,C_m…), as shown in Fig 4.10. The ensemble may be constructed using a variety of classification techniques, such as DT, SVM, LR classifiers, and so on, depending on the methodology. Alternately, we might apply the same basic classification algorithm to suit a variety of the training set's subsets. The random forest technique is a popular example of this strategy. This algorithm combines several separate decision tree classifiers into a single model [74].

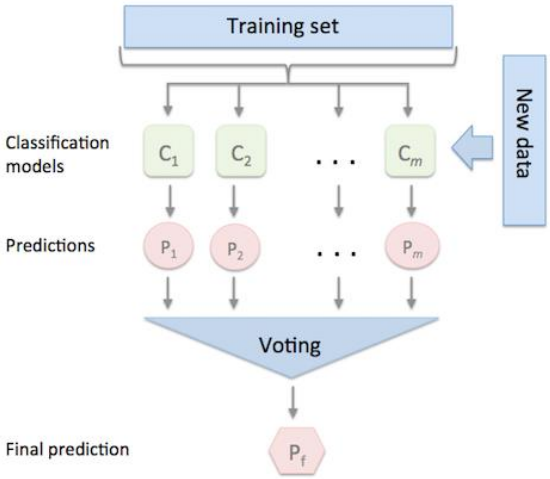

Figure 4.10. ensemble approach using majority voting [75]

The most basic example of majority voting is hard voting, as shown in Eq. 4.11. Here, we can guess the class label $\hat{\boldsymbol{y}}$ by looking at how each classifier $\boldsymbol{C_j}$ voted:

$$\hat{\boldsymbol{y}} = mode\,\{C_1(x), C_2(x), \ldots C_m(x)\} \tag{4.11}$$

In soft voting, we prediction the class labels based on the expected probabilities $\boldsymbol{p}$ for the classifier; however, this strategy is only suggested if the classifiers are well-calibrated as shown in Eq. 4.12.

$$\hat{\boldsymbol{y}} = argmax_i \sum_{j=1}^{m} w_j p_{ij} \tag{4.12}$$

where w_j is the weight that can be given to the j_{th} classifier.

4.10. STACKING MODEL

An alternative method of combining many classifiers is titled stacking, which is also identified as stacked generalization. is a type of EML strategy that transforms less capable learners into more capable individuals [76]. This integrated strategy utilizes the majority of higher-level models to combine lower-level models, hence boosting the classifier's capacity to predict the future. If we want better outcomes, we should use this method. In addition, the strategy seeks to minimize both bias and variance in the dataset in order to cut down on incorrect generalizations. The processing of the method is divided into two different layers, as shown in Fig 4.11. Multiple base models are qualified on the first training dataset at level 0 and the response variable is predicted for each model. After the results of level 0 have been combined from numerous base models to establish a single score (metamodel), that score is then applied to construct the output of level 1, which is the next stage in the process of training ensemble functions [77].

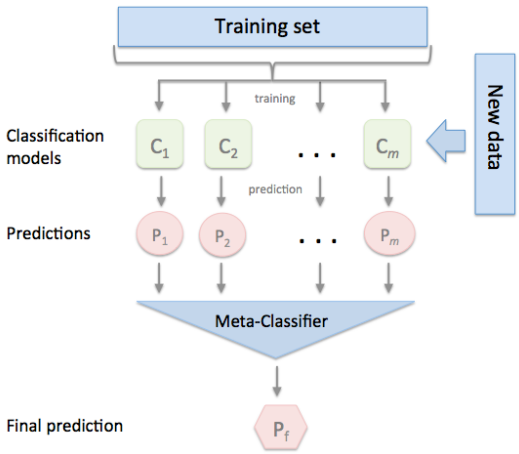

Figure 4.11. The flowchart of Stacking method [78].

4.10.1. Advantages of Stacking Model:

- Because of the structure of stacked ensembles, they often generate more robust prediction performance than standard individual models or average ensembles. In certain circumstances, little gains in prediction performance have a significant impact on the business scenario.

- Its Python Stacking Regressor and Stacking Classifier implementation is readily available through the Scikit Learn module [76,77].

4.10.2. Disadvantages of Stacking Model

- When utilizing no or low correlated base models, the improvement of stacking together models is just the most effective. The principle is similar to that of regular ensemble. A diversified model ensemble means more variability for the stacking model to optimize and achieve greater performance.
- One important drawback of employing the stacking approach is that it adds a lot of complexity to the final model, making it much more difficult to describe. As a result, businesses may not consider the implementation to be worthwhile because of the expense of interpretability.
- More complexity necessitates increased computing time. An overly sophisticated model will take years to execute as the number of dataset available rises rapidly. That makes little sense to organizations because the expenses are far higher than simply deploying a simple approach [76,77].

4.11. BAGGING MODEL

Breiman settled in 1994 a bagging technique, often identified as bootstrap aggregation [79]. It is one of the most basic and earliest ensemble ML approaches, and it works best with small training datasets. In this strategy, the dataset for training a group of unique models is randomly sampled with replacement using the bootstrap [80]. As shown in Figure 4.12, creating many copies of a predictor and then combining those copies into a single aggregated predictor is the goal of the technique known as "bagging predictor." When attempting to predict a numerical result, the aggregate uses an average of the several variants, whereas when predicting a class, it uses a plurality vote. Creation bootstrap replicas of the learning set and applying these as new learning sets outcomes in many kinds. Bagging can yield significant ACC increases in tests on actual and simulated datasets utilizing classification and regression trees, as well as subset selection in linear regression [79]. Bagging seeks

to mitigate the inconsistency of learning methods by mimicking the process using a predefined training set. Instead, then selecting a new, independent training dataset each time, the previous training dataset is updated by deleting certain instances and replicating others. In order to build a new one of the same size, instances are randomly picked from the original dataset. This sampling approach invariably duplicates some occurrences while deleting others. The Eq. 4.13 show the bagging formula.

$$S_L(.) = arg_k \max\left[card(l|w_l(.) = k)\right] \tag{4.13}$$

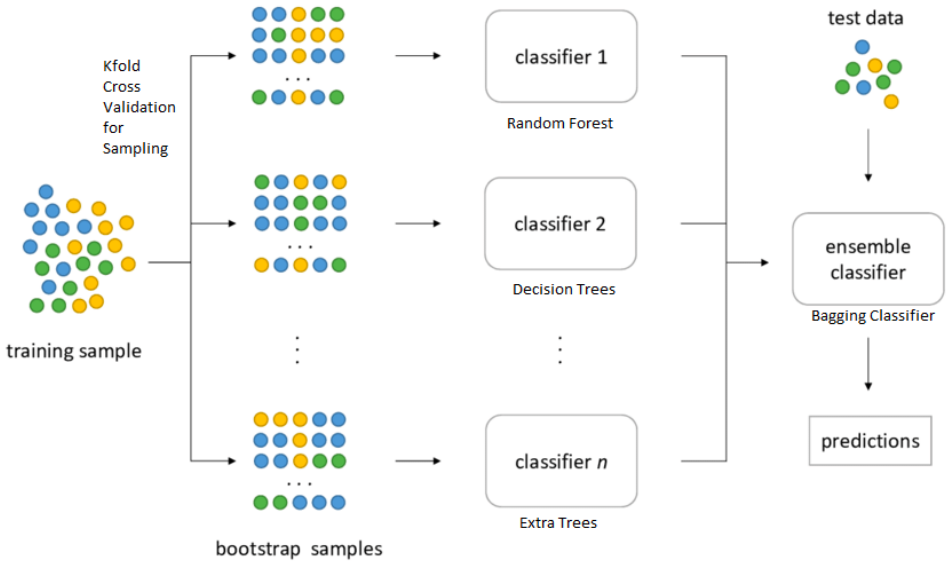

Figure 4.12. The flowchart of Bagging method [81].

4.11.1. Advantages of Bagging Model

- Bagging has the benefit of allowing numerous weak learners to work together to outperform a single good learner.
- It also improves in the decrease of variance, hence avoiding model overfitting in the method [79,80].

4.11.2. Disadvantages of Bagging Model

- that it reduces the interpretability of a model.
- When the right technique is not performed, the resulting model might have a lot of bias.
- Although highly accurate, bagging may be avoided due to its high processing cost in specific scenarios [79,80].

PART 5

MATERIALS

This Part presents the materials that were used in this thesis.

5.1. DATASET COLLECTION

We used dataset originally collected by Salman and Sonuç [29], who then used them to test and improve upon a variety of established methods. The thyroid function of 1,250 Iraqi boys and females aged 1 to 90 was recorded. All types of subjects, including those with hyperthyroidism, hypothyroidism, and normal thyroid function, were included. One to four months' worth of dataset was used to classify thyroid disorders using ML models. This dataset collection contains information on a wide range of variables, such as gender, age, T3, T4, and TSH concentrations. The 17 features in Table 5.1 collected dataset have been described in this thesis.

5.2. ETHICAL AUTHORIZATION

Salman and Sonuç are the source of the dataset [29]. With cross-sectional research, we were able to compile the dataset used in this analysis. The research was supervised by a medical expert and carried out at Al-kindi General Hospital and affiliated health center in Baghdad Governorate, Iraq. From (2020) September 1st to (2022) August 1st, this study was conducted. The prevalence of thyroid disease was investigated by a questionnaire comprised of a series of questions specifically created for that purpose. The hospital gave its approval to the studies after receiving written consent from the patient who had the test, authorization from the clinic doctors from whom they gathered dataset, and the patients' own participation.

Table 5.1. Description of original dataset.

No	Feature	Type	Range of Features
1	id	Numeral	(1,2, 3,…9999)
2	age	Numeral	(1,10,20, 50,…90)
3	Gender	1 or 0	1(Male),0(Female)
4	Query thyroxine	1 or 0	1 (Yes), 0 (No)
5	On-antithyroid- medication	1 or 0	1 (Yes), 0 (No)
6	Sick	1 or 0	1 (Yes), 0 (No)
7	Pregnant	1 or 0	1 (Yes), 0 (No)
8	Thyroid surgery	1 or 0	1 (Yes), 0 (No)
9	Query -hypothyroid	1 or 0	1 (Yes), 0 (No)
10	Query hyperthyroid	1 or 0	1 (Yes), 0 (No)
11	TSH-measured	1 or 0	1 (Yes), 0 (No)
12	TSH	Analysis Ratio	Numerical value
13	T3-measured	1 or 0	1 (Yes), 0 (No)
14	T3	Analysis Ratio	Numerical value
15	T4-measured	1 or 0	1 (Yes), 0 (No)
16	T4	Analysis Ratio	Numerical value
17	category	0 or 1 or 2	0(Normal), 1(Hypothyroid), 2(Hyperthyroid)

5.3. PLATFORM USED

The effectiveness of the software and hardware employed was crucial to the experimental study of this work. The hardware setup used in this thesis included a 2.60GHz Intel(R) Core (TM) i7-9750H CPU, a 256GB SSD (NVMe M.2), a 1TB (HDD), 16GB (of RAM), and an Nvidia GTX 1660 TI graphics processing unit. with the package description mentioned scikit-learn and Spyder Anaconda.

PART 6

METHODOLOGY

This chapter provides an illustration of the procedure of debating the proposed technique of work in this thesis, as seen in Figure 6.1. Methodology includes 4 stages such as Dataset collection, pre-processing, train-test splitting, cross-validation ML models and evaluation.

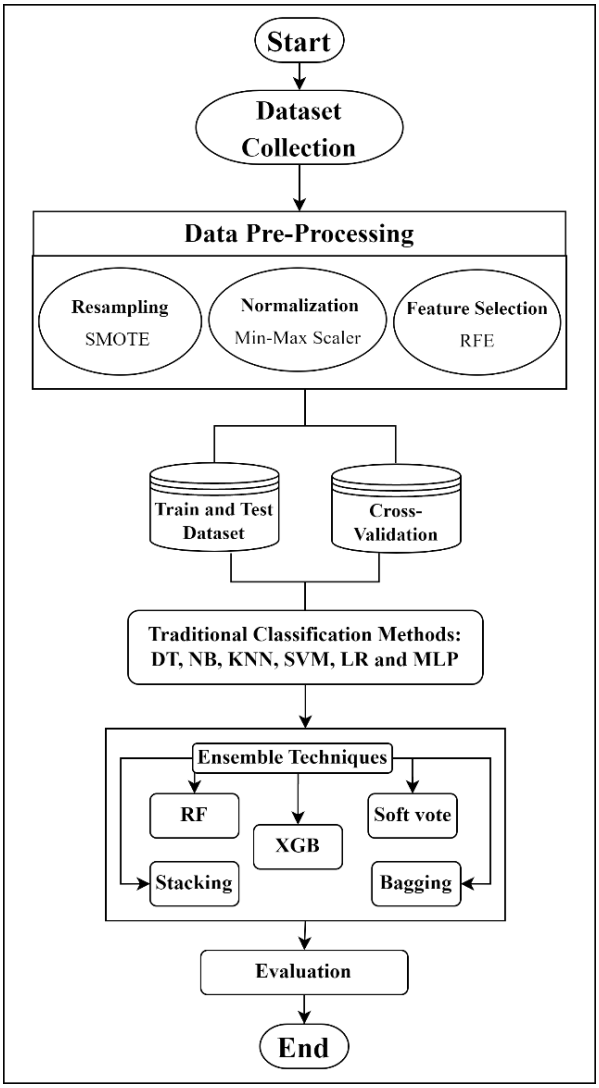

Figure 6.1. The flow chart of method.

In the pre-processing stage, the dataset is cleaned of missing values. Then look at the target, balance the dataset, normalize a dataset using Min-Max scaler and choose the best features by RFE technique.

Third, training ML models stage, classification methods are employed to predict thyroid illness. Six traditional ML models (K-NN, SVM, DT, NB, LR, MLP) are used and compared based on the performance measures between them. Then, the ensemble models (RF, XGboost, Soft Vote, Stacking, Bagging) are applied.

Fourth, in the evaluate stage, the metrics based on confusion matrix are evaluated for the models used in third stage. This is done to determine which classes and models have the highest performance.

6.1. DATA PRE-PROCESSING

Both data mining and ML depend critically on the preceding step of pre-processing dataset [82]. Due to the inherent inconsistencies and noise of real-world data, as well as the possibility of missing value, duplicate and irrelevant dataset, it can lead to inaccurately learned information and a decrease in algorithm performance. Preprocessing is used to get the dataset ready for the algorithms to utilize by cleaning it, scaling it, and transforming it into the right format. On top of that, in this model need to choose the most advantageous characteristics [82]. data pre-processing steps used in this thesis are shown in Figure 6.2.

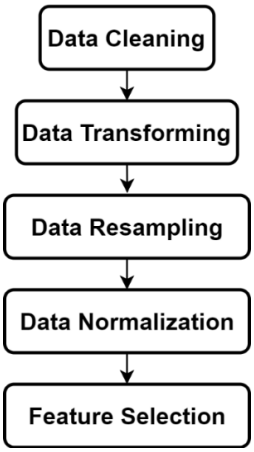

Figure 6.2. Pre-processing steps.

6.1.1. Data Cleaning

When an observation is missing data or any cell of a recorded row is null, for a particular variable, this is known as "missing data.". As a common occurrence, missing data can seriously compromise the conclusions that can be taken from the data. Although there are several options for addressing the problem of missing data, selecting the analytic method that would produce the least biased estimates is crucial. However, the approaches employed require a significant number of inputs. Therefore, missing values should be filled in. When researching the literature, there are a variety of ways [83]. The procedure is determined using dataset. If the features are relational, ML based dataset imputation algorithms are likely to succeed. However, ML-based approaches have comparatively high computing costs compared to basic statistical models. The two most widely used methods are deletion and mean/median imputation. This proposed study recognized a sample of missing data in which the missing values were detected, in dataset used in this thesis T3 and T4 were among the missing values.

6.1.2. Data Transforming

All categorical characteristics in the dataset are transformed into a numeric representation at this stage of the preprocessing procedures because dataset contains both category and numeric characteristics.

6.1.3. Data Resampling

In datasets that reflect the actual world, a certain class is frequently underrepresented in comparison to other classes. The "class imbalance" problem [84] (also known as the "curse of imbalanced datasets") is a challenge that arises from attempting to acquire a concept from a class that has a limited number of examples. This imbalance is the cause of the problem. The issue of class imbalance has been discussed in a variety of domains, such as telecommunication management, bioinformatics, fraud detection, and medical diagnostics [85]. Additionally, this issue has been placed among the top 10 challenges in data mining and pattern recognition [86]. Because the

majority of commonly used ML algorithms assume a balanced class distribution or an equal penalty of misclassification, imbalanced dataset severely hinders the learning process [87].

6.1.3.1. SMOTE Technique

In order to rebalance the dataset, the synthetic minority oversampling technique (SMOTE) algorithm utilizes a practice known as oversampling. Synthetic examples are at the heart of the SMOTE methodology, which means that the reproduction of minority class cases is not the primary focus [88]. The new information is derived by performing an interpolation between multiple minority class cases found within the defined neighborhood. Because of this, the approach is thought of as being concentrated on the "feature space" as opposed to the "data space". This means that the method is based on the values of the features and their link, as opposed to analyze the dataset points as a whole. It is necessary to do in-depth research on both the theoretical relationship between original and synthetic instances as well as the dimensionality of the dataset [88]. It is necessary to take into consideration certain characteristics, including the variance and correlation in the dataset and feature space as well as the relationship between the distributions of training and test samples [88].

6.1.4. Data Normalization

Adjusting the values of the numerical columns of the dataset to a comparable scale while maintaining their ranges. This technique is notable for preserving the integrity of such value ranges. The range of the dataset is narrowed down to a single range by the use of a linear data transformation known as min-max normalization. Within the scope of this investigation, the normalized dataset range was made between 0 and 1. In addition, classical scaling may struggle with sparse dataset due to the dense nature of the scaled dataset it generates [89,90].

6.1.5. Feature Selection

The approach of feature selection has been shown to be effective and efficient in the process of preparing dataset (especially high-dimensional dataset) for a wide range of data-mining and ML problems. The goal of feature selection is to simplify and clarify dataset, improve the effectiveness of data mining, and provide a good basis for classification models. The recent proliferation of big data has presented the problem of feature selection with substantial challenges as well as opportunities [91]. In the field of ML, feature selection strategies are employed to choose the best possible assortment of observable qualities that may be incorporated into the development of accurate models. It entails examining the link between each input factor and the target value using the evaluation criteria and identifying the variables having the strongest association. Feature selection is applied to improve decision precision, minimize the dimensions of the dataset, and speed up the process of ML training [92]. In this thesis the Recursive Feature Elimination (RFE) technique was used on the dataset.

6.1.5.1. Recursive Feature Elimination (RFE)

The RFE is a strategy for selecting features that seeks to estimate which characteristics are most useful for discriminating across classes of interest. It is able to discard any irrelevant characteristics in order to produce an input feature-set with the smallest feasible number of layers, without sacrificing classification ACC. The approach depends on variable significance evaluation, which is computed internally by RF classifiers and necessitates numerous classification rounds. Each round consists of learning a new RF classification model, evaluating its ACC, examining the feature of significance metrics for each feature utilized, and updating the feature-set that will be used in the following round. The initial round of categorization utilizes all accessible characteristics. Then, the lowest performers are identified using the significance metric variable calculated by the model during learning. Then, one (or more) of the weakest features are deleted from the dataset and the next stage of the operation is carried out. In addition, RFE seeks to reduce dependencies and collinearity in the input features [93].

6.2. IMPLEMENTING ML MODLES

The implementation of ML models used in this thesis is explained in detail in this section. Furthermore, modification of various parameters of the ML models have been explained. The dataset was divided into two parts before using ML models. 70% of the dataset was used for training and 30% for testing. Cross-validation is the typical ML evaluation technique. It was used in this model because it helps to identify overfitting or underfitting equipment. It involves dividing the dataset into several parts, and training the model on some of the parts. In this thesis, the number of k-fold is 10. Subsequently, ML models used in this work were used to predict thyroid disease.

This stage includes two steps, in the first step, the implementation using all the features, where the use of six traditional models for ML and five ensemble models to train the dataset. In the second step, the implementation using the RFE algorithm for feature selection, which selected the best 7 features as follows (Age, Gender, Pregnant, Thyroid surgery, TSH, T3, and T4). Finally, investigate six performance metrics across six traditional models and five ensemble models.

6.2.1. Implementing KNN Model

The KNN model was used to diagnose thyroid illness. KNN is one of the most basic classification models, relying mainly on the vote of KNN to classify dataset. In first step with all features, KNN was applied to the balanced dataset. The KNN parameters were set to {n*_neighbors=10, p=2, weights='distance'*}.

Where, *n_neighbors* is the quantity of neighbors of the class to be categorized and *p* is the Euclidean distance power parameter, *weight* points by the inverse of their distance. In this situation, neighbors closer to a query point will have a higher impact than those farther away.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to KNN again and adjusted to the same parameters.

6.2.2. Implementing SVM Model

SVM was used to predict thyroid disease. It is one of the most powerful classification models available. SVM has a number of elective parameters. In first step, with all feature SVM was applied to the balanced dataset. *(kernel='poly', degree=4, gamma='scale', coef0=3, shrinking=False, probability=True, random state=42)* were used to modify SVM parameters.

where, *kernel* is used to identify the kind of SVM model kernel, *degree* is the degree of the polynomial kernel function, *gamma* is the kernel coefficient, *coef0* is an independent term in the kernel function, *shrinking* is whether or not to use the shrinking heuristic, *probability* is whether or not to use probability estimates, and *random state* is utilized to regulate the production of random numbers for mixing data for probability estimations.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to SVM again and adjusted to the same parameters.

6.2.3. Implementing DT Model

In order to diagnose thyroid illness, a decision tree algorithm known as CART was employed. It is currently one of the most often used classification algorithms for illness diagnosis [99]. The DT possesses a group of parameters that are essential to the successful operation of the algorithm. For instance, not properly setting parameters such as (*max_features, max_leaf_nodes, max_depth, etc*.) might result in fully developed trees that are likely to be rather huge and contribute to overfitting of the model. Therefore, these parameters need to be tuned in order to increase the performance of the DT method, as well as to limit the amount of memory that is used and the likelihood of overfitting. In order to predict cases of thyroid illness, in first step with all features the DT has been implemented to the balanced dataset.

{max*_features=6, random_state=42, max_leaf_nodes=9, max_depth=9*}

Where, *max_features* are the number of features that should be considered when trying to find the optimal split, *random_state* is what you should use to regulate how random the estimator is, and so on. Create a tree with the maximum number of *leaf nodes* possible. The best nodes are characterized by a relative decrease in levels of impurities. If None is specified, there will be an unlimited number of leaf nodes, and the value of max depth will indicate the tree's *maximum depth.*

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to DT again and adjusted to the same parameters.

6.2.4. Implementing NB Model

For the purpose of predicting thyroid illness, the NB algorithm was applied. NB classifiers are constructed using Bayes' theorem. NB presupposes that the presence or absence of a feature inside a class is unrelated to any other feature within the class.

In first step with all features the balanced dataset was used to implement NB. {*priors=None, var_smoothing=1e-09*} were used as the default settings for the NB parameters.

Where, *priors* are the probabilities assigned to each class at the start of training, *var_smoothing* is the greatest variance component that has been smoothed into the computation.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to NB again and adjusted to the same parameters.

6.2.5. Implementing LR Model

To predict thyroid illness, LR algorithm was implemented. LR is one of the most used classification systems in the medical world, and it is used to diagnose disorders. The LR contains a number of optional parameters that are crucial to the model’s functionality. The most crucial parameters have been adjusted. In first step with all

features, LR parameters were tuned as: {*penalty='none', fit_intercept=True, class_weight ='balanced', solver='saga', random_state=2*}. Where, *penalty* is used to establish the penalization standard and *fit_intercept* is used to regulate if a constant need be introduced to the decision function, *solver* usable algorithm for the optimization problem, *random_state* is what you should use to regulate how random the estimator is.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to LR again and adjusted to the same parameters.

6.2.6. Implementing MLP Model

The MLP algorithm was utilized to predict Thyroid illness. Numerous possible MLP settings are important to the algorithm's functionality. Therefore, emphasis was placed on the most significant parameter, such as the quantity of neurons in the hidden layer, the maximum numbers of iterations, and the random numbers generator for weights. In step 1 with all features, MLP parameters were set to *{hidden_layer_sizes = (10,10,10), random_state = 1000, learning_rate = 'constant,' max_iter = 200, activation ='relu,' alpha = 0.1}.*

Where, *hidden_layer_sizes* represent the numeral of neurons in the hidden layer. *random_state* should be used to control the randomness of the estimator, etc. *learning_rate* weight update rate schedule. *max_iter* is the maximum iteration count. The *activation* function activates the hidden layer, *alpha* the L2 regularization term's strength. When the L2 regularization term is added to the loss, it is divided by the sample size.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to MLP again and adjusted to the same parameters.

6.2.7. Implementing RF Model

The RF algorithm was created in order to predict thyroid illness. It is among the most often used classification models. The RF contains several elective parameters that are essential for the model to function properly. Therefore, emphasis was placed on the most crucial parameters, including the quantity of estimators, maximum depth of tree, and random state. Since the purpose of RF is to generate a large number of separate trees, the predictions with the most votes are chosen first. Consequently, the quantity of trees, also known as the quantity of estimators, and the maximum trees depth are excellent parameters to adjust in order to enhance the performance of the RF method. The RF has been implemented for thyroid disease prediction. In first step with all features, The RF parameters were set as follows: {*max_depth = 9, n_estimators = 2, criteria = 'gini,', min_samples_split = 2, min_samples_leaf = 1, random_state = 10}*

Where, *max_depth* is the maximum tree depth, *n_estimators* are the total number of trees in the forest, and *criteria* is a role that measures the value of a split. *min_samples_split* is the minimal sample count necessary to divided an internal node. *min_samples_leaf* the minimum needed quantity of samples at a leaf node. A split point at any depth is only evaluated if it leaves at least min samples leaf training samples in each branch. This may result in a smoother model, particularly in regression. *random_state* regulates the unpredictability of the bootstrapping samples utilized in tree building.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to RF again and adjusted to the same parameters.

6.2.8. Implementing XGBoost Model

XGBoost has proven to be a highly successful method for boosting trees. XGBoost is based on the gradient-boosted machine concept proposed by (Friedman) [70]. The XGBRegressor model from Python's XGBoost module was applied to train the model using the same dataset. In first step with all features, the following XGBoost

parameters were specified: *{n_estimators = 100, max_depth = 6, learning_rate = 0.13, reg_lambda = 5, eval_metric = 'auc'}.*

Where, *max_depth* is defined as the extreme depth that restricts the number of tree nodes. *Learning_rate* reduces each tree's contribution. *Reg_lambda* is the L2 term for regularizing weights. By increasing this amount, the model will become more conservative. The User can add numerous evaluation metrics using the *eval_metric* property. Users of Python should remember to give metrics as a list of parameter pairs rather than a map, so that a subsequent *eval_metric* does not overwrite the prior one.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to XGBoost again and adjusted to the same parameters.

6.2.9. Implementing Soft Voting Model

The Voting classifiers consider the probabilities generated by each model and calculate an average. By employ three classifiers in estimator parameter {KNN, DT, NB}. In first step with all features, the following parameters for the voting classifier were specified: *estimators=model list, voting='soft,' and n jobs=-1.*

Where, *estimator* is going to be able to fit clones of the original estimators, which are going to be saved in the class. For *voting*, there are two options. If *voting* is set to “hard”, majority-rule class labels are utilized. If *voting* is set to” soft”, the class label is predicted using the argmax of the predicted probabilities. “soft” is suggested for a well-calibrated classifier ensemble. *n_jobs* is the number of jobs that can be completed simultaneously.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to Soft Voting again and adjusted to the same parameters.

6.2.10. Implementing Stacking Model

Is a kind of Ensemble Methods Learning that transforms weaker learners into strong ones. The processing of the strategy is separated into two levels. Numerous base models 1 to k are trained using the preliminary training dataset, and the response variable is predicted for each model at level 0. The output of level 0 is then used to construct the output of level 1 for training ensemble functions. In the first step, by employ four classifiers in estimator parameter {SVM, MLP, KNN, NB} and tuned as with all features to the balanced dataset:

SVM:{*kernel=''poly'', degree='4', gamma=''scale'', coef0='3', shrinking='False', probability='False', random_state='0'*}

Where, *kernel* is Specifies the kind of kernel to be utilized by the algorithm. *degree* is Polynomial kernel purpose degree ('poly'). *gamma* is Kernel coefficient. *coef0* is term independent to the kernel purpose. *Shrinking* is use the shrinking heuristic or not. *probability* is used to allow probability estimates or not. *random_state* is regulating the unpredictability and to control the randomness of the estimator.

MLP:*{activation = "relu", alpha =0.1,hidden_layer_sizes= (10,10,10),learning_rate = "constant", max_iter = 2000, random_state = 1000}*

Where, *activation* is activates the hidden layer. *Alpha* is strength of the L2 term for regularization. When applied to the loss, the L2 regularization term is split by the sample size. *hidden_layer_sizes* are representing the number of neurons in the hidden layer. *learning_rate* is weight update rate schedule. *Max_iter* is the maximum iteration count. *random_state* to control the randomness of the estimator.

KNN:*{ n_neighbors=6,weights='distance', algorithm='auto',p=2}*

Where, *n_neighbors* is the quantity of neighbors of the class to be categorized. *Weight* used in prediction. *algorithm* is Algorithm used to compute the near

neighbors will attempt to decide the most appropriate algorithm based on the values passed to fit method. and *p=2* is the Euclidean distance power parameter.

NB:*{ priors=None, var_smoothing=1e-09}*
Where, *priors* are the probabilities assigned to each class at the start of training, *var_smoothing* is the greatest variance component that has been smoothed into the computation.

After that, we implement the stacking classifier, adjust the parameters with the estimators above, and determine the final meta-class, which is LR.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to Stacking again and adjusted to the same parameters.

6.2.11. Implementing Bagging Model

Using basic learner such as DT, this is the easiest way to form an ensemble. Consequently, a bagged DT is comprised of trees that are individually trained on dataset bootstrapped from the input dataset. In first step with all features, the bagging parameter was set to *{random state =42 and n estimators = 100.}*

Where, *random_state* is used to regulate the randomness of the estimator. *N_estimators* represent the total number of trees.

In the second step, RFE was applied to the balanced dataset. After that, step 1 was applied to Bagging again and adjusted to the same parameters.

6.3. PERFORMANCE MEASUREMENT

In this thesis, a confusion matrix was utilized to depict the performance of the ML approaches in order to assess the performance of the classification models. In addition, the analysis was based on six different ML algorithms to measure

performance using the following metrics: ACC, precision, F1 score, sensitivity, specificity, and Matthew's correlation coefficient (MCC).

6.3.1. Confusion Matrix

confusion matrix is using to consolidate and check dataset. The four primary components shown in Figure 6.3, are true positive (TP), false positive (FP), false negative (FN), and true negative (TN) [94].

- TP signifies situations where detection is successful.
- FP refers to incorrectly detected conditions.
- FN represents situations that are incorrectly deemed undesirable.
- TN represents conditions that are assessed and determined to be undesirable.

Thyroid samples received three classes; class 0 "normal" as a negative. class 1 "hyperthyroidism" and class 2 "hypothyroidism" as a positive.

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Figure 6.3. Confusion Matrix [95].

6.3.2. Accuracy

ACC is one of the most often used metrics of classification performance, and it is defined as a ratio of properly categorized samples to a total number of samples [96]. The formula of ACC is given in Eq. 6.1.

$$Accuracy = (TN + TP)/(TN + FN + TP + FP) \quad (6.1)$$

6.3.3. Sensitivity

The true positive rate (TPR), hit rate or recall of a classifier is the ratio of properly categorized positive samples to the total number of positive samples, and it is calculated using Eq. 6.2 [96].

$$Sensitivity = TP/(FN + TP) \quad (6.2)$$

6.3.4. Specificity

True negative rate (TNR), or inverse recall is represented as the proportion of properly identified negative samples relative to the total number of negative samples, as shown in Eq. 6.3 [96].

$$Specificity = TN/(FP + TN) \quad (6.3)$$

6.3.5. Precision

Precision or positive predictive value is the ratio of true positives. Precision is determined by dividing the number of accurately positive predictions by the total number of positive predictions [97]. The formula of precision is given in Eq. 6.4.

$$Precision = TP/(FP + TP) \quad (6.4)$$

6.3.6. F1 Score

F1 Score called F1 measure is a measurement that combines Precision and Recall to create a unified evaluation. The formulation is shown in Eq. 6.5 and 6.6 as follows:

$$F1\ Score\ =\ 2TP/(2*TP+FN+FP) \tag{6.5}$$

Or

$$F1\ Score\ =\ 2*(Sensitivity*Precision/Sensitivity+Precision) \tag{6.6}$$

6.3.7. Matthew's correlation coefficient (MCC)

Brian W. Matthews established this metric in 1975 [98], and it quantifies the correlation between observed and expected classifications. It is computed directly from the confusion matrix, as shown in Eq. 6.7. A coefficient of 1 denotes a perfect predicted, -1 reflects absolute discrepancy between prediction and true values. This measure is susceptible to dataset imbalances.

$$MCC = \frac{TN*TP-FN*FP}{\sqrt{(TP+FP)(TP+FN)\ (TN+FP)(TN+FN)}} \tag{6.7}$$

PART 7

RESULTS & DISCUSSION

7.1. PREFACE

The six ML traditional models and the five Ensemble models were designed using the scikit-learn package which is a convenient package used to implement the above models and preprocessing stages. The dataset consists of 1250 samples with 957 normal samples without thyroid disorder, 142 samples with Hypothyroid, and 151 samples with Hyperthyroid. Due to the imbalance of dataset where there is a large confusion in the percentage of classes, we balanced the dataset using SMOTE algorithm as in Table 7.1. Finally, 70% of the dataset was used as a training set and 30% of the dataset for the test set.

Table 7.1. Using the smote algorithm to balance the thyroid dataset.

Dataset	**No. classes**	**Total records**	**Normal**	**Hypothyroid**	**Hyperthyroid**
Normal Dataset	3	1250	957	142	151
After SMOTE	**3**	**2871**	**957**	**957**	**957**

7.2. BASIC STATISTICS OF DATASET

Figure 7.1 shows the distribution of the number of unaffected people who suffer from Hyperthyroid and Hypothyroid. After balancing the dataset, Figure 7.2 shows that it became more balanced as the positive and negative samples are similar.

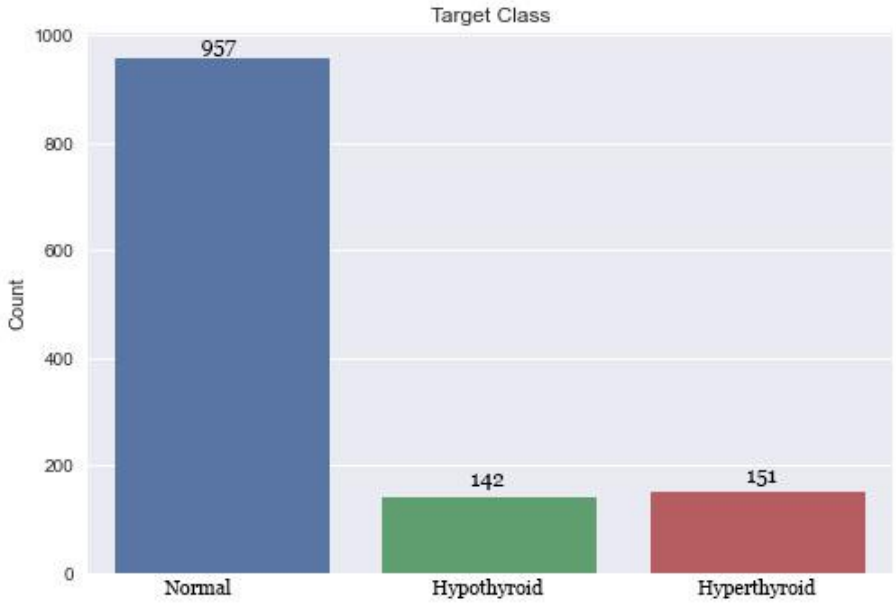

Figure 7.1. Histogram of distribution from non-patient to patient.

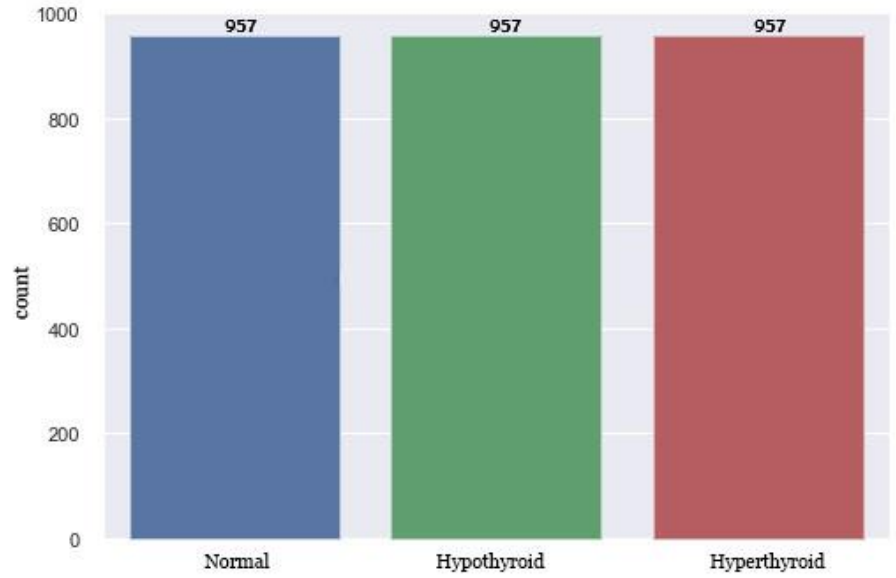

Figure 7.2. Histogram of distribution after balancing.

Figure 7.3 shows the gender distribution of thyroid patients in the samples. According to [100], the proportion of women with hypothyroidism and hyperthyroidism is much higher than that of men, and it can be distinguished that the samples correspond to the general distribution of the gender of thyroid patients.

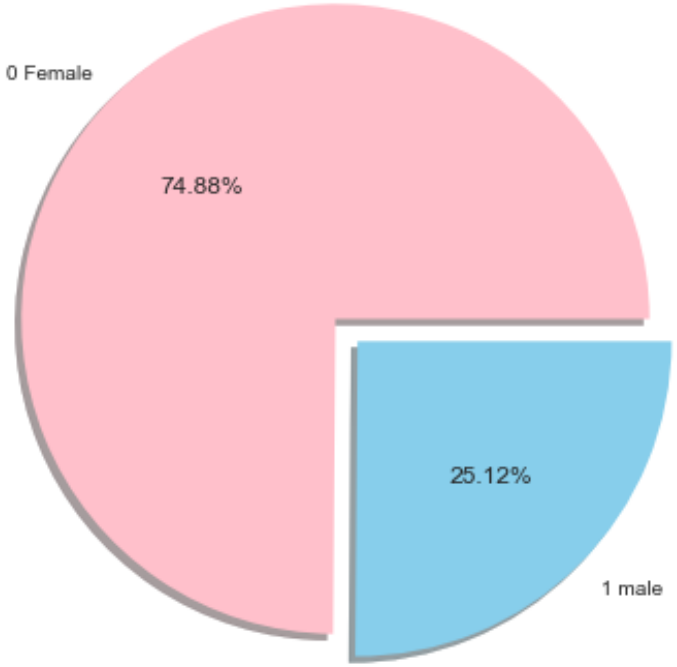

Figure 7.3. The Gender Distribution of Thyroid Patients.

7.3. EXPERIMENTAL RESULTS USING ALL FEATURES

Thyroid disease was predicted, which includes all the features {Age, Gender, Thyroxine Query, Thyroid Treatment, On-Antithyroid-Medication, Pregnant, Thyroid Surgery, Hypothyroid Query Thyroid, Query Hyperthyroidism, Measured-TSH, TSH, Measured-T3, T3, Measured-T4, T4}.

For traditional models, each model will be implemented and performance metrics calculated for each class. Table 7.2 shows the performance evaluation of the KNN model with all the features, Figure 7.4 and Figure 7.5 illustrate the confusion matrix and ROC curve when KNN models is applied to the balanced dataset with all features.

Table 7.2. Performance evaluation of KNN with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	92.11%	85.60%	95.44%	90.60%	88.00%	0.88
Hypothyroid	98.30%	89.60%	98.11%	96.14%	97.33%	0.97
Hyperthyroid	93.20%	91.50%	94.02%	88.80%	90.08%	0.90

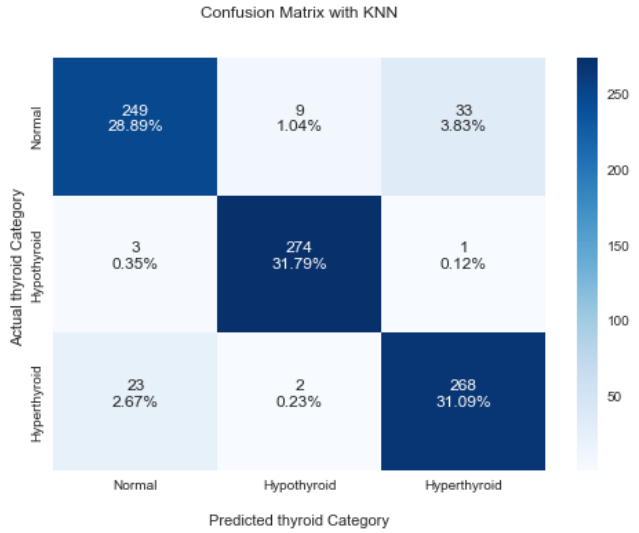

Figure 7.4. Confusion Matrix of KNN Model with all features.

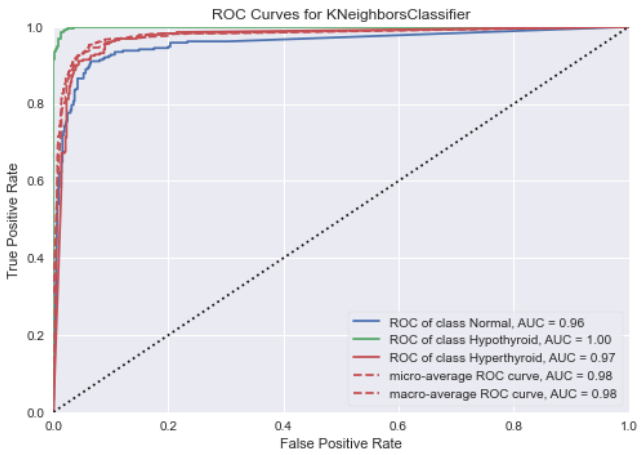

Figure 7.5. ROC Curve of KNN Model with all features.

Table 7.33 shows the performance evaluation of the SVM model with all the features, Figure 7.6 and Figure 7.7 illustrate the confusion matrix and ROC curve when SVM models is applied to the balanced dataset with all features.

Table 7.3.Performance evaluation of SVM with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	94.80%	86.60%	99.09%	98.05%	92.00%	0.88
Hypothyroid	99.29%	100%	99.00%	97.90%	99.00%	0.98
Hyperthyroid	95.60%	98.29%	94.19%	89.71%	93.90%	0.91

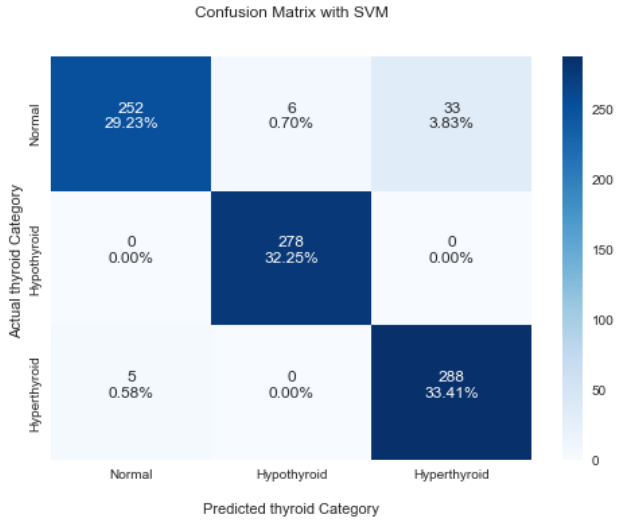

Figure 7.6. Confusion Matrix of SVM Model with all features.

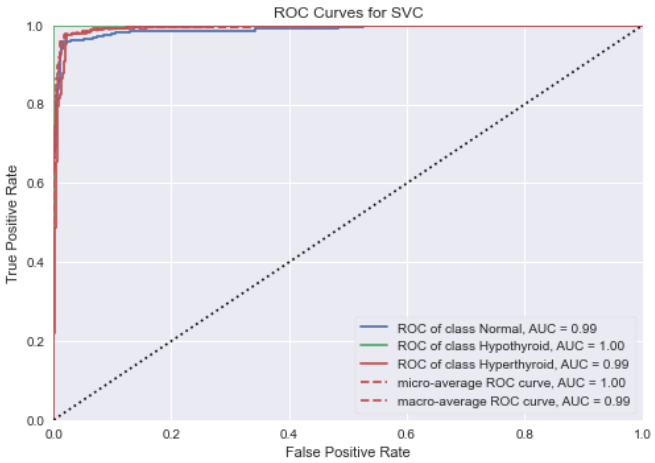

Figure 7.7. ROC Curve of SVM Model with all features.

Table 7.4 shows the performance evaluation of the DT model with all the features, Figure 7.8 and Figure 7.9 illustrates the confusion matrix and ROC curve when DT models is applied to the balanced dataset with all features.

Table 7.4. Performance evaluation of DT with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	99.90%	99.70%	100%	100%	99.82%	0.99
Hypothyroid	99.90%	100%	99.82%	99.64%	99.82%	0.99
Hyperthyroid	100%	100%	100%	100%	100%	1.00

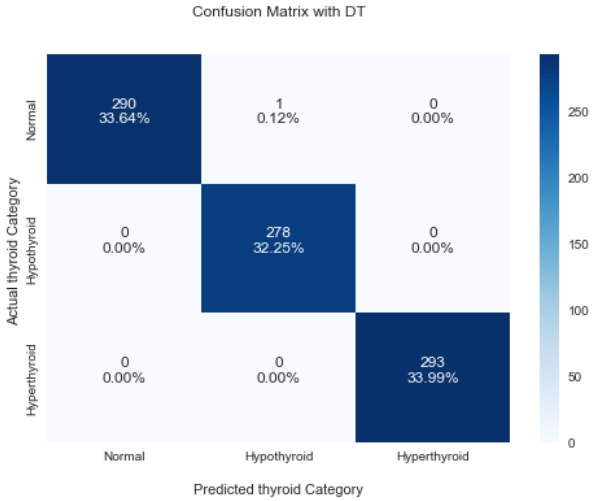

Figure 7.8. Confusion Matrix of DT Model with all features.

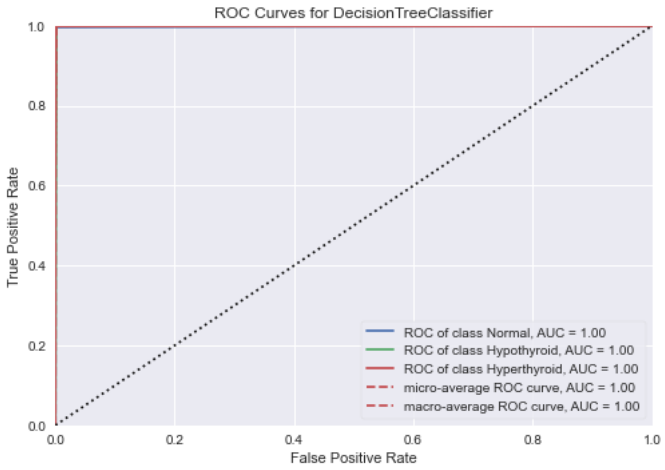

Figure 7.9. ROC Curve of DT Model with all features.

Table 7.5 shows the performance evaluation of the NB model with all the features, Figure 7.10 and Figure 7.11 illustrate the confusion matrix and ROC curve when NB models is applied to the balanced dataset with all features.

Table 7.5. Performance evaluation of NB with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	67.00%	5.84%	98.07%	60.71%	10.70%	0.10
Hypothyroid	74.50%	99.30%	66.60%	48.80%	65.40%	0.56
Hyperthyroid	96.90%	91.12%	99.82%	99.62%	95.20%	0.93

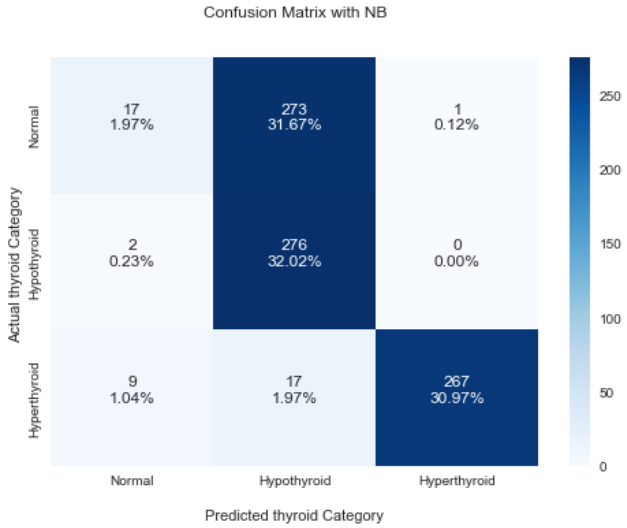

Figure 7.10. Confusion Matrix of NB Model with all features.

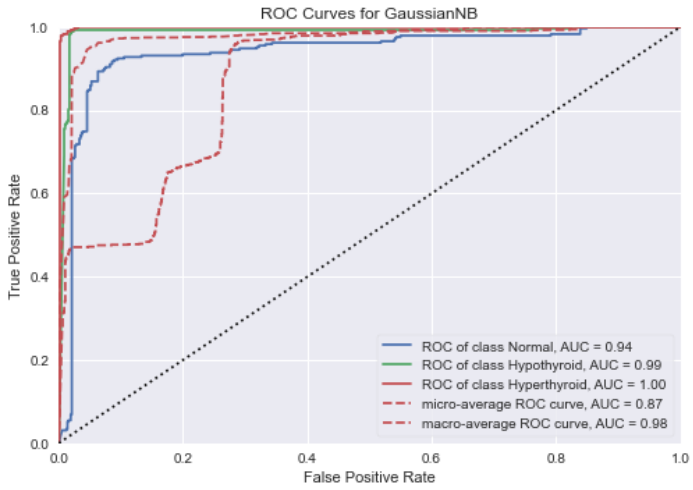

Figure 7.11. ROC Curve of NB Model with all features.

Table 7.6 shows the performance evaluation of the LR model with all the features, Figure 7.12 and Figure 7.13 illustrates the confusion matrix and ROC curve when LR models is applied to the balanced dataset with all features.

Table 7.6. Performance evaluation of LR with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	86.42%	82.13%	88.61%	78.61%	80.33%	0.70
Hypothyroid	99.30%	99.30%	99.31%	98.60%	98.92%	0.98
Hyperthyroid	87.12%	78.50%	91.60%	82.73%	80.60%	0.71

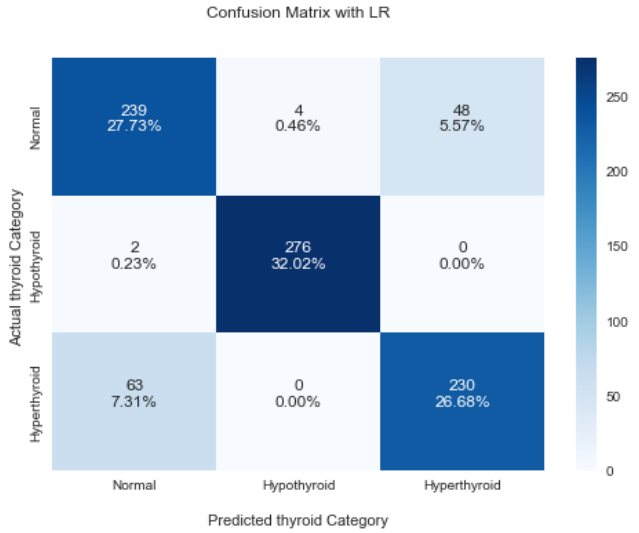

Figure 7.12. Confusion Matrix of LR Model with all features.

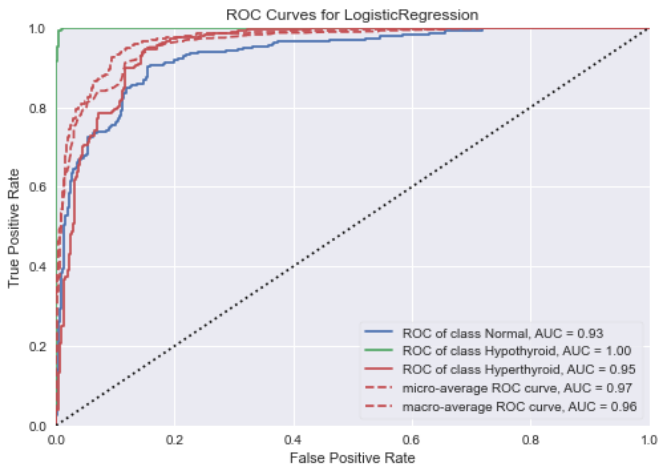

Figure 7.13. ROC Curve of LR Model with all features.

Table 7.7 shows the performance evaluation of the MLP model with all the features, Figure 7.14 and Figure 7.15 illustrates the confusion matrix and ROC curve when MLP models is applied to the balanced dataset with all features.

Table 7.7. Performance evaluation of MLP with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	95.93%	88.31%	99.82%	99.61%	93.62%	0.91
Hypothyroid	99.90%	100%	99.83%	99.64%	99.82%	0.99
Hyperthyroid	96.05%	99.70%	94.20%	89.84%	94.50%	0.92

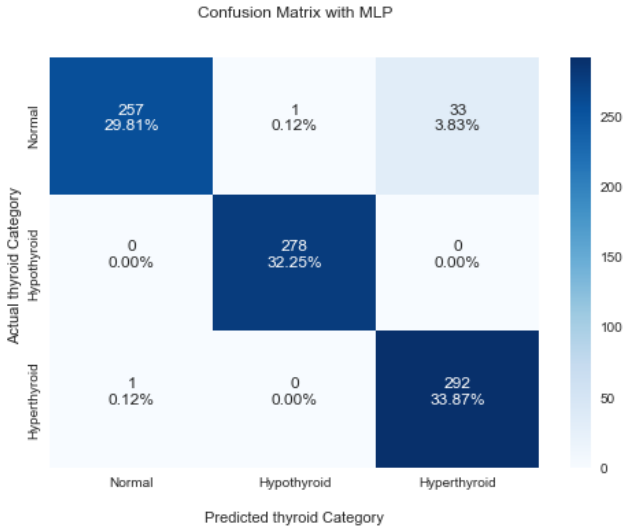

Figure 7.14. Confusion Matrix of MLP Model with all features.

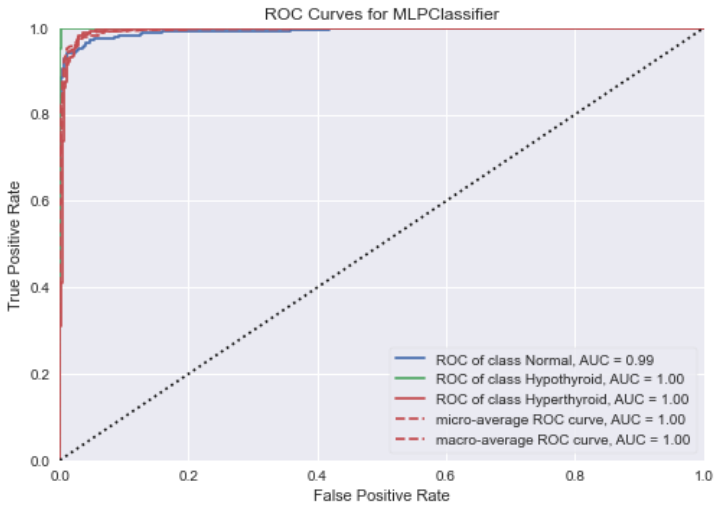

Figure 7.15. ROC Curve of MLP Model with all features.

The performance measures were calculated for each traditional models with all features above. The average model performances were shown in Table 7.8.

Table 7.8. Comparison based on the average performances for traditional models with all features.

Models	**KNN**	**SVM**	**DT**	**NB**	**LR**	**MLP**
ACC	94.50%	96.60%	99.92%	79.44%	91.00%	97.30%
Sensitivity	91.90%	95.00%	99.89%	65.41%	86.63%	96.00%
Specificity	95.90%	97.41%	99.94%	88.15%	93.16%	98.00%
Precision	91.81%	95.22%	99.89%	69.70%	86.64%	96.40%
F1 score	91.80%	94.90%	99.88%	57.08%	86.60%	96.00%
MCC	0.88	0.93	0.99	0.53	0.80	0.94

For Ensembles models, each model will be implemented and performance metrics calculated for each class. Table 7.9 shows the performance evaluation of the RF model with all the features, Figure 7.16 and Figure 7.17 illustrate the confusion matrix and ROC curve when RF models is applied to the balanced dataset with all features.

Table 7.9. Performance evaluation of RF with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	98.00%	97.25%	98.24%	96.60%	96.91%	0.95
Hypothyroid	98.25%	97.12%	98.80%	97.50%	97.29%	0.96
Hyperthyroid	98.72%	98.00%	99.12%	98.28%	98.11%	0.97

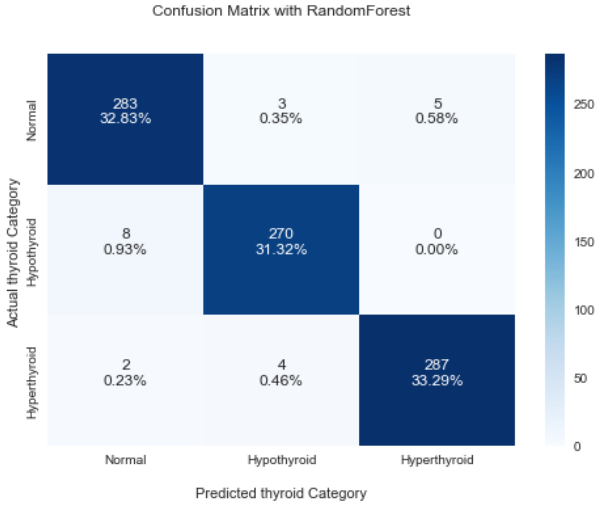

Figure 7.16. Confusion Matrix of RF Model with all features.

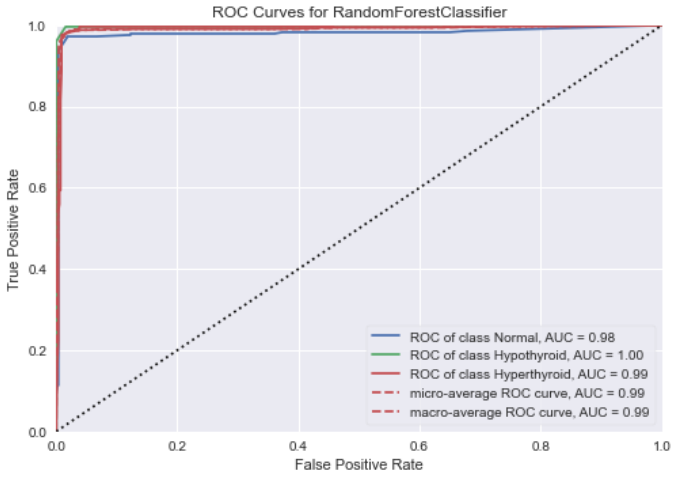

Figure 7.17. ROC Curve of RF Model with all features.

Table 7.10 shows the performance evaluation of the XGboost model with all the features, Figure 7.18 and Figure 7.19 illustrates the confusion matrix and ROC curve when XGboost models is applied to the balanced dataset with all features.

Table 7.10. Performance evaluation of XGboost with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	100%	100%	100%	100%	100%	1.0
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	100%	100%	100%	100%	100%	1.0

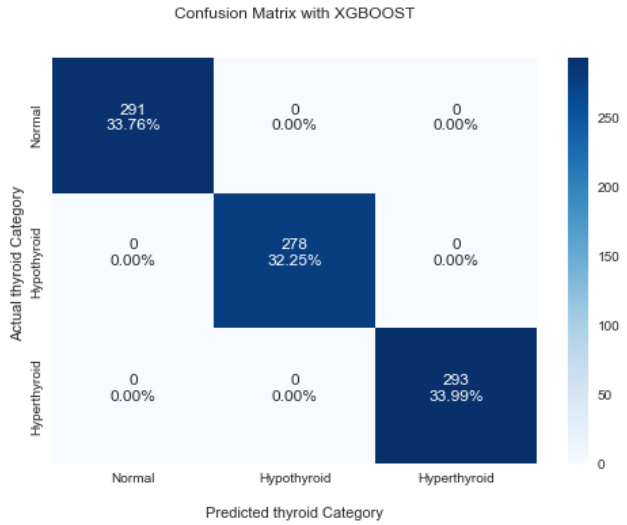

Figure 7.18. Confusion Matrix of XGBoost Model with all features.

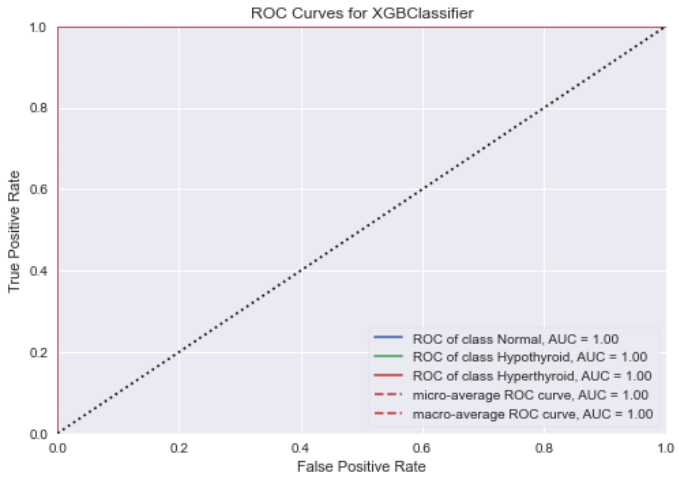

Figure 7.19. ROC Curve of XGBoost Model with all features.

Table 7.11 shows the performance evaluation of the Soft Vote model with all the features, Figure 7.20 and Figure 7.21 illustrates the confusion matrix and ROC curve when Soft Vote models is applied to the balanced dataset with all features.

Table 7.11. Performance evaluation of Soft Vote with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	98.37%	95.90%	99.64%	99.28%	97.60%	0.96
Hypothyroid	98.60%	100%	97.94%	95.90%	97.90%	0.97
Hyperthyroid	99.80%	99.31%	100%	100%	99.70%	0.99

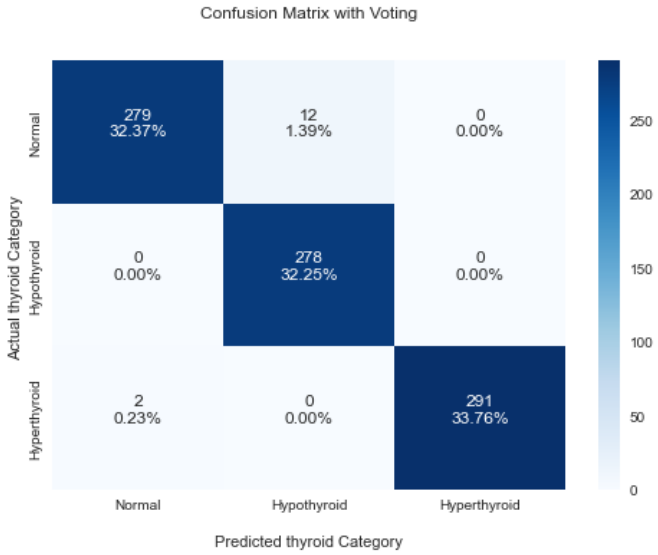

Figure 7.20. Confusion Matrix of Soft Voting Model with all features.

Figure 7.21. ROC Curve of Soft Voting Model with all features.

Table 7.12 shows the performance evaluation of the Stacking model with all the features, Figure 7.22 and Figure 7.23 illustrates the confusion matrix and ROC curve when Stacking models is applied to the balanced dataset with all features.

Table 7.12. Performance evaluation of Stacking with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	98.72%	97.25%	99.50%	99.00%	98.09%	0.98
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	98.72%	99.00%	98.60%	97.31%	98.13%	0.97

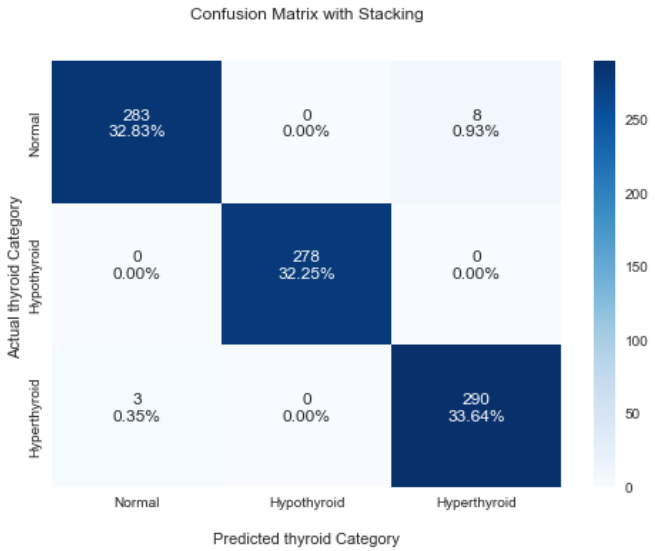

Figure 7.22. Confusion Matrix of Stacking Model with all features.

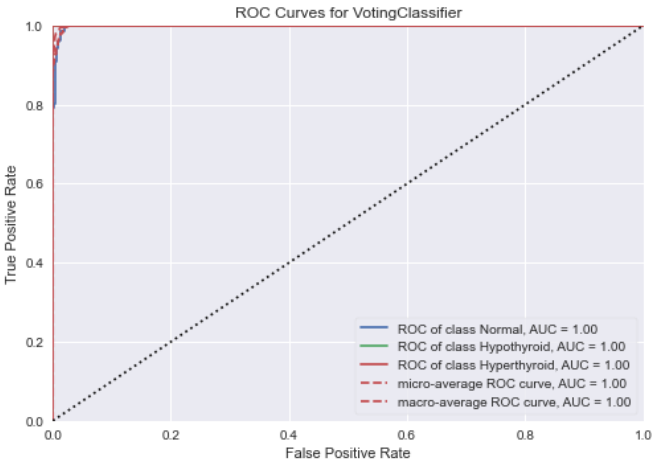

Figure 7.23. ROC Curve of Stacking Model with all features.

Table 7.13 shows the performance evaluation of the Bagging model with all the features, Figure 7.24 and Figure 7.25 illustrates the confusion matrix and ROC curve when Bagging models is applied to the balanced dataset with all features.

Table 7.13. Performance evaluation of Bagging with all features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	100%	100%	100%	100%	100%	1.0
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	100%	100%	100%	100%	100%	1.0

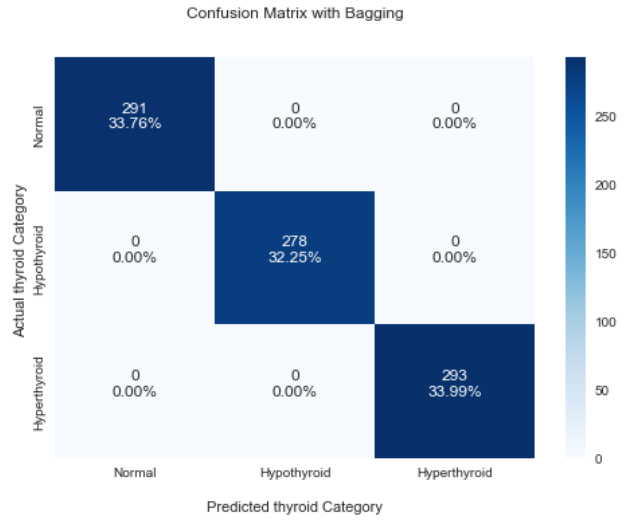

Figure 7.24. Confusion Matrix of Bagging Model with all features.

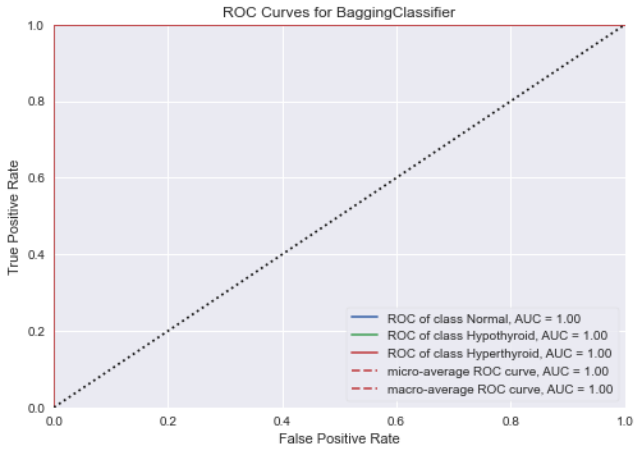

Figure 7.25. ROC Curve of Bagging Model with all features.

The performance measures were calculated for each ensemble models with all features above. The average model performances were shown in Table 7.14.

Table 7.14. Comparison based on the average performances for ensemble models with all features.

Models	**RF**	**XGBoost**	**Soft Vote**	**Stacking**	**Bagging**
ACC	98.30%	100%	98.91%	99.14%	100%
Sensitivity	97.44%	100 %	98.40%	98.74%	100 %
Specificity	98.72%	100 %	99.20%	99.40%	100%
Precision	97.44%	100 %	98.38%	98.50%	100 %
F1 score	97.44 %	100 %	98.40%	98.74%	100 %
MCC	0.96	1.0	0.98	0.98	1.0

7.3.1. Train Accuracy and Test Accuracy for All Features

In order to find out if there is an overfitting, one of the ways is to know the ACC of training to the ACC of the test, where there should not be a big difference between the ACC of the training and the ACC of the test. Where the ACC of the training is the ACC of the model in the examples on which it is based, and the ACC of the test is the ACC of the model in the examples that it did not see as shown in Table 7.15.

Table 7.15. Comparison between training ACC and test ACC using all features.

Models	**Train ACC**	**Test ACC**
KNN	100%	94.50%
SVM	95.62%	96.60%
DT	100%	99.92%
NB	66.9%	79.44%
LR	88.4%	91.00%
MLP	96.12%	97.30%
RF	98.51%	98.30%
XGBoost	100%	100%
Soft Vote	100%	98.91%
Stacking	99.85%	99.14%
Bagging	100%	100%

7.3.2. Time of Prediction for All Features

In this experiment, when predicting the same sample from the same test set, the difference between the training time and prediction time of six models of traditional models and five models using ensemble were recorded. The difference between the training time and prediction time of the method is the run-time of each model did run 10 times to decrease the impact of the computer cache. The results of the run-time comparison were shown in Table 7.16. The following is a list of the computer's setup that is used to execute the preceding classifiers: A Nvidia GTX 1660 TI GPU, an Intel(R) Core (TM) i7-9750H processor running at 2.60GHz, a 256GB SSD (NVMe M.2), a 1TB HDD, and 16GB of RAM. Windows 11 64-bit operating system is the system type.

Table 7.16. difference between the training time and prediction time for all features.

Models	Training Time (seconds)	Predict Time (seconds)
KNN	0.067	0.028
SVM	0.583	0.018
DT	0.005	0.001
NB	0.006	0.002
LR	0.086	0.005
MLP	2.209	0.003
RF	0.008	0.003
XGB	0.253	0.006
Soft Vote	3.643	0.033
Stacking	14.787	0.054
Bagging	0.401	0.015

7.4. EXPERIMENTAL RESULTS USING FEATURE SELECTION

Thyroid disease was predicted, using the RFE algorithm for feature selection. which includes the features {age, gender, pregnant, thyroid surgery, TSH, T3, T4}.

For traditional models, each model will be implemented and performance metrics calculated for each class. Table 7.17 shows the performance evaluation of the KNN model for selected features, Figure 7.26 and Figure 7.27 illustrates the confusion

matrix and ROC curve when KNN models is applied to the balanced dataset for selected features.

Table 7.17. Performance evaluation of KNN for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	92.53%	84.53%	97.02%	93.53%	88.80%	0.84
Hypothyroid	98.25%	98.60%	98.11%	96.14%	97.33%	0.96
Hyperthyroid	94.31%	95.22%	93.84%	88.90%	91.92%	0.88

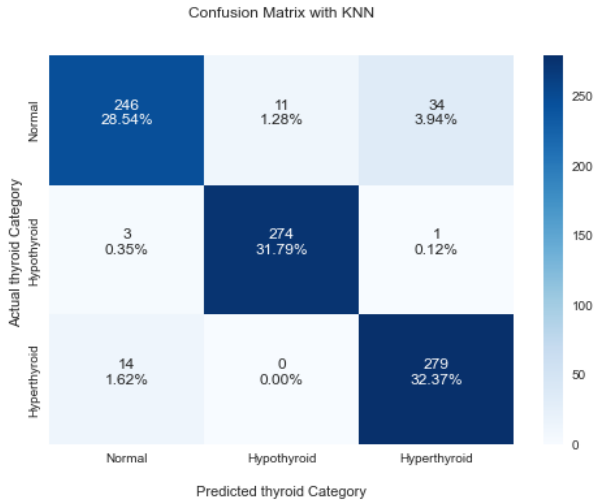

Figure 7.26. Confusion Matrix of KNN Model for selected features.

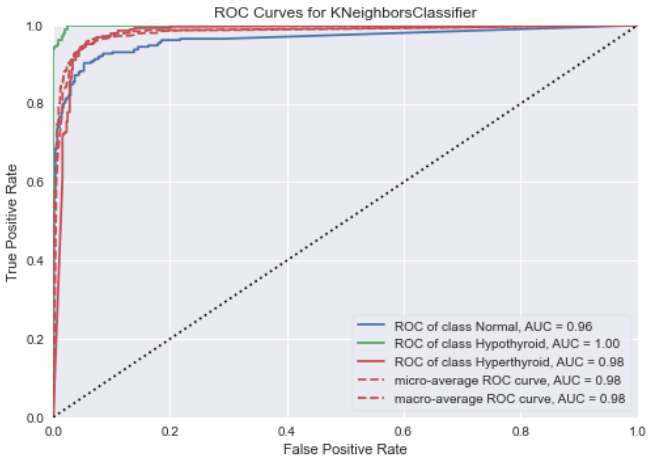

Figure 7.27. ROC Curve of KNN Model for selected features.

Table 7.18 shows the performance evaluation of the SVM model for selected features, Figure 7.28 and Figure 7.29 illustrates the confusion matrix and ROC curve when SVM models is applied to the balanced dataset for selected features.

Table 7.18. Performance evaluation of SVM for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	96.80%	90.72%	100%	100%	95.13%	0.95
Hypothyroid	99.41%	100%	99.13%	98.23%	99.10%	0.99
Hyperthyroid	97.40%	100%	96.04%	93.01%	96.40%	0.95

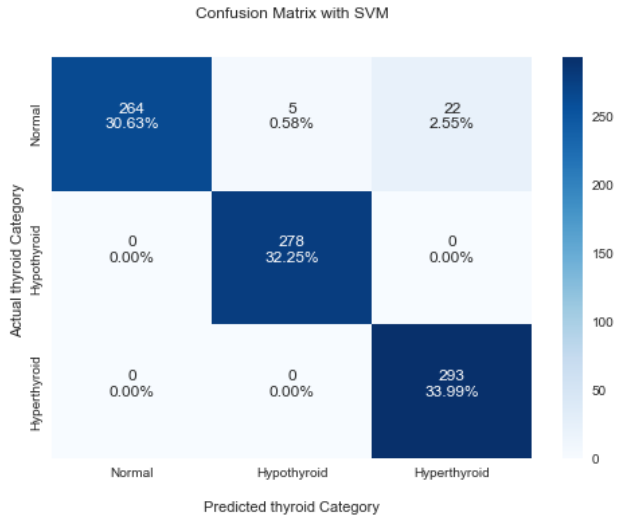

Figure 7.28. Confusion Matrix of SVM Model for selected features.

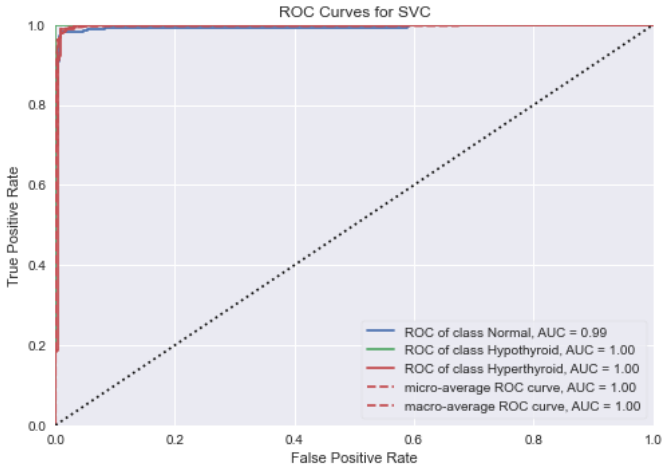

Figure 7.29. ROC Curve of SVM Model for selected features.

Table 7.19 shows the performance evaluation of the DT model for selected features, Figure 7.30 and Figure 7.31 illustrates the confusion matrix and ROC curve when DT models is applied to the balanced dataset for selected features.

Table 7.19. Performance evaluation of DT for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	100%	100%	100%	100%	100%	1.0
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	100%	100%	100%	100%	100%	1.0

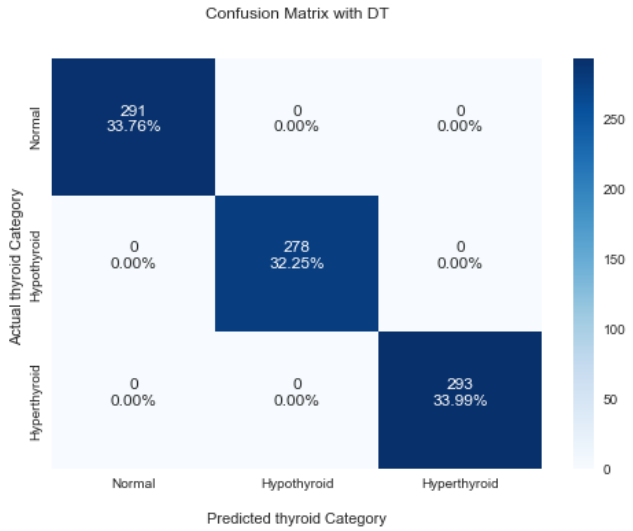

Figure 7.30. Confusion Matrix of DT Model for selected features.

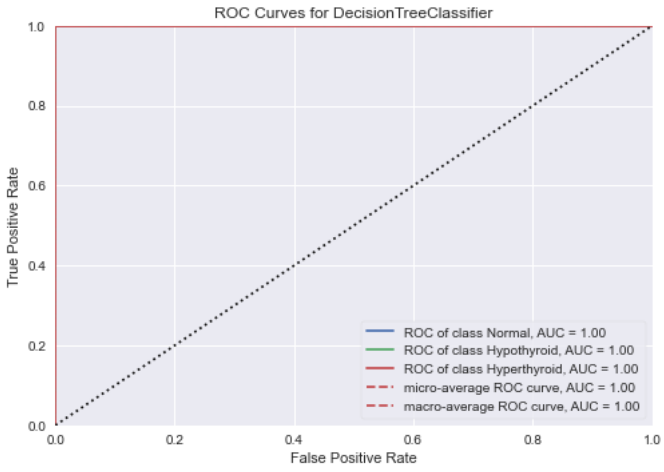

Figure 7.31. ROC Curve of DT Model for selected features.

Table 7.20 shows the performance evaluation of the NB model for selected features, Figure 7.32 and Figure 7.33 illustrates the confusion matrix and ROC curve when NB models is applied to the balanced dataset for selected features.

Table 7.20. Performance evaluation of NB for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	97.02%	91.40%	100%	100%	95.51%	0.94
Hypothyroid	97.50%	100%	96.40%	92.70%	96.19%	0.94
Hyperthyroid	99.70%	100%	99.50%	99.00%	99.50%	0.99

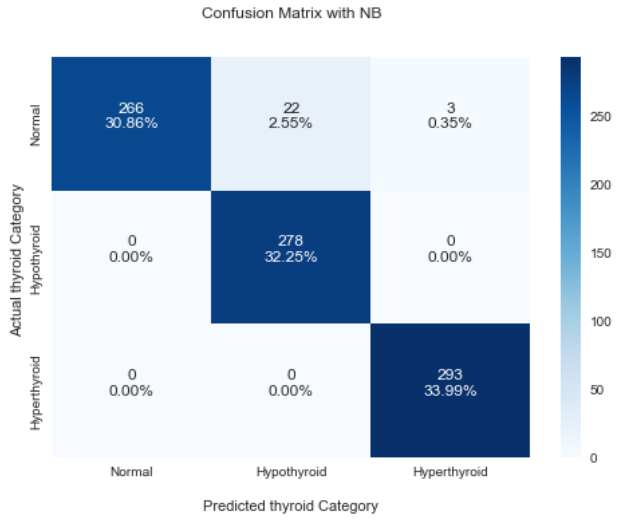

Figure 7.32. Confusion Matrix of NB Model for selected features.

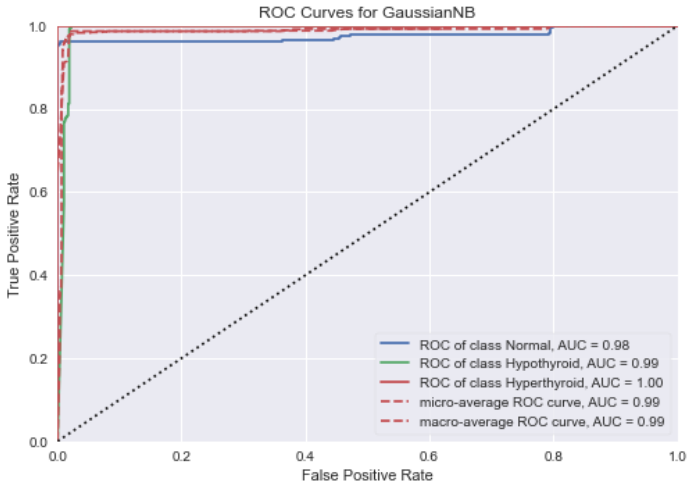

Figure 7.33. ROC Curve of NB Model for selected features.

Table 7.21 shows the performance evaluation of the LR model for selected features, Figure 7.34 and Figure 7.35 illustrates the confusion matrix and ROC curve when LR models is applied to the balanced dataset for selected features.

Table 7.21. Performance evaluation of LR for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	89.80%	85.91%	91.80%	84.17%	85.03%	0.77
Hypothyroid	99.53%	99.64%	99.50%	98.92%	99.28%	0.99
Hyperthyroid	90.25%	84.30%	93.32%	86.70%	85.50%	0.78

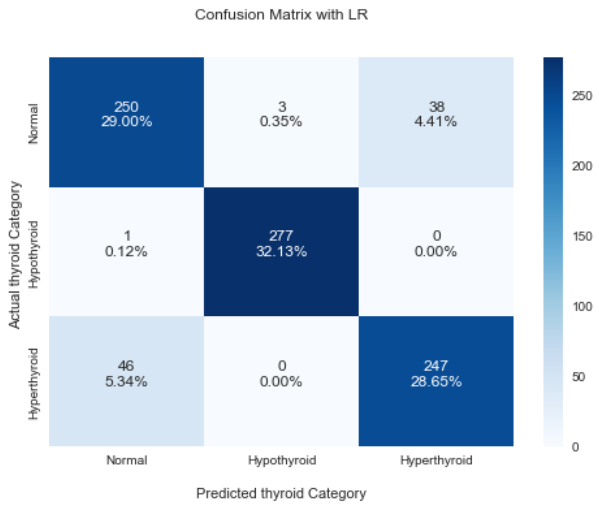

Figure 7.34. Confusion Matrix of LR Model for selected features.

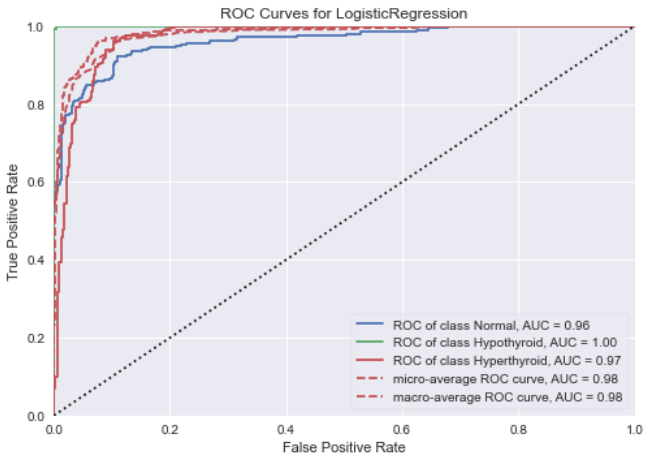

Figure 7.35. ROC Curve of LR Model for selected features.

Table 7.22 shows the performance evaluation of the MLP model for selected features, Figure 7.36 and Figure 7.37 illustrates the confusion matrix and ROC curve when MLP models is applied to the balanced dataset for selected features.

Table 7.22. Performance evaluation of MLP for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	96.63%	90.03%	100%	100%	94.80%	0.93
Hypothyroid	99.70%	100%	99.50%	98.93%	99.50%	0.99
Hyperthyroid	97.00%	100%	95.43%	91.84%	95.80%	0.94

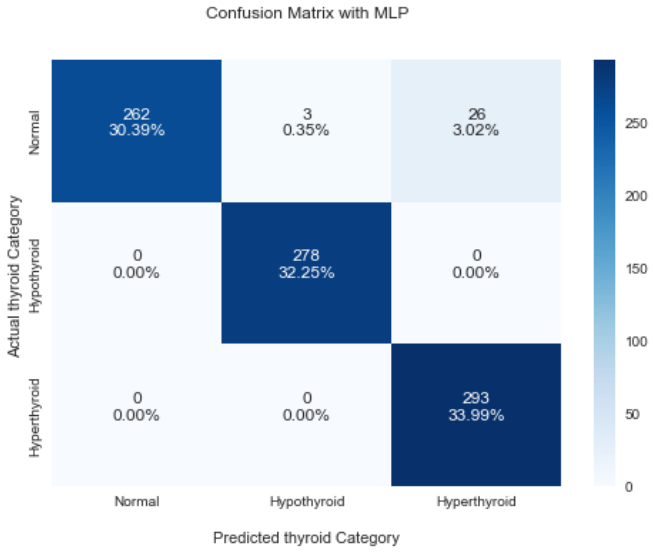

Figure 7.36. Confusion Matrix of MLP Model for selected features.

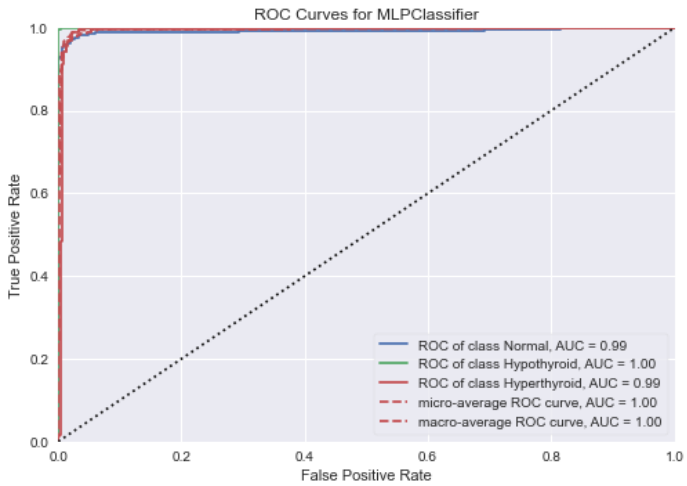

Figure 7.37. ROC Curve of MLP Model for selected features.

The performance measures were calculated for each traditional models with selected features above. The average model performances were shown in Table 7.23.

Table 7.23. Comparison based on the average performances for traditional models for selected features.

Models	KNN	SVM	DT	NB	LR	MLP
ACC	95.12%	97.90%	100%	98.06%	93.19%	97.80%
Sensitivity	92.80%	96.90%	100%	97.13%	90.00%	96.70%
Specificity	96.32%	98.40%	100%	98.60%	94.90%	98.30%
Precision	92.84%	97.08%	100%	97.21%	89.92%	96.92%
F1 score	92.70%	96.90%	100%	97.06%	89.92%	96.70%
MCC	0.89	0.95	1.0	0.96	0.85	0.95

For Ensembles models, each model will be implemented and performance metrics calculated for each class. Table 7.24 shows the performance evaluation of the RF model for selected features, Figure 7.38 and Figure 7.39 illustrates the confusion matrix and ROC curve when RF models is applied to the balanced dataset for selected features.

Table 7.24. Performance evaluation of RF for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	97.60%	100%	96.32%	93.26%	96.51%	0.95
Hypothyroid	99.07%	97.50%	99.82%	99.63%	98.54%	0.98
Hyperthyroid	98.25%	94.90%	100%	100%	97.40%	0.96

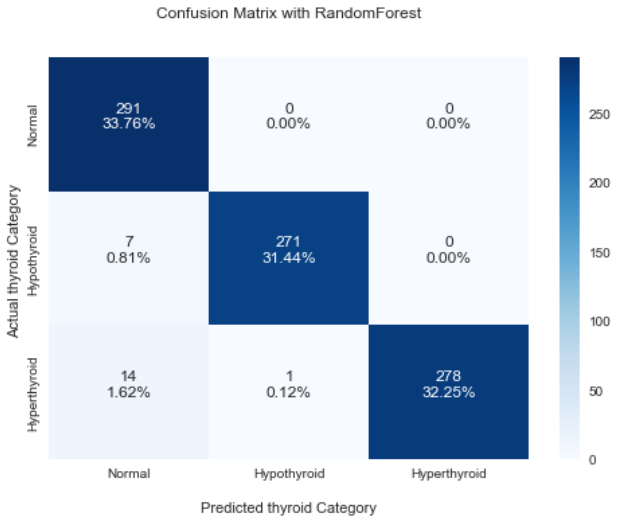

Figure 7.38. Confusion Matrix of RF Model for selected features.

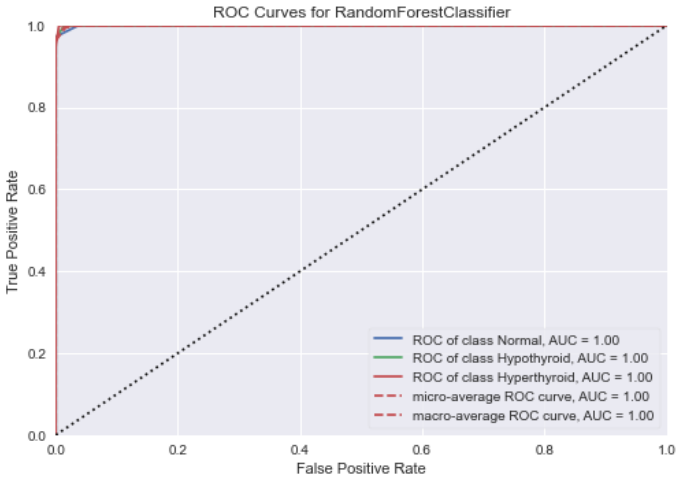

Figure 7.39. ROC Curve of RF Model for selected features.

Table 7.25 shows the performance evaluation of the XGboost model for selected features, Figure 7.40 and Figure 7.41 illustrates the confusion matrix and ROC curve when XGboost models is applied to the balanced dataset for selected features.

Table 7.25. Performance evaluation of XGboost for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	100%	100%	100%	100%	100%	1.0
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	100%	100%	100%	100%	100%	1.0

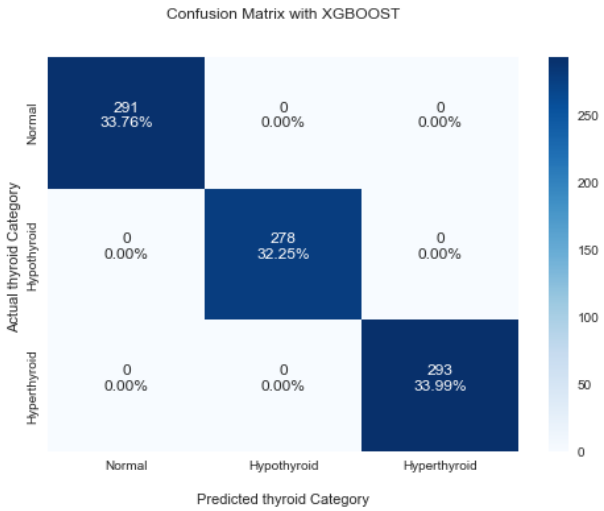

Figure 7.40. Confusion Matrix of XGBoost Model for selected features.

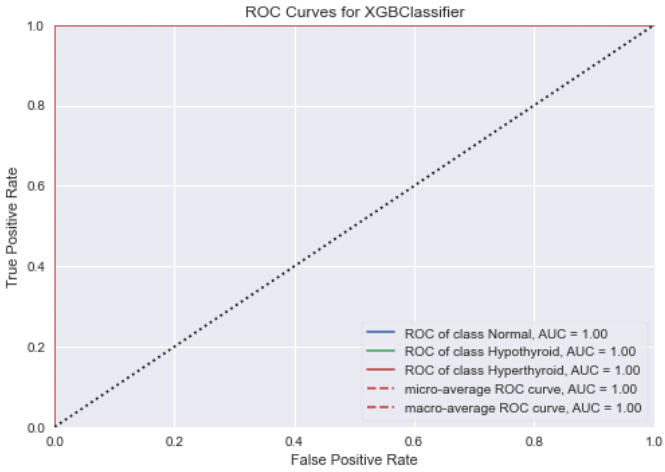

Figure 7.41. ROC Curve of XGBoost Model for selected features.

Table 7.26 shows the performance evaluation of the Soft Vote model for selected features, Figure 7.42 and Figure 7.43 illustrates the confusion matrix and ROC curve when Soft Vote models is applied to the balanced dataset for selected features.

Table 7.26. Performance evaluation of Soft Vote for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	99.07%	97.25%	100%	100%	98.60%	0.98
Hypothyroid	99.07%	100%	98.63%	97.20%	98.60%	0.98
Hyperthyroid	100%	100%	100%	100%	100%	1.0

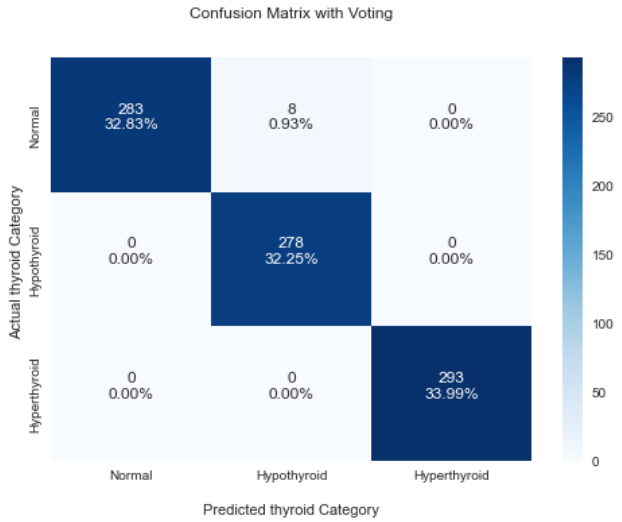

Figure 7.42. Confusion Matrix of Soft Vote Model for selected features.

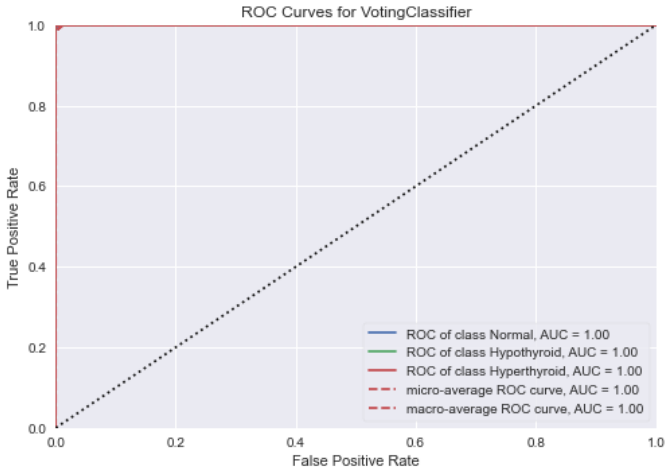

Figure 7.43. ROC Curve of Soft Vote Model for selected features.

Table 7.27 shows the performance evaluation of the Stacking model for selected features, Figure 7.44 and Figure 7.45 illustrates the confusion matrix and ROC curve when Stacking models is applied to the balanced dataset for selected features.

Table 7.27. Performance evaluation of Stacking for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	99.30%	97.93%	100%	100%	99.00%	0.98
Hypothyroid	99.90%	100%	99.82%	99.64%	99.82%	0.99
Hyperthyroid	99.41%	100%	99.12%	98.32%	99.15%	0.99

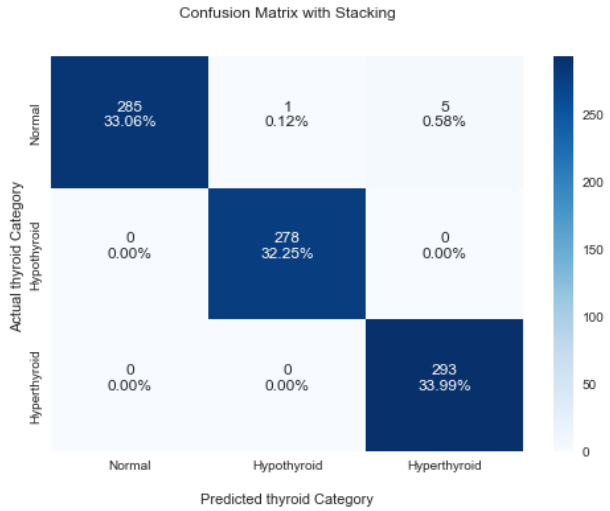

Figure 7.44. Confusion Matrix of Stacking Model for selected features.

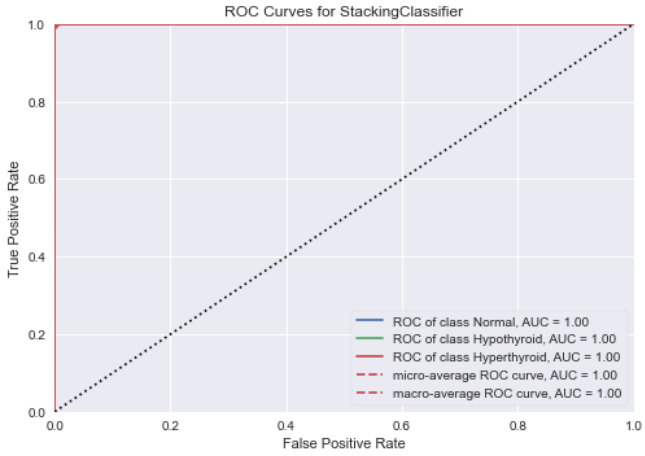

Figure 7.45. ROC Curve of Stacking Model for selected features.

Table 7.28 shows the performance evaluation of the Bagging model for selected features, Figure 7.46 and Figure 7.47 illustrates the confusion matrix and ROC curve when Bagging models is applied to the balanced dataset for selected features.

Table 7.28. Performance evaluation of Bagging for selected features.

Class	ACC	Sensitivity	Specificity	Precision	F1 score	MCC
Normal	100%	100%	100%	100%	100%	1.0
Hypothyroid	100%	100%	100%	100%	100%	1.0
Hyperthyroid	100%	100%	100%	100%	100%	1.0

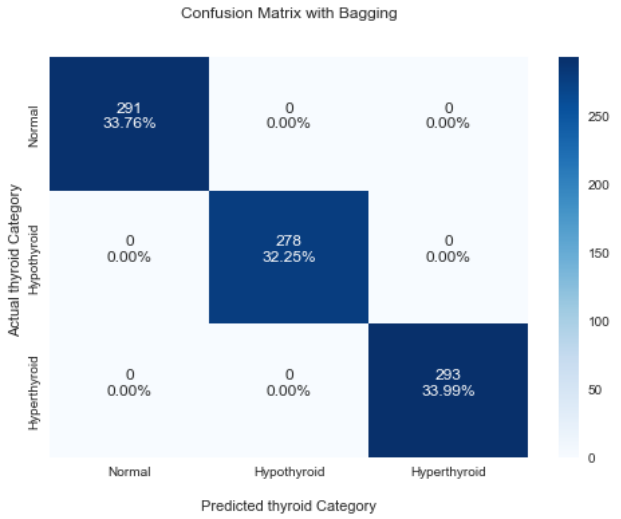

Figure 7.46. Confusion Matrix of Bagging Model for selected features.

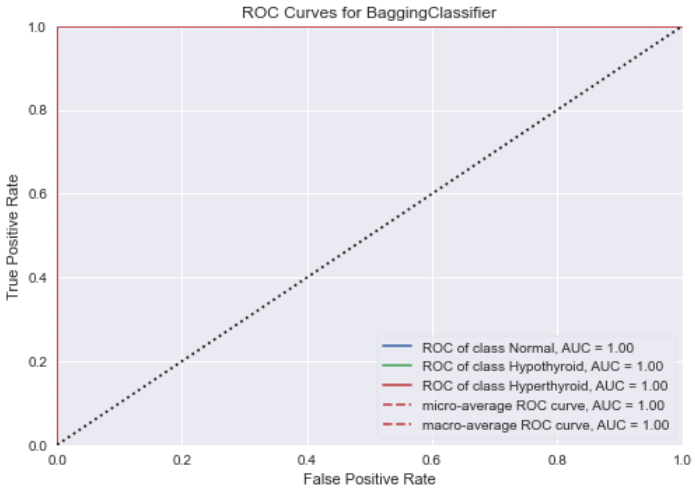

Figure 7.47. ROC Curve of Bagging Model for selected features.

The performance measures were calculated for each ensemble models for selected features above. The average model performances were shown in Table 7.29.

Table 7.29. Comparison based on the average performances for ensemble models with selected features.

Models	**RF**	**XGB**	**Soft Vote**	**Stacking**	**Bagging**
ACC	98.30%	100%	99.40%	99.53%	100%
Sensitivity	97.50%	100 %	99.08%	99.31%	100 %
Specificity	98.71%	100 %	99.54%	99.65%	100%
Precision	97.63%	100 %	99.06%	99.32%	100 %
F1 score	97.50%	100 %	99.06%	99.31%	100 %
MCC	0.96	1.0	0.98	0.99	1.0

7.4.1. Train Accuracy and Test Accuracy for Selected Features

As we mentioned earlier in all features, as in Table 7.30 shows the compare the ACC of training and the ACC of the test to see if there is overfitting.

Table 7.30. Comparison between training ACC and test ACC for selected features.

Models	Train ACC	Test ACC
KNN	100%	95.12%
SVM	97.21%	97.90%
DT	100%	100%
NB	98.11%	98.06%
LR	91.34%	93.19%
MLP	96.57%	97.80%
RF	99.20%	98.30%
XGB	100%	100%
Soft Vote	100%	99.40%
Stacking	99.75%	99.53%
Bagging	100%	100%

7.4.2. Time of Prediction for Selected Features

As above, in this experiment, the difference between the training time and the prediction time of six traditional models and five ensemble models was recorded. As with all features, we ran each model 10 times to reduce the impact of the computer's cache. The results of the comparison in time are shown in Table 7.31.

Table 7.31. Difference between the training time and prediction time for selected features.

Models	Training Time (seconds)	Predict Time (seconds)
KNN	0.056	0.025
SVM	2.564	0.022
DT	0.012	0.003
NB	0.007	0.003
LR	0.15	0.004
MLP	6.06	0.005
RF	0.016	0.007
XGB	0.877	0.018
Soft Vote	3.863	0.032
Stacking	58.159	0.113
Bagging	1.454	0.068

7.5. CROSS-VALIDATION RESULTS

Cross-validation is a technique used to evaluate the performance of a ML model on unseen data. It involves splitting the dataset into multiple subsets, training the model on one subset, and evaluating its performance on the remaining subsets. This process is repeated multiple times, with each subset being used as the evaluation set at least once. The final performance score is typically calculated as the average performance across all iterations.

Table 7.32 presents the results of the cross-validation procedure performed on the model for all features and feature selection with RFE results. The table shows the performance of the model on each iteration of the cross-validation, as well as the average performance across all iterations.

Table 7.32. Performance evaluation of cross-validation with all feature and selected features.

Model	Mean ACC with all features	Mean ACC with selected features
KNN	92.20%	93.40%
SVM	95.50%	97.00%
DT	87.00%	100%
NB	64.60%	97.00%
LR	88.10%	93.20%
MLP	97.80%	98.00%
RF	93.20%	99.20%
XGboost	100%	100%
Soft Vote	87.00%	99.30%
Stacking	99.10%	99.60%
Bagging	100%	100%

7.6. DISCUSSION

ML models achieved accurate results in diagnosing thyroid disease and exploring the features that determine the diagnosis of this disease, as this study proved that TSH, T4, and T3 are the features that determine the prediction of this disease. According to the ACC results, the classification performance of six traditional models and five Ensemble models were compared using all the features. It was found that the highest

ACC was obtained by DT and MLP for traditional models, XGboost and Bagging for Ensemble models. Cross-validation results for traditional models with all features found that the highest mean ACC was obtained by MLP, XGboost and Bagging for Ensemble models. When the RFE was used for feature selection, it was found that the DT and NB algorithm obtained the highest ACC of the traditional models, and the XGboost, Bagging, and Stacking models obtained the highest ACC of the Ensemble models. Cross-validation results for traditional models with RFE found that the highest mean ACC was obtained by DT and MLP for traditional models, XGboost and Bagging for Ensemble models.

7.6.1. Performance Comparison of The Proposed Model with State-of-The-Art Works

The results of this study were significantly improved to those of the previous studies discussed in the literature review. This improvement can be attributable to a number of study-implemented elements.

Firstly, the proposed study used a bigger and more diversified dataset than previous studies, allowing the proposed ML models to acquire more extensive and generalizable patterns from the dataset. Second, this thesis utilized more advanced and sophisticated ensemble models in ML, which were better able to capture the complicated and nonlinear interactions between the dataset variables.

In addition, more effective strategies were used to fine-tune the parameters of the ML model, which improved the performance of the models. In addition, the proposed study performed thorough cross-validation and feature selection, allowing for a more accurately assess the model's generalizability and dependability.

Overall, the combination of these factors allows the proposed study to generate more precise and reliable results than those presented in Table 7.32 of the literature review. These results have significant ramifications for the use of ML in this field.

Table 7.33. Comparison of results with literature review.

Authors (Year)	Methods	Dataset	ACC (%)
Shivastuti & Haneet Kour [24] (2021)	SVM and RF	Irvine (UCI)	ACC=91, ACC=89
Salman&Sonuç [29] (2021)	RF, KNN, LR, SVM, NB, MLP, LDA, DT	Private dataset Iraqi patients	ACC=98.93, ACC=90.93, ACC=91.47, ACC=92.27, ACC=81.33, ACC=97.6, ACC=83.2, ACC=98.4
Kousarrizi & F. Seiti [25] (2012)	SVM	Irvine (UCI) and Imam Khomeini Hospital	ACC=98.26
Chaubey & Bisen [26] (2020)	KNN	UCI	ACC=96.87
K. Geetha & Baboo [27] (2016)	NB	UCI	ACC=97.97
Aswathi and A. Antony [28] (2018)	SVM using particle swarm optimization	UCI	N/A
Sidiq U, Aaqib [30] (2019)	Decision Tree	UCI	ACC= 98.89
Yasir Iqbal Mir & Dr. Sonu Mittal [31] (2020)	Bagging, SVM, J48	Indian patients	ACC=98.56, ACC=99.08, ACC=92.07
Solmaz, R., Alkan, A., & Gunay, M. [32] (2020)	Ensemble Method	UCI	ACC=99.06, ACC=99.08
Shiva Borzouei, Hossein Mahjub [33] (2020)	LR, neural networks models	Imam Khomeini Hospital	Mean-ACC=91.4, ACC=96.3
Authors (Year)	**Methods**	**Dataset**	**ACC (%)**
Proposed Study	KNN, SVM, DT, NB, LR, MLP, RF, XGB, SoftVoting, Stacking, Bagging	Private dataset Iraqi patients	ACC=95.12, ACC=97.90, ACC=100, ACC=98.06, ACC=93.19, ACC=97.80, ACC=98.30, ACC=100, ACC=99.40, ACC=99.53, ACC=100

The proposed model outperforms the state-of-the-art works because the values of performance evaluation metrics are higher than the previous study that used this dataset. Salman K. and Sonuç [29] collected the dataset used in this thesis from private labs in Iraq, where the RF model achieved the highest ACC of 98.93%.

Although they used a significantly unbalanced data set, which means their ACC is confusing, the proposed model balances the dataset and still outperforms it. In this proposed model, several methods were used to verify overfitting, such as the difference between training ACC to test ACC and cross-validation.

PART 8

CONCLUSION

In this thesis, a complete approach was presented to classify thyroid disease using six traditional models (KNN, SVM, DT, NB, LR, MLP) and five Ensemble models (RF, XGboost, Soft Vote, Stacking, Bagging). These models can successfully diagnose thyroid disease through analysis and understanding of the dataset and achieve accurate results in prediction. The proposed method was tested in two steps. All features of the dataset were used in the first step after data from the missing values was added, and the dataset was processed and balanced. The highest ACC of traditional models was found to be obtained by DT and MLP at 99.92% and 97.30%, respectively. Ensemble models obtained 100% ACC in the XGboost and Bagging models.

In the second step, the RFE model was used to determine the best correlated features for prediction. The RFE model was also applied to traditional models and achieved 100%, and 98.06% ACC in DT and NB, respectively. As for ensemble models, XGboost and Bagging also achieved 100% ACC, and the Stacking model achieved 99.53% ACC. The proposed solution used a more suitable model than state-of-the-art works, based on the balancing of data that was considered highly confusing and the selection of features. the proposed model was creating to be better than the previous one. The results show that the proposed ensemble models outperform traditional models in terms of ACC, Sensitivity, Specificity, Precision, F1 score, and MCC. A comparison was made between training time and prediction time, and it was found that the time taken for training and prediction is a relatively good time to apply both traditional and ensemble models at the lowest possible cost. Cross-validation and the difference between training ACC to test ACC were two of the methods that were used in this model in order to find out whether or not its overfitting the model. This

proposed model for development will be running on the website platform and can be run on iOS and Android applications, as a future work.

REFERENCES

1. Ma, C., Cheng, X., Xue, F., Li, X., Yin, Y., Wu, J., ... & Xu, T. (2020). Validation of an approach using only patient big data from clinical laboratories to establish reference intervals for thyroid hormones based on data mining. ***Clinical Biochemistry***, 80, 25-30.

2. Azar, A. T., Hassanien, A. E., & Kim, T. H. (2012). Expert system based on neural-fuzzy rules for thyroid diseases diagnosis. In Computer applications for bio-technology, multimedia, and Ubiquitous City (pp. 94-105). ***Springer***, Berlin, Heidelberg.

3. Keleş, A., & Keleş, A. (2008). ESTDD: Expert system for thyroid diseases diagnosis. ***Expert Systems with Applications***, 34(1), 242-246.

4. Vanderpump, M. P. (2011). The epidemiology of thyroid disease. ***British medical bulletin***, 99(1).

5. Zimmermann, M. B., & Boelaert, K. (2015). Iodine deficiency and thyroid disorders. The lancet Diabetes & endocrinology, 3(4), 286-295.

6. Roshan, B. D., & Sharmili, K. C. (2017). A Study of Data Mining Techniques to Detect Thyroid Disease. ***International Journal of Innovative Research in Science***, Engineering and Technology, 6(11), 549-553.

7. Chakrabarti , S. , Ester , M. , Fayyad , U. , Gehrke , J. , Han , J. , Morishita , S. , ... & Wang , W. (2006). Data mining curriculum: A proposal (Version 1.0). ***Intensive Working Group of the ACM SIGKDD Curriculum Committee***, 140, 1-10.

8. Banu, G. R. (2016). Predicting thyroid disease using linear discriminant analysis (LDA) data mining technique. ***Commun. Appl. Electron***.(CAE), 4, 4-6.

9. Rajam, K., & Priyadarsini, R. J. (2016). A Survey on Diagnosis of Thyroid Disease Using Data Mining Techniques. ***IJCSMC***, 5(5), 354-358.

10. Temurtas, F. (2009). A comparative study on thyroid disease diagnosis using neural networks. ***Expert Systems with Applications***, 36(1), 944-949.

11. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. ***Computational and structural biotechnology journal***, 13, 8-17.

12. Shukla, A., Tiwari, R., Kaur, P., & Janghel, R. R. (2009, March). Diagnosis of thyroid disorders using artificial neural networks. ***In 2009 IEEE International Advance Computing Conference*** (pp. 1016-1020). IEEE.

13. Aswad, S. A., & Sonuç, E. (2020, October). Classification of VPN network traffic flow using time related features on Apache Spark***. In 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*** (pp. 1-8). IEEE.

14. Banu, G. R. (2016). A Role of decision Tree classification data Mining Technique in Diagnosing Thyroid disease. ***International Journal of Computer Sciences and Engineering***, 4(11), 64-70.

15. Chandio, J. A., Sahito, A., Soomrani, M. A. R., & Abbasi, S. A. (2016, April). TDV: Intelligent system for thyroid disease visualization. ***In 2016 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*** (pp. 106-112). IEEE.

16. Murdoch, T. B., & Detsky, A. S. (2013). The inevitable application of big data to health care. ***Jama***, 309(13), 1351-1352.

17. Seo, C. K., Kim, J. H., & Kwon, S. Y. (2018, December). A study on modeling using big data and deep learning method for failure diagnosis of system. ***In 2018 IEEE International Conference on Big Data (Big Data)*** (pp. 4747-4751). IEEE.

18. Yue, F., Chen, C., Yan, Z., Chen, C., Guo, Z., Zhang, Z., ... & Lv, X. (2020). Fourier transform infrared spectroscopy combined with deep learning and data enhancement for quick diagnosis of abnormal thyroid function. ***Photo diagnosis and Photodynamic Therapy***, 32, 101923.

19. Lee, C. H., & Yoon, H. J. (2017). Medical big data: promise and challenges. ***Kidney research and clinical practice***, 36(1), 3.

20. Werner, S. C., Ingbar, S. H., Braverman, L. E., & Utiger, R. D. (Eds.). (2005). Werner & Ingbar's the thyroid: a fundamental and clinical text (Vol. 549). Lippincott Williams & Wilkins.

21. Can, A. S., & Rehman, A. (2021). Goiter. In StatPearls [Internet]. ***StatPearls Publishing.***

22. Umar Sidiq, D., Aaqib, S. M., & Khan, R. A. (2019). Diagnosis of various thyroid ailments using data mining classification techniques. ***Int J Sci Res Coput Sci Inf Technol***, 5, 131-6.

23. Tafti, A. P., Behravesh, E., Assefi, M., LaRose, E., Badger, J., Mayer, J., ... & Peissig, P. (2017, December). bigNN: An open-source big data toolkit focused on biomedical sentence classification. ***In 2017 IEEE International Conference on Big Data (Big Data)*** (pp. 3888-3896). IEEE.

24. Shivastuti, H. K., Manhas, J., & Sharma, V. (2021). Performance Evaluation of SVM and Random Forest for the Diagnosis of Thyroid Disorder. ***Int. J. Res. Appl. Sci. Eng. Technol***, 9, 945-947.

25. Kousarrizi, M. N., Seiti, F., & Teshnehlab, M. (2012). An experimental comparative study on thyroid disease diagnosis based on feature subset selection and classification. ***International Journal of Electrical & Computer Sciences IJECS-IJENS***, 12(01), 13-20.

26. Chaubey, G., Bisen, D., Arjaria, S., & Yadav, V. (2021). Thyroid disease prediction using machine learning approaches. ***National Academy Science Letters***, 44(3), 233-238.

27. Geetha, K., & Baboo, S. S. (2016). An empirical model for thyroid disease classification using evolutionary multivariate Bayseian prediction method. ***Global Journal of Computer Science and Technology.***

28. Aswathi, A. K., & Antony, A. (2018, April). An intelligent system for thyroid disease classification and diagnosis. ***In 2018 Second international conference on inventive communication and computational technologies (ICICCT)*** (pp. 1261-1264). IEEE.

29. Sonuç, E. (2021, July). Thyroid Disease Classification Using Machine Learning Algorithms. In Journal of Physics: Conference Series (Vol. 1963, No. 1, p. 012140). ***IOP Publishing.***

30. Umar Sidiq, D., Aaqib, S. M., & Khan, R. A. (2019). Diagnosis of various thyroid ailments using data mining classification techniques***. Int J Sci Res Coput Sci Inf Technol***, 5, 131-6.

31. Mir, Y. I., & Mittal, S. (2020). Thyroid disease prediction using hybrid machine learning techniques: An effective framework. ***International Journal of Scientific & Technology Research***, 9(2), 2868-2874.

32. SOLMAZ, R., ALKAN, A., & GUNAY, M. (2020). Mobile diagnosis of thyroid based on ensemble classifier. ***Dicle University Engineering Faculty Journal of Engineering***, 11 (3), 915-924.

33. Borzouei, S., Mahjub, H., Sajadi, N. A., & Farhadian, M. (2020). Diagnosing thyroid disorders: Comparison of logistic regression and neural network models. ***Journal of Family Medicine and Primary Care***, 9(3), 1470.

34. Van Rossum, G. (2003). An introduction to Python (p. 115). F. L. Drake (Ed.). ***Bristol: Network Theory Ltd.***

35. Beazley, D. M. (2009). Python essential reference. ***Addison-Wesley Professional***.

36. VanderPlas, J. (2016). Python data science handbook: Essential tools for working with data. " ***O'Reilly Media, Inc.***".

37. Nkasu, M. M. (2020). Investigation of the effects of critical success factors on enterprise resource planning (ERP) systems implementation in the United Arab Emirates. In Smart Intelligent Computing and Applications (pp. 611-623). ***Springer***, Singapore.

38. Jordan, M. I., & Mitchell, T. M. (2015). Machine learning: Trends, perspectives, and prospects. ***Science***, 349(6245), 255-260.

39. Kodratoff, Y. (2014). Introduction to machine learning. ***Elsevier***.

40. Simeone, O. (2018). A brief introduction to machine learning for engineers. ***Foundations and Trends® in Signal Processing***, 12(3-4), 200-431.

41. "Types of Machine Learning | MLK - Machine Learning Knowledge", ***https://machinelearningknowledge.ai/types-of-machine-learning***/ (2021).

42. Wahbeh, A. H., Al-Radaideh, Q. A., Al-Kabi, M. N., & Al-Shawakfa, E. M. (2011). A comparison study between data mining tools over some classification methods. ***International Journal of Advanced Computer Science and Applications***, 8(2), 18-26.

43. jose, italo. (2018, November 8). KNN (K-Nearest Neighbors) #1 - towardsdatascience.com. Retrieved November 21, 2022, from ***https://towardsdatascience.com/knn-k-nearest-neighbors-1-a4707b24bd1d***.

44. Imandoust, S. B., & Bolandraftar, M. (2013). Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background. ***International journal of engineering research and applications***, 3(5), 605-610.

45. Sha'Abani, M. N. A. H., Fuad, N., Jamal, N., & Ismail, M. F. (2020). kNN and SVM classification for EEG: a review. ***InECCE2019***, 555-565.

46. Cortes, C., & Vapnik, V. (1995). Support-vector networks. ***Machine learning***, 20(3), 273-297.

47. Yang, Y., Li, J., & Yang, Y. (2015, December). The research of the fast SVM classifier method***. In 2015 12th international computer conference on wavelet active media technology and information processing (ICCWAMTIP)*** (pp. 121-124). IEEE.

48. Yıldız, T. K., Yurtay, N., & Öneç, B. (2021). Classifying anemia types using artificial learning methods. ***Engineering Science and Technology, an International Journal***, 24(1), 50-70.

49. Agarap, A. F. M. (2018, February). A neural network architecture combining gated recurrent unit (GRU) and support vector machine (SVM) for intrusion detection in network traffic data. ***In Proceedings of the 2018 10th international conference on machine learning and computing*** (pp. 26-30).

50. Maimon, O., & Rokach, L. (2005). Decomposition methodology for knowledge discovery and data mining. In Data mining and knowledge discovery handbook (pp. 981-1003). ***Springer***, Boston, MA.

51. "1.10. Decision Tree-Scikit-Learn 0.24.2 Documentation", ***https://scikit-learn.org/stable/modules/tree.html#mathematical-formulation*** (2021).

52. The 365 Team (Ed.). (2021, November 17). Introduction to decision trees: Why should you use them? 365 Data Science. Retrieved November 22, 2022, from ***https://365datascience.com/tutorials/machine-learning-tutorials/decision-trees/***

53. Meena, K., Tayal, D. K., Gupta, V., & Fatima, A. (2019). Using classification techniques for statistical analysis of Anemia. ***Artificial Intelligence in Medicine***, 94, 138-152.

54. Ghiasi, M. M., Zendehboudi, S., & Mohsenipour, A. A. (2020). Decision tree-based diagnosis of coronary artery disease: CART model. ***Computer methods and programs in biomedicine***, 192, 105400.

55. Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., ... & Steinberg, D. (2008). Top 10 algorithms in data mining. ***Knowledge and information systems***, 14(1), 1-37.

56. Duda, R. O., Hart, P. E., & Stork, D. G. (1973). Pattern classification and scene analysis (Vol. 3, pp. 731-739). ***New York: Wiley***.

57. Zhang, L., Jiang, L., Li, C., & Kong, G. (2016). Two feature weighting approaches for naive Bayes text classifiers. ***Knowledge-Based Systems***, 100, 137-144.

58. Fadhil, Z. M. (2021). Hybrid of K-means clustering and naive Bayes classifier for predicting performance of an employee. ***Periodicals of Engineering and Natural Sciences (PEN)***, 9(2), 799-807.

59. Christodoulou, E., Ma, J., Collins, G. S., Steyerberg, E. W., Verbakel, J. Y., & Van Calster, B. (2019). A systematic review shows no performance benefit of machine learning over logistic regression for clinical prediction models. ***Journal of clinical epidemiology***, 110, 12-22.

60. Korkmaz, M., Güney, S., & YİĞİTER, Ş. (2012). The importance of logistic regression implementations in the Turkish livestock sector and logistic regression implementations/fields. ***Harran Tarım ve Gıda Bilimleri Dergisi***, 16(2), 25-36.

61. Dreiseitl, S., & Ohno-Machado, L. (2002). Logistic regression and artificial neural network classification models: a methodology review. ***Journal of biomedical informatics***, 35(5-6), 352-359.

62. Alenzi, H. Z., & Aljehane, N. O. (2020). Fraud detection in credit cards using logistic regression. ***International Journal of Advanced Computer Science and Applications***, 11(12).

63. Stoltzfus, J. C. (2011). Logistic regression: a brief primer. ***Academic emergency medicine***, 18(10), 1099-1104.

64. Pacheco, W. D. N., & López, F. R. J. (2019, April). Tomato classification according to organoleptic maturity (coloration) using machine learning algorithms K-NN, MLP, and K-Means Clustering. ***In 2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)*** (pp. 1-5). IEEE.

65. Sarraf Shirazi, A., & Frigaard, I. (2021). SlurryNet: Predicting Critical Velocities and Frictional Pressure Drops in Oilfield Suspension Flows. ***Energies***, 14(5), 1263.

66. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. ***Communications of the ACM***, 60(6), 84-90.

67. Boulesteix, A. L., Janitza, S., Kruppa, J., & König, I. R. (2012). Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. ***Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery***, 2(6), 493-507.

68. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees. ***Routledge***.

69. Siva Sai, Y. S. S. S. S., Liming, B. O., Longhao, Z., Xuanchang, L., & Balaji, A. (2022, May 20). Which is the real money maker, Random Forest Regressors or LSTM networks? Medium. Retrieved November 29, 2022, from ***https://medium.com/@nusfintech.ml/which-is-the-real-money-maker-random-forest-regressors-or-lstm-networks-a2153c0f8e92***

70. Dikker, J. (2017). Master thesis Boosted tree learning for balanced item recommendation in online retail.

71. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. ***In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*** (pp. 785-794).

72. Salam Patrous, Z. (2018). Evaluating XGBoost for user classification by using behavioral features extracted from smartphone sensors.

73. Guo, R., Zhao, Z., Wang, T., Liu, G., Zhao, J., & Gao, D. (2020). Degradation state recognition of piston pump based on ICEEMDAN and XGBoost. ***Applied Sciences***, 10(18), 6593.

74. Raschka, S. (2015). Python machine learning. ***Packt publishing ltd.***

75. Patil, D. R., & Patil, J. B. (2018). Malicious URLs detection using decision tree classifiers and majority voting technique. ***Cybernetics and Information Technologies***, 18(1), 11-29.

76. Wolpert, D. H. (1992). Stacked generalization. ***Neural networks***, 5(2), 241-259.

77. Sun, W., & Li, Z. (2020). Hourly PM2. 5 concentration forecasting based on feature extraction and stacking-driven ensemble model for the winter of the Beijing-Tianjin-Hebei area. ***Atmospheric Pollution Research***, 11(6), 110-121.

78. D'Souza, J. (2018, March 22). Introducing ensemble: More is better than one! Medium. Retrieved November 29, 2022, from ***https://medium.com/greyatom/introducing-ensemble-more-is-better-than-one-436a448350cf***

79. Leo, B. (1996). Bagging Predictors in Machine Learning.

80. Polikar, R. (2012). Ensemble learning. In Ensemble machine learning (pp. 1-34). ***Springer***, Boston, MA.

81. Galdi, P., & Tagliaferri, R. (2018). Data mining: accuracy and error measures for classification and prediction. ***Encyclopedia of Bioinformatics and Computational Biology***, 431-436.

82. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, JM, & Herrera, F. (2016). Big data preprocessing: methods and prospects. ***Big Data Analytics***, 1 (1), 1-22.

83. Han, J., Kamber, M., & Pei, J. (2012). Outlier detection. Data mining: concepts and techniques, 543-584.

84. Prati, R. C., Batista, G. E., & Monard, M. C. (2009, December). Data mining with imbalanced class distributions: concepts and methods. ***In IICAI*** (pp. 359-376).

85. Yang, Q., & Wu, X. (2006). 10 challenging problems in data mining research. ***International Journal of Information Technology & Decision Making***, 5(04), 597-604.

86. Rastgoo, M., Lemaitre, G., Massich i Vall, J., Morel, O., Marzani, F., García Campos, R., & Meriaudeau, F. (2016). Tackling the problem of data imbalancing for melanoma classification.

87. He, H., & Garcia, E. A. (2009). Learning from imbalanced data. ***IEEE Transactions on knowledge and data engineering***, 21(9), 1263-1284.

88. Blagus, R., & Lusa, L. (2013). SMOTE for high-dimensional class-imbalanced data. ***BMC bioinformatics***, 14(1), 1-16.

89. Hoaglin, D. C., Mosteller, F., & Tukey, J. W. (1983). Understanding robust and exploratory data anlysis. ***Wiley series in probability and mathematical statistics***.

90. Borg, I., & Groenen, P. J. (2005). Modern multidimensional scaling: Theory and applications. ***Springer Science & Business Media.***

91. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R. P., Tang, J., & Liu, H. (2017). Feature selection: A data perspective**.** ***ACM computing surveys (CSUR)***, 50(6), 1-45.

92. Das, A. K., Sengupta, S., & Bhattacharyya, S. (2018). A group incremental feature selection for classification using rough set theory based genetic algorithm. ***Applied Soft Computing***, 65, 400-411.

93. Demarchi, L., Kania, A., Ciężkowski, W., Piórkowski, H., Oświecimska-Piasko, Z., & Chormański, J. (2020). Recursive feature elimination and random forest classification of natura 2000 grasslands in lowland river valleys of poland based on airborne hyperspectral and LiDAR data fusion. ***Remote Sensing***, 12 (11), 1842.

94. Sehirli, E., & Turan, M. K. (2021). A Novel Method for Segmentation of QRS Complex on ECG Signals and Classify Cardiovascular Diseases via a Hybrid Model Based on Machine Learning. ***International Journal of Intelligent Systems and Applications in Engineering***, 9(1), 12-21.

95. Tharwat, A. (2020). Classification assessment methods. ***Applied Computing and Informatics.***

96. Sokolova, M., Japkowicz, N., & Szpakowicz, S. (2006, December). Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. ***In Australasian joint conference on artificial intelligence*** (pp. 1015-1021). ***Springer***, Berlin, Heidelberg.

97. Ayon, S. I., Islam, M. M., & Hossain, M. R. (2020). Coronary artery heart disease prediction: a comparative study of computational intelligence techniques. ***IETE Journal of Research***, 1-20.

98. Matthews, B. W. (1975). Comparison of the predicted and observed secondary structure of T4 phage lysozyme. ***Biochimica et Biophysica Acta (BBA)-Protein Structure***, 405(2), 442-451.

99. Bahramirad, S., Mustapha, A., & Eshraghi, M. (2013, September). Classification of liver disease diagnosis: a comparative study. ***In 2013 Second International Conference on Informatics & Applications (ICIA)*** (pp. 42-46). IEEE.

100. Kirov, G., Tredget, J., John, R., Owen, M. J., & Lazarus, J. H. (2005). A cross-sectional and a prospective study of thyroid disorders in lithium-treated patients. ***Journal of affective disorders***, 87(2-3), 313-317.

RESUME

Muntadher ALSAADAWI completed his high school studies in Al-Maaref secondary school in Diwaniyah in 2013. He completed his bachelor’s studies at the Islamic University / Computer Technology Engineering in 2019. To complete his master’s degree, he moved to Turkey/Karabük in 2020. He started Master's degree at Karabük University / Department of Computer Engineering.