PREDICTIVE DIAGNOSIS ANALYSIS OF THYROID DISEASE USING MACHINE LEARNING

Project report submitted in partial fulfilment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering

By

Utkarsh Jaiswal (181346)

Under the supervision of

Dr. Rakesh Kanji Sir

to

Department of Computer Science & Engineering and Information Technology
Jaypee University of Information Technology Waknaghat,
Solan-173234, Himachal Pradesh

Candidate's Declaration

I hereby declare that the work presented in this report entitled **" Predictive Diagnosis Analysis of Thyroid Disease Using Machine Learning"** in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering** submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology, Waknaghat is an authentic record of my own work carried out over a period from January 2022 to May 2022 under the supervision of **Dr. Rakesh Kanji** (Assistant professor (SG), CSE & IT).

Utkarsh Jaiswal.

Utkarsh Jaiswal, 181346

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Rakesh Kanji

Dr Rakesh Kanji
Assistant Professor
Computer Science & Engineering
Dated: 14 May 2022

ACKNOWLEDGEMENT

I would like to thank and express our gratitude to our Project supervisor Dr. Rakesh Kanji Sir for the opportunity that he provided us with his vast experience and wonderful guidance in my project “Predictive Diagnosis Analysis of Thyroid Disease Using Machine Learning”. The outcome would not be possible without his guidance. This project taught me many new things and helped to strengthen concepts of Machine Learning. Next, I would like to express my special thanks to the Lab Assistant for cordially contacting us and helping us in finishing this project within the specified time. Lastly, I would like to thank my friends and parents for their help and support.

Utkarsh Jaiswal.

Utkarsh Jaiswal (181346)

PROJECT REPORT STRUCTURE

TABLE OF CONTENTS

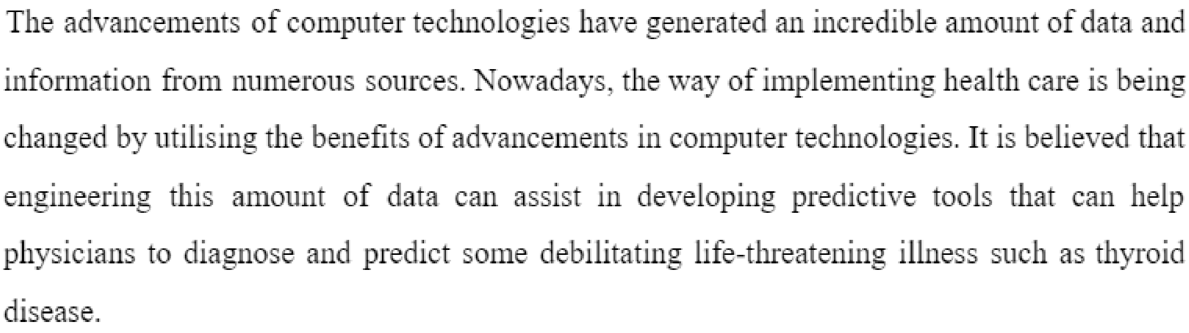
LIST OF TABLES

LIST OF FIGURES

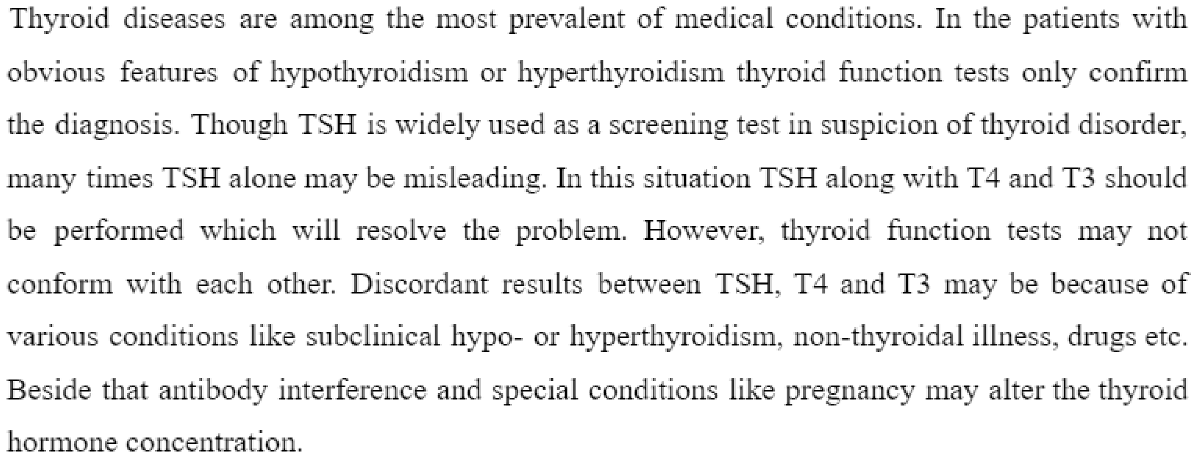
LIST OF GRAPHS

ABSTRACT

Thyroid disorder is the leading cause of medical diagnosis and prediction development, which medical science is a complicated axiom. Thyroid organ is one of our body's fundamental organs. Thyroid chemical emissions are answerable for directing digestion. Hyperthyroidism and hypothyroidism are the two unmistakable thyroid issues that produce thyroid chemicals for control of body digestion.



The advancements of computer technologies have generated an incredible amount of data and information from numerous sources. Nowadays, the way of implementing health care is being changed by utilising the benefits of advancements in computer technologies. It is believed that engineering this amount of data can assist in developing predictive tools that can help physicians to diagnose and predict some debilitating life-threatening illness such as thyroid disease.



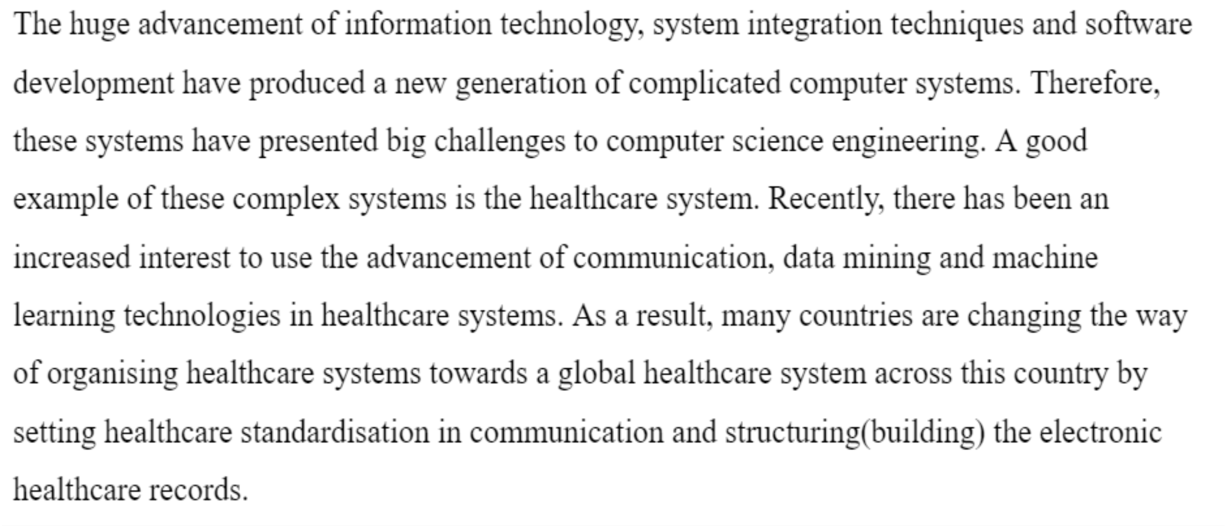
Thyroid diseases are among the most prevalent of medical conditions. In the patients with obvious features of hypothyroidism or hyperthyroidism thyroid function tests only confirm the diagnosis. Though TSH is widely used as a screening test in suspicion of thyroid disorder, many times TSH alone may be misleading. In this situation TSH along with T4 and T3 should be performed which will resolve the problem. However, thyroid function tests may not conform with each other. Discordant results between TSH, T4 and T3 may be because of various conditions like subclinical hypo- or hyperthyroidism, non-thyroidal illness, drugs etc. Beside that antibody interference and special conditions like pregnancy may alter the thyroid hormone concentration.

The major goal of this project is to give a concise source of reference for researchers who want to use decision trees, a common data mining method, in their field of work. With this goal in mind, we examined six performance indicators / metrics (Classification Accuracy [ACC (in percentage)], Mean Absolute Error (MAE), Precision (PRE), Recall (REC), and F-Measure (FME) to compare commonly used decision tree algorithms for classifying forms of thyroid illness. We believe that this project will give a useful summary of current demonstrations on this Thyroid disease diagnosis analysis and demonstrate how to utilise decision trees and various classification algorithms as a data mining tool.

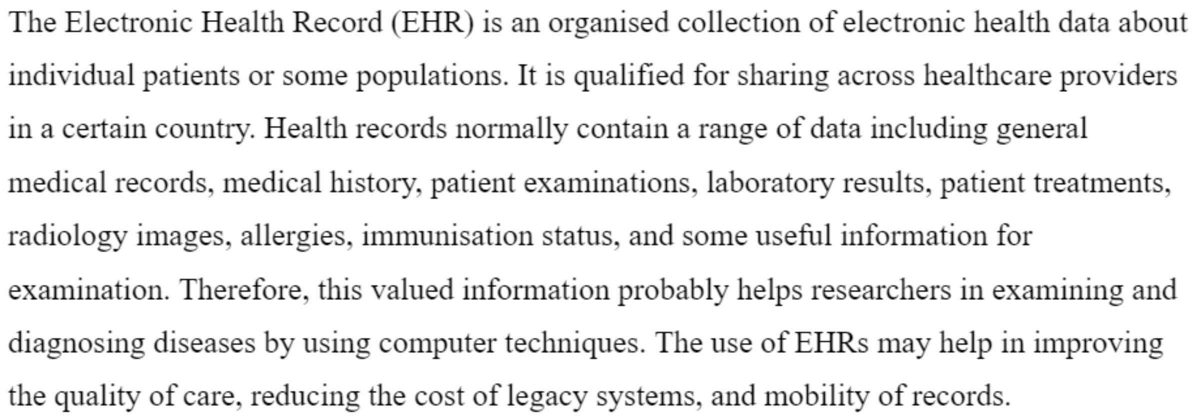
CHAPTER 1

INTRODUCTION

1.1 Overview

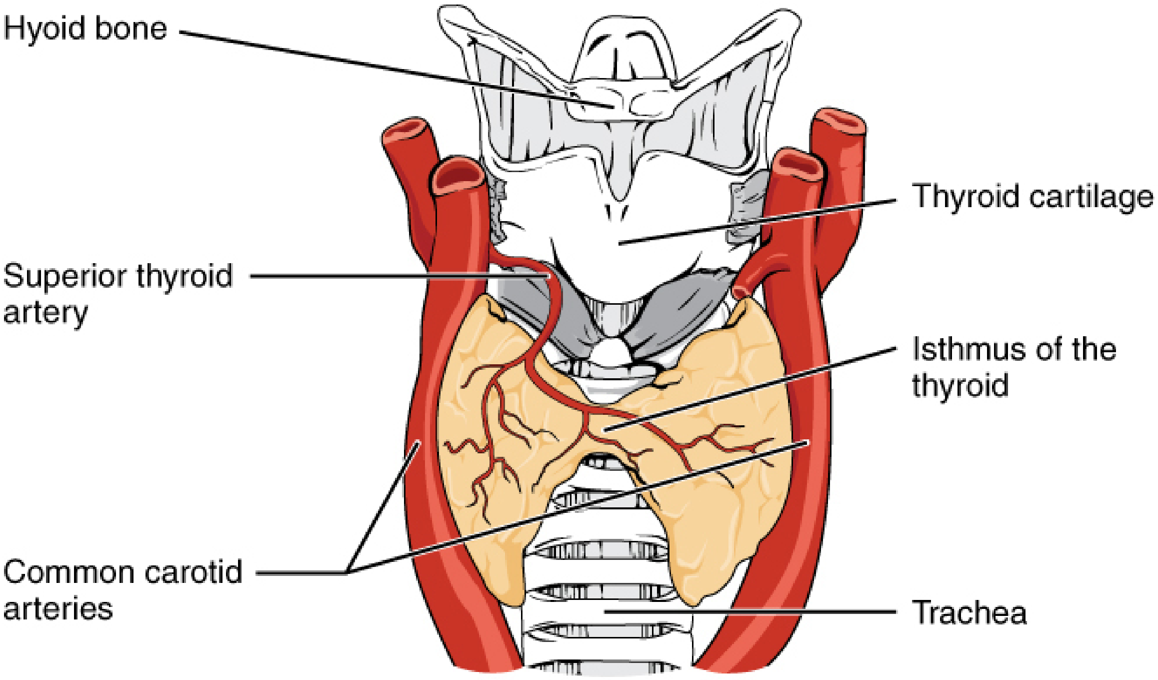


The huge advancement of information technology, system integration techniques and software development have produced a new generation of complicated computer systems. Therefore, these systems have presented big challenges to computer science engineering. A good example of these complex systems is the healthcare system. Recently, there has been an increased interest to use the advancement of communication, data mining and machine learning technologies in healthcare systems. As a result, many countries are changing the way of organising healthcare systems towards a global healthcare system across this country by setting healthcare standardisation in communication and structuring(building) the electronic healthcare records.



The Electronic Health Record (EHR) is an organised collection of electronic health data about individual patients or some populations. It is qualified for sharing across healthcare providers in a certain country. Health records normally contain a range of data including general medical records, medical history, patient examinations, laboratory results, patient treatments, radiology images, allergies, immunisation status, and some useful information for examination. Therefore, this valued information probably helps researchers in examining and diagnosing diseases by using computer techniques. The use of EHRs may help in improving the quality of care, reducing the cost of legacy systems, and mobility of records.

Therefore, our aim in the current work is to investigate the aspects of utilising the repository of health data for the benefit of humans by using machine learning. Our idea is to propose an automated method for diagnosing diseases based on previously stored data and information. However, there are many problems related to effectively using this previously obtained patient data, which could make any electronic tool less efficient. Some of these problems are: the issue of huge features or attributes and how to select the most beneficial ones, the problem of missing values and how to process that, the problem of extracting accurate diagnostic markers that can predict the early start of the disease. This thesis will try to investigate some of these issues and propose a predictive tool for thyroid gland disease diagnosis,based on the potential.



The power of automated technologies and the previous patients or data. However, the scope of the thesis is exclusive to the problems outlined above, and does not include other equally important problems like privacy and security.

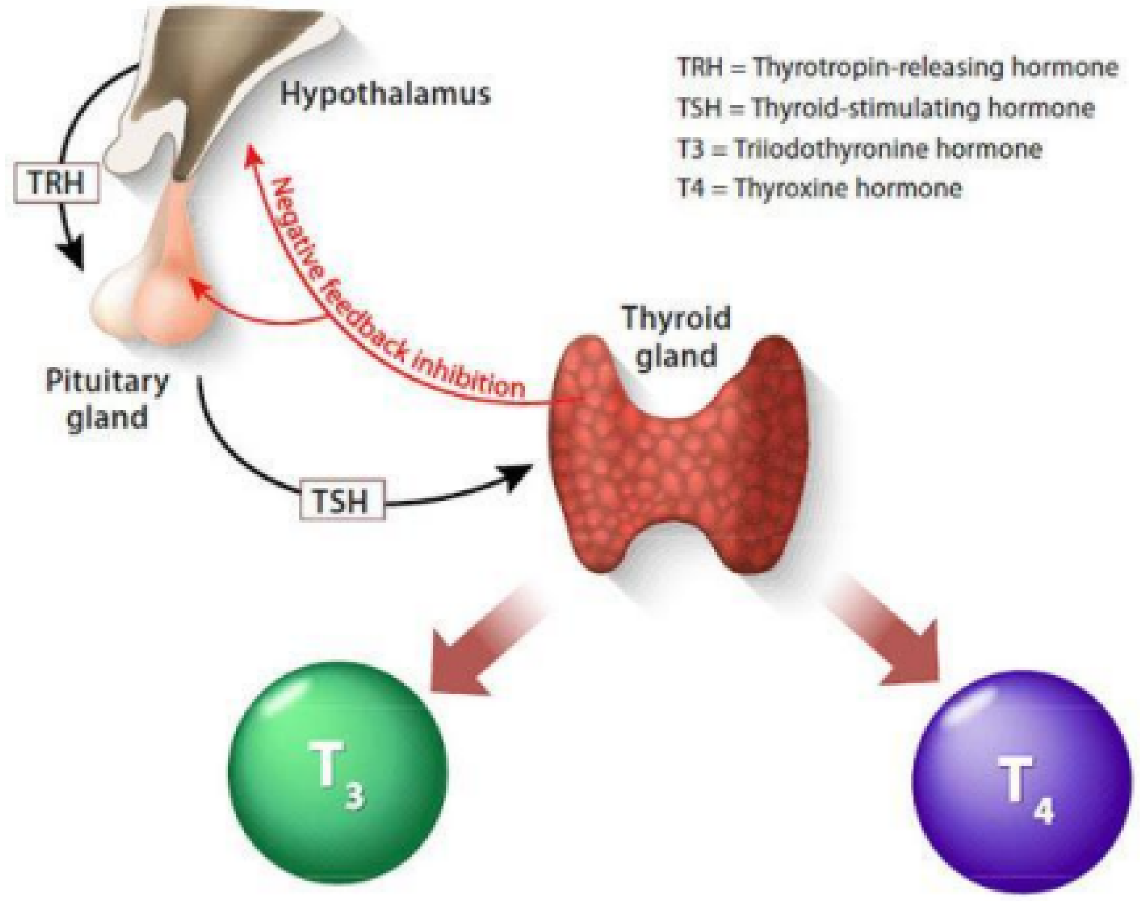

Hypothalamus
TRH = Thyrotropin-releasing hormone
TSH = Thyroid-stimulating hormone
T3 = Triiodothyronine hormone
T4 = Thyroxine hormone
TRH
Negative feedback inhibition
Thyroid gland
Pituitary gland
TSH
T_3
T_4

Figure Hormones in Thyroid Gland

In this research, UCI repository will be used as our data sources for developing automatic machine learning tools, in order to produce a useful predictive method for diagnosing thyroid gland disease. To keep track of the investigations for this project, the study used well- known datasets, which are publicly available for research purposes. It is planned that the tool developed based on decision tree algorithm techniques, and validating on this dataset can be extended to real clinical environments.

1.2 Problem Statement

Essentially an individual out of ten is experiencing thyroid sickness in India. The issue of thyroid sickness fundamentally occurs in ladies having the age of 17-54. The outrageous phase of thyroid brings about cardiovascular intricacies, expansion in pulse, boosts the cholesterol level, misery and diminished ripeness.

The chemicals, all out serum thyroxine (T4) and all out serum triiodothyronine (T3) are the two dynamic thyroid chemicals created by the thyroid organ to control the digestion of the body. For the working of every cell and each tissue and organ in the correct manner, in generally energy yield and guideline and to create proteins in the mandate of internal heat level, these chemicals are essential.

The thought for thyroid infection analysis and treatment is addressed by the utilitarian way of behaving of the thyroid illness and is the key in most thyroid sicknesses. The premise of order of thyroid infection is euthyroidism, hyperthyroidism and hypothyroidism which are meaning ordinary, over the top or flawed degrees of thyroid chemicals. The state euthyroidism portrays the typical creation of thyroid chemicals and ordinary levels at the cell level by the thyroid organ. The state hyperthyroidism is a clinical side effect because of inordinate dissemination and intracellular thyroid chemicals. The state hypothyroidism is generally because of the absence of thyroid chemical age and unfortunate substitute treatment.

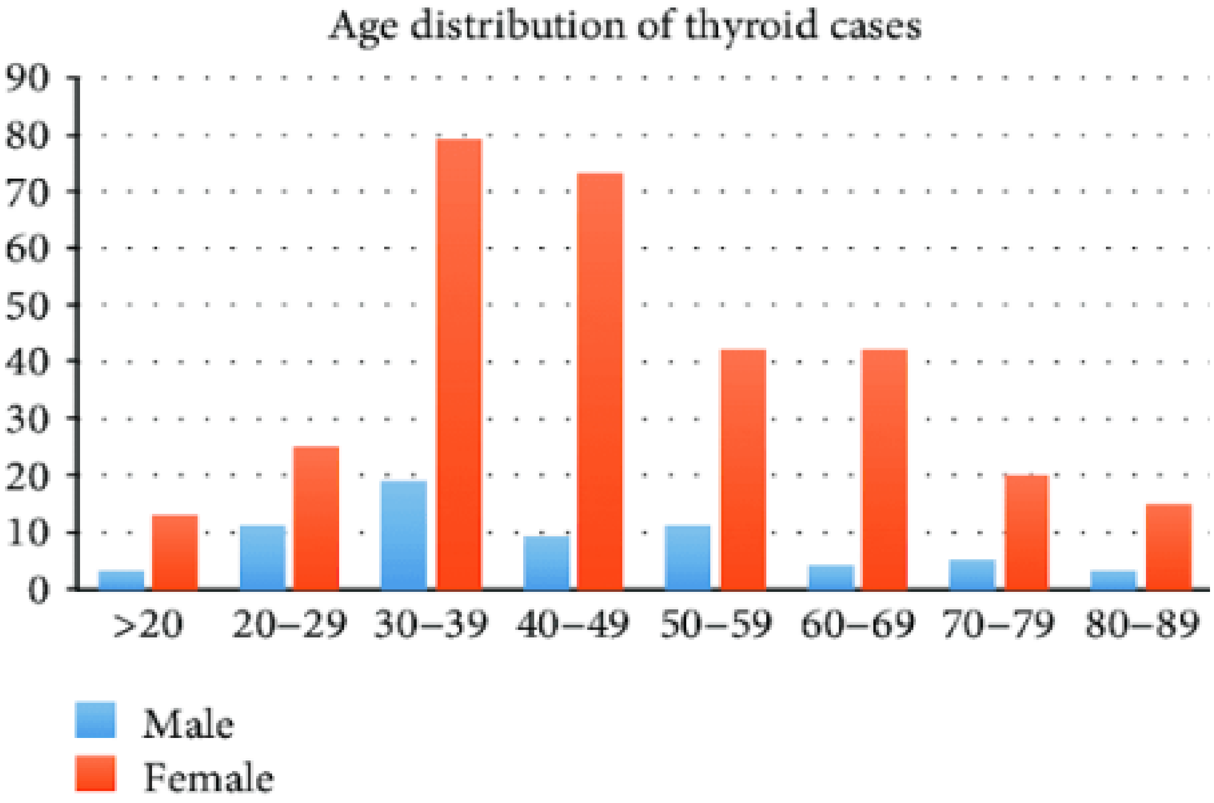


Figure 2: Age and Gender distribution of Thyroid Disease patients.

Fixing illness is a normal worry for the medical care experts, and the errorless conclusion brilliantly for a patient is vital. As of late, by some high level analysis strategies, the normal clinical report can be produced with an extra report in light of side effects. The various inquiries like "what are the reasons for influencing the thyroid?", "Which age group are impacted due to thyroid?", "what is the significant treatment for a disease?", and so on may find replies on executing AI strategies. Medical services information can be handled and after carried out with specific approaches; it can give data that can be utilised in finding and therapy of illnesses all the more proficiently and precisely with better navigation and limiting the passing gamble.

High level machine science is utilised in the space of medical care. It expected information to be gathered for clinical illness expectation. For beginning phase illness location, different clever expectation calculations are utilised. The Clinical Data Framework is great with informational collections, yet smart frameworks are not accessible for the quick finding of infections. At last, AI calculations play a vital role in taking care of complex and non-straight issues during the formation of forecast models. The attributes that can be chosen from the different informational indexes that can be utilised as depiction in a solid patient as explicitly as conceivable are required in any infection forecast models. Any other way, misclassification can bring about a decent understanding getting unseemly consideration. The truth of estimating any condition related with thyroid ailment is likewise of the best cardinal number.

Thyroid organ is endocrine in the stomach. It is raised in a brought down piece of human neck, under the Adam's apple, and helps in discharge of thyroid chemicals and which at last influences digestion rate and protein amalgamation. To control body digestion, these chemicals rely on how rapidly the heart beats and how rapidly calories are consumed. The structure of thyroid chemicals assists with controlling the body's digestion. These organs are composed of two mature levothyroxine (contracted T4) and triiodothyronine thyroid chemicals (truncated T3). These thyroid chemicals are fundamental for assembling and general development and guideline to control internal heat level. T4 and T3 are solely two actuated thyroid chemicals that generally make up thyroid organs.

The huge measure of information can be taken care of utilising AI procedures. Grouping models are appropriate for the order and qualification of the information classes. The treatment of both mathematical and straight out values should be possible by the arrangement processes. Order is a two-venture characterization model in sync one, in light of some preparation information a model is built, and in sync two an obscure tuple is given to the model to group into a class mark.

In human existence, the grouping has an incredible impact. The examination of various characterization strategies is non-insignificant and has an incredible reliance on the informational index properties. In the measurements local area, strategic relapse, choice tree and k-closest neighbour have a regarded position for arrangement issues.

The strategies for order utilised are the notable techniques. To zero in on the above-talked about issues, this report makes sense of the utilisation of three characterization AI calculations: strategic relapse grouping, choice tree order and closest neighbours arrangement to arrange individuals pruned by thyroid infection utilising the thyroid illness data set. The report makes sense exhaustively about the planning, preparing and testing of the information, bit by bit portrayal of every one of the strategies utilised, and an examination of the exactness of the techniques utilised in the expectation.

1.3 Objectives

The objectives of this project work are : -

i). To find a new way to utilise patient's histories, health information, and databases for detecting and diagnosing diseases, also provide predictive tools as medical professionals. This research is expected to establish an app that can assist physicians in diagnosing diseases and classifying patients in useful patterns based on different attributes, and how machine-learning techniques can be effective to identify such patterns. This can help in discovering early onset of the disease, treatment plans and identification of disease stages.

b) To deal with a large number of features and attributes in the dataset, and identify the significance of some features over others. However, a large number of features can lead up to the curse of dimensionality, also could render a machine learning algorithm or technique limited in terms of accuracy, specificity and precision.

c) To address an important issue related to making up a GUI combined with a machine learning algorithm, that can play an important role in determining the acceptability and ease of use achieved by designing technologies and machine learning algorithms.

1.4 Methodology

The first stage of this investigation is data collection. The data collection approach we choose is determined by our overall study, goals, objectives, feasibility, and resource constraints. The collected data is next reviewed in order to prepare it for the model selection phase. Data Preprocessing is a data mining technique used to alter raw data obtained from various sources into clearer data that is more suitable for work. Raw data may have missing or inconsistent numbers, as well as a large amount of duplicate information. During this step of data processing, we may check for missing data, erroneous data, and outliers in the data collection, as well as the lack of data constraints. The lack of data usually results in inconsistent data. This results in variances, which we must manage with prior to analysis. Following that, the data is divided into train and test datasets. The train-test split is an approach for assessing machine learning performance systems. It may be used to solve classification or regression problems and can be used in any situation Supervised learning technique. The procedure entails splitting a dataset into subsets. two divisions The training subset is the first subset that is utilised to fit the model. dataset. The input element of the second subset is utilised instead of the second subset to train the model. The dataset is fed into the model, which generates predictions and compares them to the data. The expected results the test dataset is the name given to the second dataset.

The objective is to estimate how well the machine learning model performs on fresh data: not used for training the prototype Following that, a feature selection technique is performed. The feature choice is the method of developing a predictive model while minimising the number of input variables. To both save time and money, the number of input variables should be kept to a minimum. Modelling and, in certain cases, improving the model's performance It evaluates the employing statistics, determines the link between each input variable and the target variable and picks those input variables having the strongest relationship to the target variable. The dataset must only contain numerical properties in order to be classified.

Here, the Figure 1 shows our workflow Architecture.

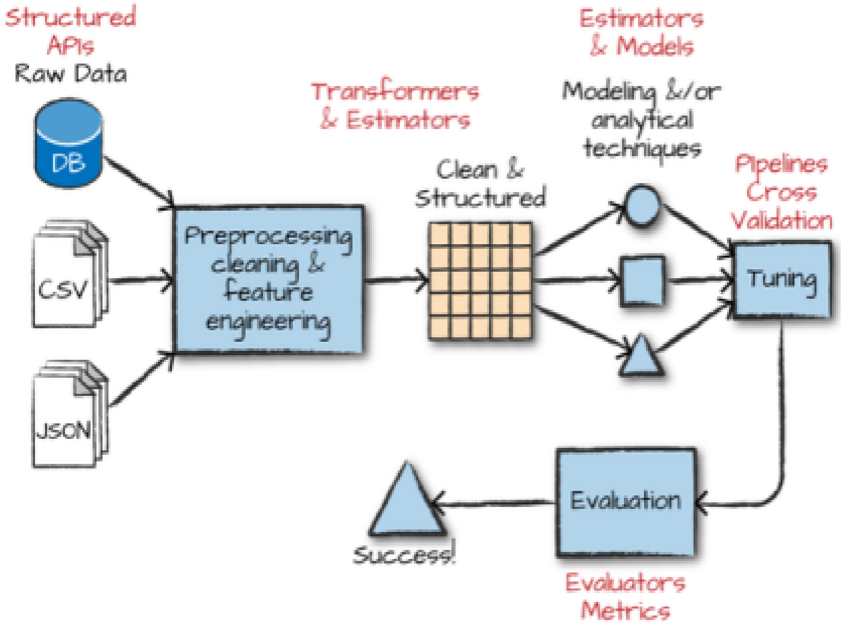


Figure 1: Workflow Architecture.

1.5 Organisation

The workflow Diagram of my project “Thyroid Detection Using Machine Learning” is shown below. Our bigger problem is to create a model or to train a model which is going to predict whether a certain wafer is faulty or not. So we are breaking this bigger problem into a smaller problem e.g; How to read Data, then how to validate the data, then how to do data pre-processing steps, how to train a mode and so on… By combining these small steps we are going to give bigger solutions.

1.6 Scope

The scope of this project is to build a classification methodology to predict the type of thyroid based the given training dataset. Either a person suffering from thyroid or not, it can be a positive case or negative case ; if it is a positive case then what type of case it’s?. What types of thyroid a person is suffering from?. So, at the first level of specialisation my model will predict that a person is suffering from thyroid or not, whether it’s a Hypothyroid or Hyperthyroid, it will predict whether the person is suffering from it or not.

If the result is found to be positive then a treatment of that person will be a bit fast-track, a doctor will come and give him attention and they will treat their treatment on priority.

But let’s say if the result is coming negative then what will happened is report of that particular patients it will go to the junior doctor, then that junior doctor with their own expertise they will identify okay weather the model has done correct prediction or not; mostly the model does correct prediction , but still we can’t directly go ahead and move the human intervention.

If the model has made a correct prediction and the junior doctors agree that the person is not suffering from any thyroid disease or thyroid symptoms, they will release the patient. But again based on the reading and different tests, the doctor can conclude okay, the person might be suffering from the thyroid disease. Again the report of that patient will again be sent to the senior doctor and they will treat the patient, so this is how the mechanism is going to work.

1.7 Language Used

The project is built on PyCharm which is an integrated development environment (IDE) used in computer programming, specifically for the Python language and Anaconda which is a distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics , etc.), that aims to simplify package management and deployment.

This project is purely based on Python Programming Languages. Different python libraries such as pandas, numpy, matplotlib, pyplot, seaborn, resample, KNNImputer, LabelEncoder and several other python libraries are also used.

1.8 Technical Requirements

1.8.1 Hardware Requirements

Device Specification

Processor Intel(R) Celeron(R) J4005 CPU @ 2.00GHz 2.00 GHz
Installed RAM 4.00 GB (3.85 GB usable)

Device ID 5D876BDF-30FA-45D6-9F96-2D96BF4830B5
System Type 64-bit operating system, x64-based processor

1.8.2 Software Requirements

- Web Browser: Microsoft Internet Explorer, Mozilla, Google Chrome
- Operating System: Windows 10
- PyCharm Version 2021.1
- Anaconda Version 2020.11
- ipython 7.19.0
- jupyterlab 2.2.6
- matplotlib 3.3.2
- notebook 6.1.4
- numpy 1.19.2
- pandas 1.1.3
- python 3.8.5
- scikit-learn 0.23.2
- scipy 1.5.2
- sqlalchemy 1.3.20
- statsmodels 0.12.0

CHAPTER 2

LITERATURE SURVEY

2.1. Literature review on diagnosis of thyroid diseases

A study of the literature on thyroid disease diagnosis reveals that various data mining and artificial intelligence approaches such as fuzzy logic, artificial neural networks, fuzzy neural networks, support vector machines, immune recognition systems, and so on have been utilised to determine the kind of thyroid.There has been a lot of work done to diagnose the different thyroid illnesses. The authors demonstrated a sufficient method and assurance in identifying disorders similar to thyroid disease through work that involves diverse datasets and algorithms coupled with future work to get effective and improved outcomes. The purpose of this research is to interpret numerous machine learning and statistical qualities that have been popular in recent years for the interpretation of thyroid illnesses with the assurance of achieving various possibilities and methods. Machine learning methods include random forest, decision tree, naive Bayes, and SVM. The writers examined and contrasted the four classifications.

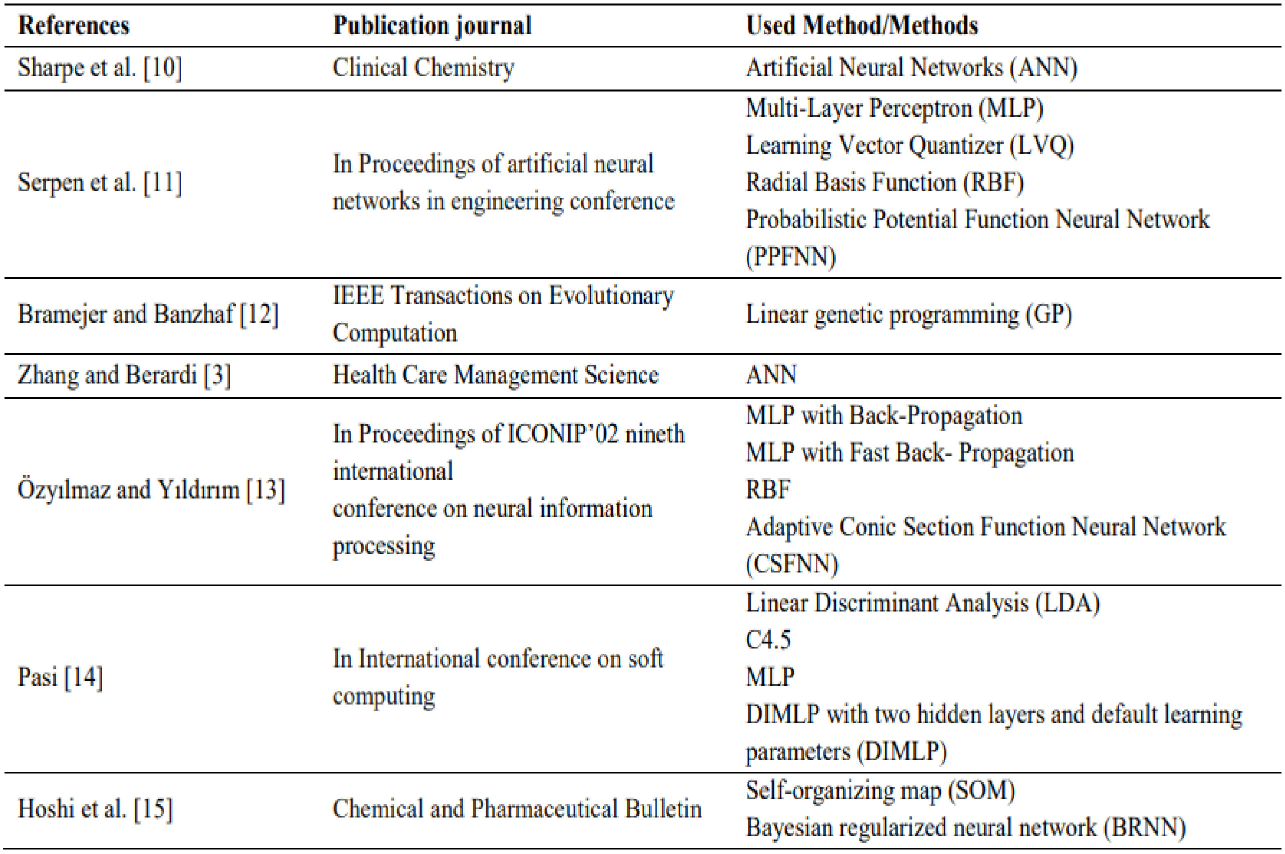
Naive Bayes, Decision Tree, Multilayer Perceptron, and Radial Basis Function Network are the four categorization models. The conclusion reveals that all categorization models are quite accurate. The Decision Tree classification model outperforms the other classification models. In this study, 29 dataset attributes are conscripted and imposed as a Feature Selection approach, namely Chi-Square, The datasets are filtered by applying unsupervised coated filters to the attributes to convert continuous values to nominal values, reducing the 29 attributes to ten.

Machine learning (ML) is a subset of artificial intelligence that is infiltrating scientific study at an increasing rate. Machine learning enables algorithms to review from experience without being explicitly prioritised. This has induced machine learning. An advanced mixed contemporary data science strategy to harnessing the powers of cultivated data is input detonation, which is linked with a growing computing capability.

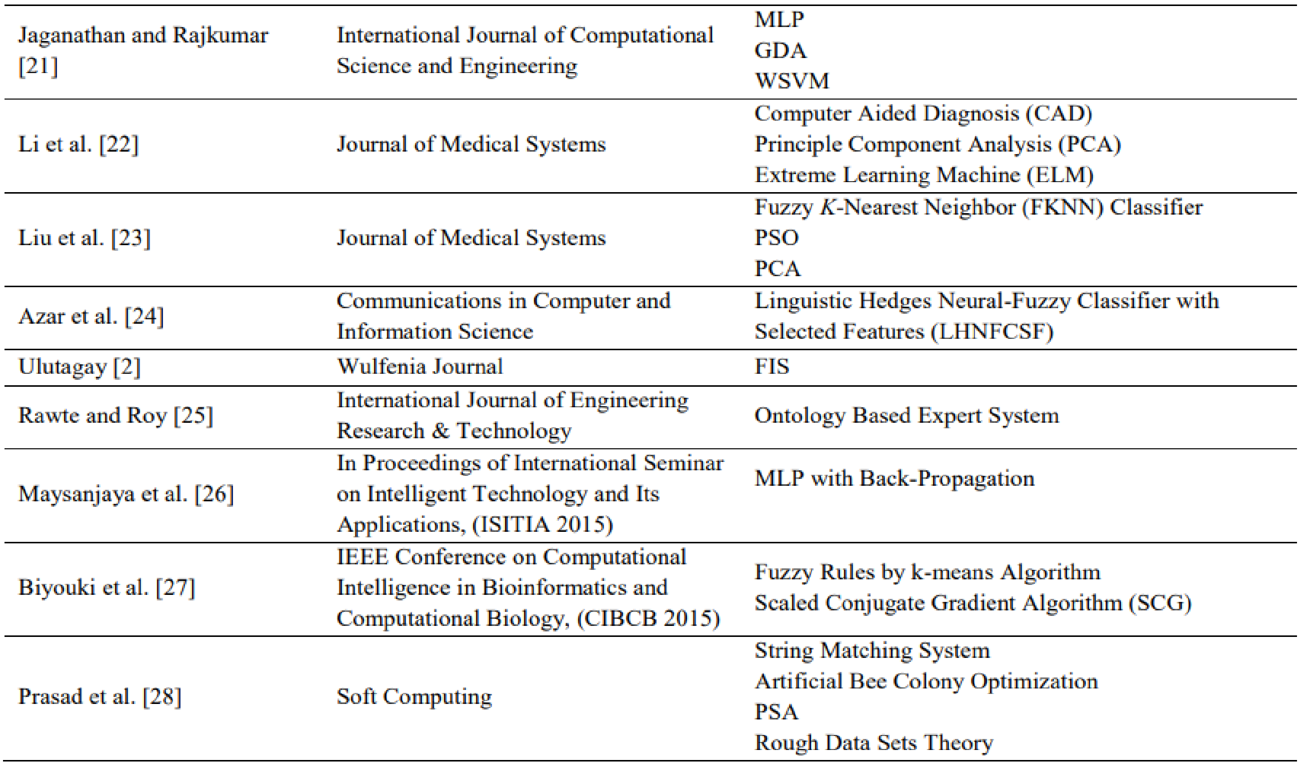
2.1.1. Related Articles and Research Papers

The table shows the Articles published in accordance with the study of Thyroid disease diagnosis with respect to the used methods.

Table 1: Published articles in thyroid diseases diagnosis according to used method/methods



References	Publication journal	Used Method/Methods
Sharpe et al. [10]	Clinical Chemistry	Artificial Neural Networks (ANN)
Serpen et al. [11]	In Proceedings of artificial neural networks in engineering conference	Multi-Layer Perceptron (MLP) Learning Vector Quantizer (LVQ) Radial Basis Function (RBF) Probabilistic Potential Function Neural Network (PPFNN)
Bramejer and Banzhaf [12]	IEEE Transactions on Evolutionary Computation	Linear genetic programming (GP)
Zhang and Berardi [3]	Health Care Management Science	ANN
Özyılmaz and Yıldırım [13]	In Proceedings of ICONIP'02 nineth international conference on neural information processing	MLP with Back-Propagation MLP with Fast Back- Propagation RBF Adaptive Conic Section Function Neural Network (CSFNN)
Pasi [14]	In International conference on soft computing	Linear Discriminant Analysis (LDA) C4.5 MLP DIMLP with two hidden layers and default learning parameters (DIMLP)
Hoshi et al. [15]	Chemical and Pharmaceutical Bulletin	Self-organizing map (SOM) Bayesian regularized neural network (BRNN)



Jaganathan and Rajkumar [21]	International Journal of Computational Science and Engineering	MLP GDA WSVM
Li et al. [22]	Journal of Medical Systems	Computer Aided Diagnosis (CAD) Principle Component Analysis (PCA) Extreme Learning Machine (ELM)
Liu et al. [23]	Journal of Medical Systems	Fuzzy *K*-Nearest Neighbor (FKNN) Classifier PSO PCA
Azar et al. [24]	Communications in Computer and Information Science	Linguistic Hedges Neural-Fuzzy Classifier with Selected Features (LHNFCSF)
Ulutagay [2]	Wulfenia Journal	FIS
Rawte and Roy [25]	International Journal of Engineering Research & Technology	Ontology Based Expert System
Maysanjaya et al. [26]	In Proceedings of International Seminar on Intelligent Technology and Its Applications, (ISITIA 2015)	MLP with Back-Propagation
Biyouki et al. [27]	IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology, (CIBCB 2015)	Fuzzy Rules by k-means Algorithm Scaled Conjugate Gradient Algorithm (SCG)
Prasad et al. [28]	Soft Computing	String Matching System Artificial Bee Colony Optimization PSA Rough Data Sets Theory

2.2 Thyroid disease treatment prediction with machine learning approaches.

Publisher : Procedia Computer Science

Author : Aversano et al. (2021)

They used a number of machine learning methods to analyse data. They specifically compared the output of 10 different classifiers. The other algorithms' performance is encouraging, notably the Extra-TreeClassifier, which obtains an accuracy of 84%. In addition, they used a catboost classifier and attained a precision of 71%.

2.3 Automated segmentation of thyroid nodule, gland, and cystic components from ultrasound images using deep learning

Publisher : IEEE Access

Author : Kumar (2020)

They used SVM to diagnose thyroid stage with an accuracy of 83.37 percent. Other classification techniques have been shown to be less efficient than Multiclass SVM. Furthermore, the model accurately differentiates between the four thyroid states. The above discussion demonstrates that ANN, CatBoost, XGBoost, Random Forest, LightGBM, Decision Trees and Extra Trees, as well as SVC, KNN, and Naive Bayes, outperform SVC, KNN, and Naive Bayes on numerous common multiclass datasets. In terms of accuracy and overall performance, these algorithms exceed all others.

2.4 Predictive model based on neural networks to assist the diagnosis of malignancy of thyroid nodules

Publisher : Proceedings of the 41st international conference on computers & industrial engineering

Author : Bastias et al. (2011)

An expert approach for detecting thyroid illness, was presented by Bastias et al. (2011), which was radial basis function (RBF), and adaptive conic section function (CSFNN). However, because of technological restrictions, it is only suited for limited data sets and cannot be employed in operation. set out to create a machine learning classifier capable of identifying health issues and investigating the capabilities of the recommended classifier The suggested classifier significantly assisted in the diagnosis of thyroid gland illnesses.

2.5 Interactive Thyroid Disease Prediction System Using Machine Learning Techniques

Publisher : 5th IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC-2018)

Author : Ankita Tyagi and Rikitha Mehra

They apply many categorization techniques in their paper, including Decision Tree, Artificial Neural Network, and the k-Nearest-Neighbour algorithm. Based on the data set collected from the UCI Repository, classification and prediction were done, and accuracy was determined based on the output provided. They analysed the accuracy of the algorithms employed and compared them to determine the optimum strategy with high accuracy.

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Thyroid Disease Dataset

Website : UCI
Repository(*https://archive.ics.uci.edu/ml/datasets/thyroid+disease*)
Data Set Characteristics : Multivariate, Domain-Theory
Number Of Instances : 7200
Area : Life
Attribute Characteristics : Categorical, Real
Number Of Attributes : 21
Date Donates : 1987-01-01
Associated Tasks : Classification
Missing Values : N/A
Number Of Web Hits : 268078
File Formats : hypothyroid.csv

3.1.1 Data Set Features

This is a thyroid informational index obtained from the UCI (University of California, Irvine, School of Data and Software engineering). It is an assortment of data sets, area hypotheses, and information generators utilised by the AI people group to direct accurate assessments of AI calculations. This was made to help concentrating by using state of the art advances.

This study made use of a dataset from the UCI repository. It has 7200 instances and three classes, including 3772 training instances, 3428 testing instances, and 21 characteristics. The goal is to determine if a certain patient is normal or has hyperthyroidism or hypothyroidism.

3.1.2 Description of Data Set

The data set consists of : -

Number of Instances : 7200

Number of Attribute : 30

The features are classified into different categories : -

age - Age of the person,

sex - Male or Female,

on_thyroxine - true or false,

on_antithyroid_medication - true or false,

sick - true or false,

pregnant - true or false,

thyroid_surgery - true or false,

I131_treatment - true or false,

query_hypothyroid - true or false,

lithium - true or false,

goitre - true or false,

tumour - true or false,

hypopituitary- true or false,

psych - true or false,

TSH_measured - true or false,

TSH - thyroid stimulating hormone floating value,

T3_measured - true or false,

T3 - triiodothyronine value,

TT4_measured- true or false,

TT4 - Thyroxine value,

T4U_measured- true or false,

T4U - numerical value,

FTI_measured true or false,

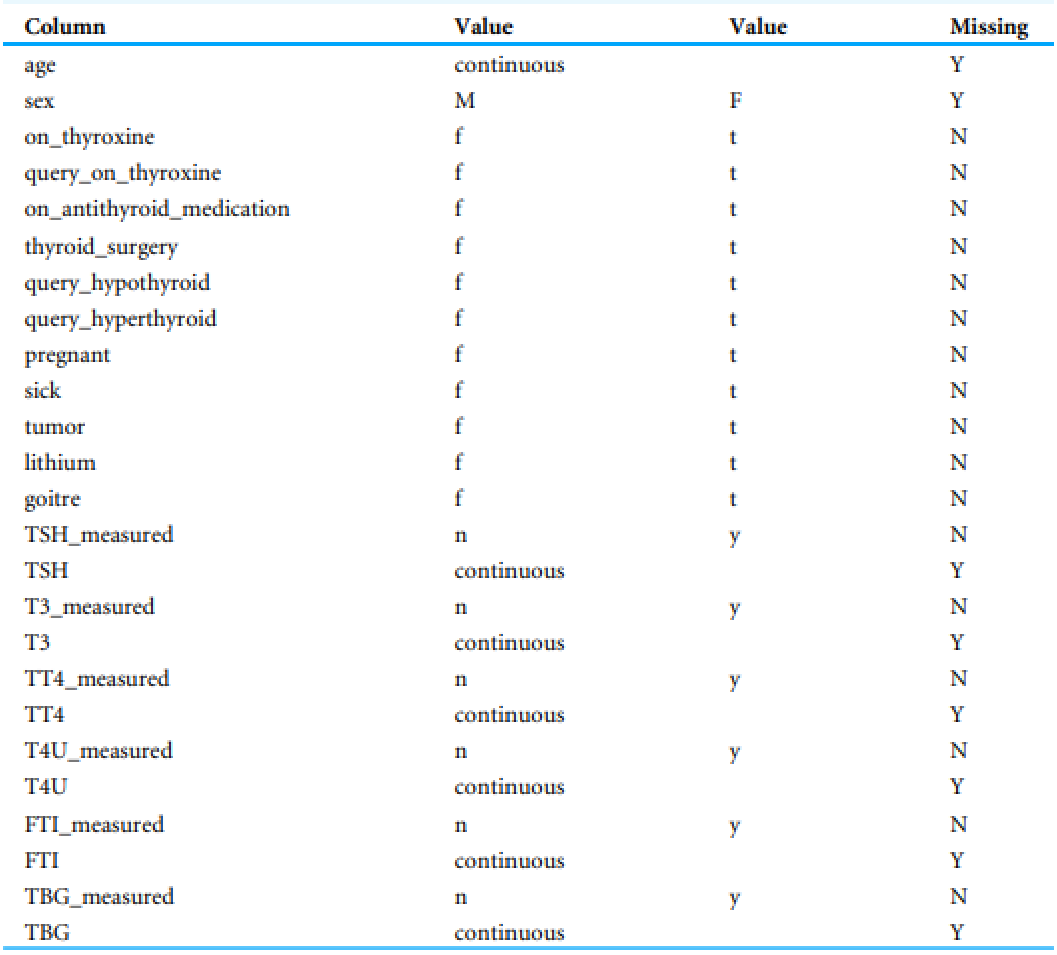
FTI -Free Thyroxine Index,

TBG_measured- true or false,

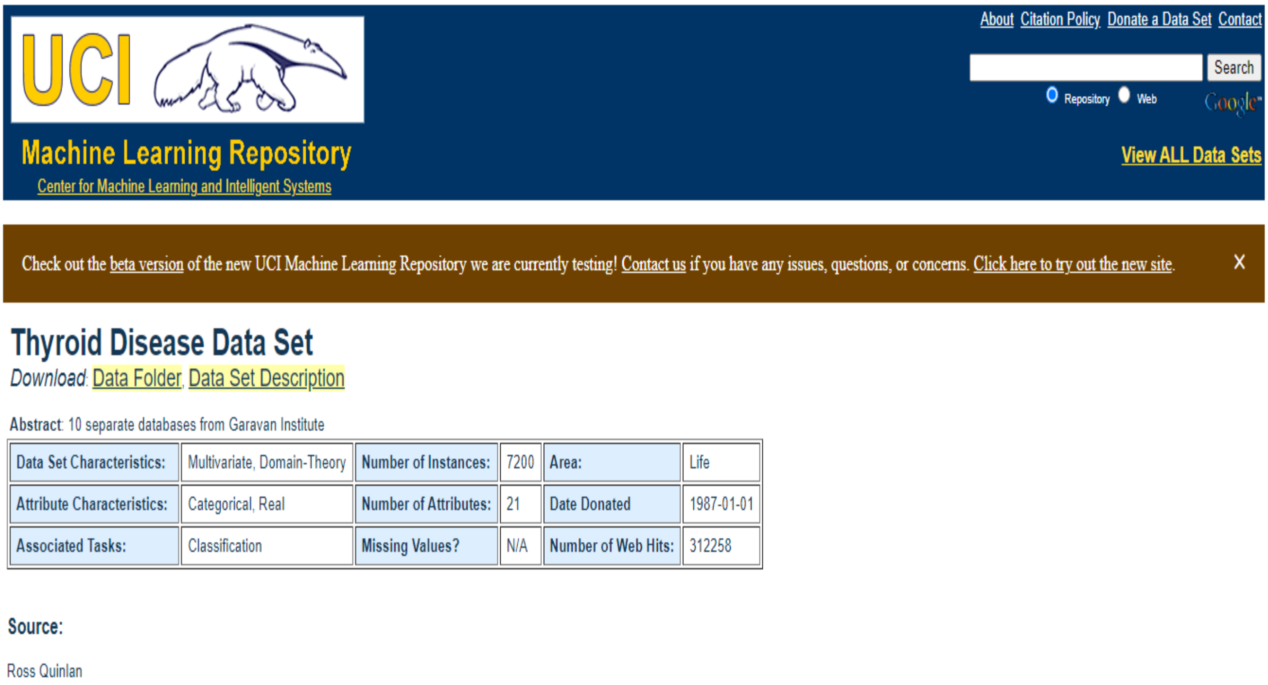
TBG -Thyroid-Binding Globulin value,

referral_source - different sources of referrals,

Class - different types of thyroid.



Column	Value	Value	Missing
age	continuous		Y
sex	M	F	Y
on_thyroxine	f	t	N
query_on_thyroxine	f	t	N
on_antithyroid_medication	f	t	N
thyroid_surgery	f	t	N
query_hypothyroid	f	t	N
query_hyperthyroid	f	t	N
pregnant	f	t	N
sick	f	t	N
tumor	f	t	N
lithium	f	t	N
goitre	f	t	N
TSH_measured	n	y	N
TSH	continuous		Y
T3_measured	n	y	N
T3	continuous		Y
TT4_measured	n	y	N
TT4	continuous		Y
T4U_measured	n	y	N
T4U	continuous		Y
FTI_measured	n	y	N
FTI	continuous		Y
TBG_measured	n	y	N
TBG	continuous		Y


UCI
About Citation Policy Donate a Data Set Contact
Search
○ Repository ● Web
Google
Machine Learning Repository
Center for Machine Learning and Intelligent Systems
View ALL Data Sets
X

Check out the beta version of the new UCI Machine Learning Repository we are currently testing! Contact us if you have any issues, questions, or concerns. Click here to try out the new site.

Thyroid Disease Data Set

Download: Data Folder, Data Set Description

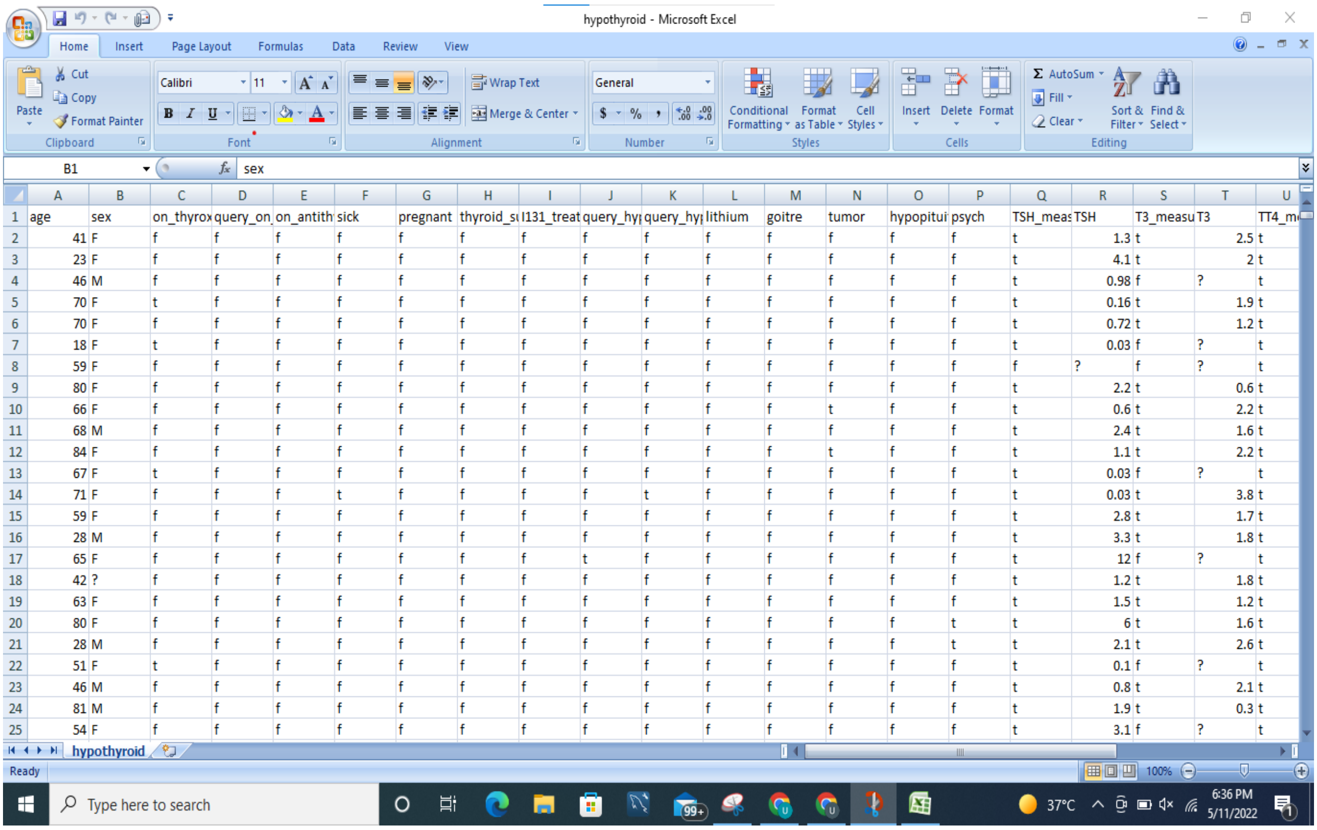
Abstract: 10 separate databases from Garavan Institute

Data Set Characteristics:	Multivariate, Domain-Theory	Number of Instances:	7200	Area:	Life
Attribute Characteristics:	Categorical, Real	Number of Attributes:	21	Date Donated	1987-01-01
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	312258

Source:

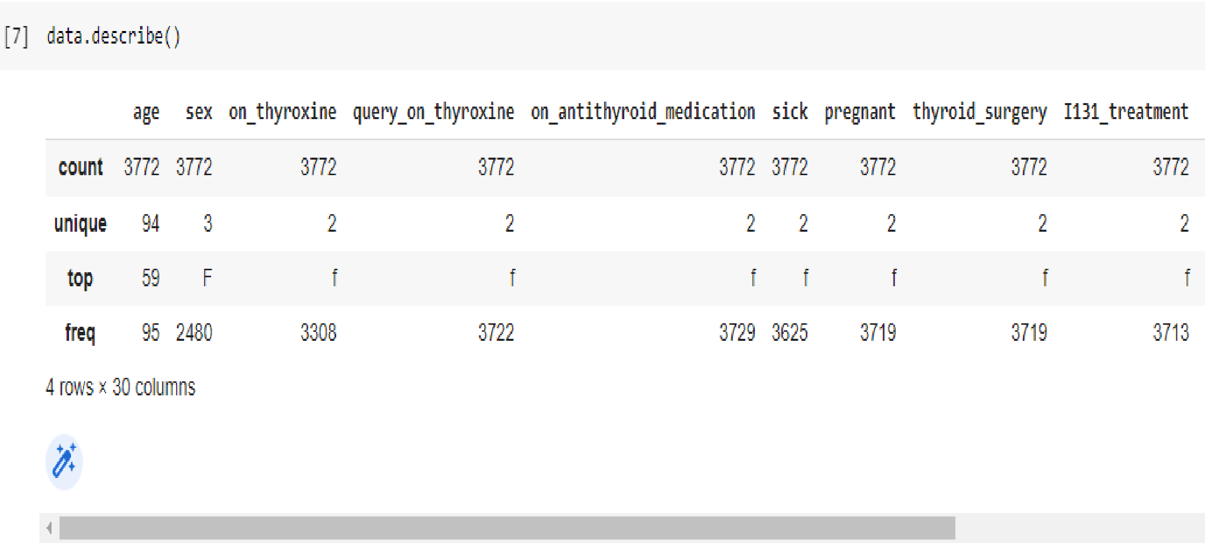
Ross Quinlan

3.1.3 Dataset Contents



B1 sex

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	age	sex	on_thyrox	query_on_	on_antith	sick	pregnant	thyroid_su	I131_treat	query_hyp	query_hyp	lithium	goitre	tumor	hypopitui	psych	TSH_meas	TSH	T3_measu	T3	TT4_m
2	41	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.3	t	2.5	t
3	23	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	4.1	t	2	t
4	46	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.98	f	?	t
5	70	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.16	t	1.9	t
6	70	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.72	t	1.2	t
7	18	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.03	f	?	t
8	59	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	f	?	f	?	t
9	80	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.2	t	0.6	t
10	66	F	f	f	f	f	f	f	f	f	f	f	f	t	f	f	t	0.6	t	2.2	t
11	68	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.4	t	1.6	t
12	84	F	f	f	f	f	f	f	f	f	f	f	f	t	f	f	t	1.1	t	2.2	t
13	67	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.03	f	?	t
14	71	F	f	f	f	t	f	f	f	f	t	f	f	f	f	f	t	0.03	t	3.8	t
15	59	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	2.8	t	1.7	t
16	28	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	3.3	t	1.8	t
17	65	F	f	f	f	f	f	f	f	t	f	f	f	f	f	f	t	12	f	?	t
18	42	?	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.2	t	1.8	t
19	63	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.5	t	1.2	t
20	80	F	f	f	f	f	f	f	f	f	f	f	f	f	f	t	t	6	t	1.6	t
21	28	M	f	f	f	f	f	f	f	f	f	f	f	f	f	t	t	2.1	t	2.6	t
22	51	F	t	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.1	f	?	t
23	46	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	0.8	t	2.1	t
24	81	M	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	1.9	t	0.3	t
25	54	F	f	f	f	f	f	f	f	f	f	f	f	f	f	f	t	3.1	f	?	t



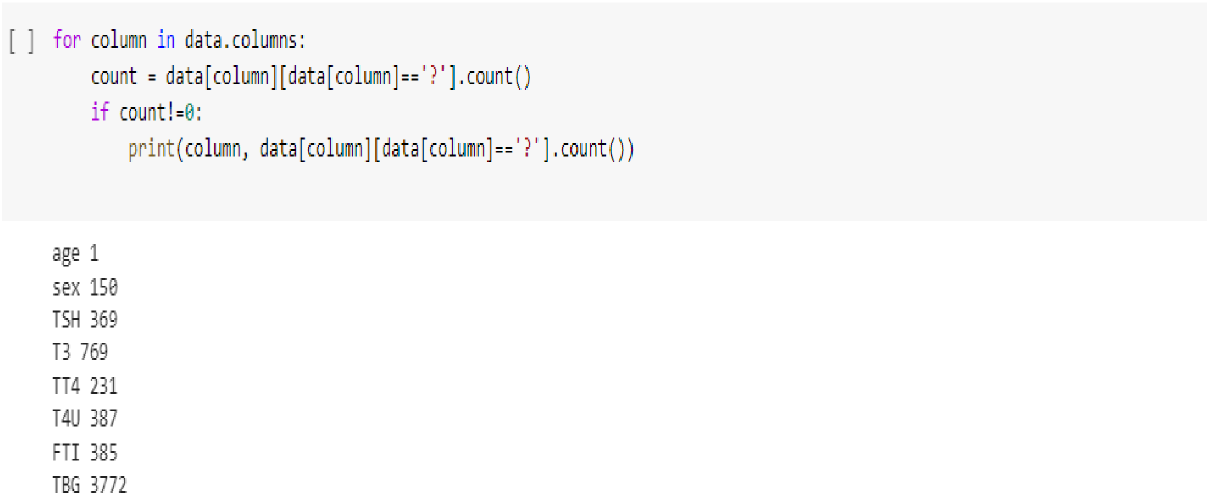
```
[7] data.describe()
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_medication	sick	pregnant	thyroid_surgery	I131_treatment
count	3772	3772	3772	3772	3772	3772	3772	3772	3772
unique	94	3	2	2	2	2	2	2	2
top	59	F	f	f	f	f	f	f	f
freq	95	2480	3308	3722	3729	3625	3719	3719	3713

4 rows × 30 columns

3.1.4 Dataset Missing Values

We can see from the data description that there are no missing values. But if you check the dataset the missing values are replaced with invalid values like '?'. Therefore, to balance the dataset, I replaced such values with 'nan' and check for missing values again in the dataset.



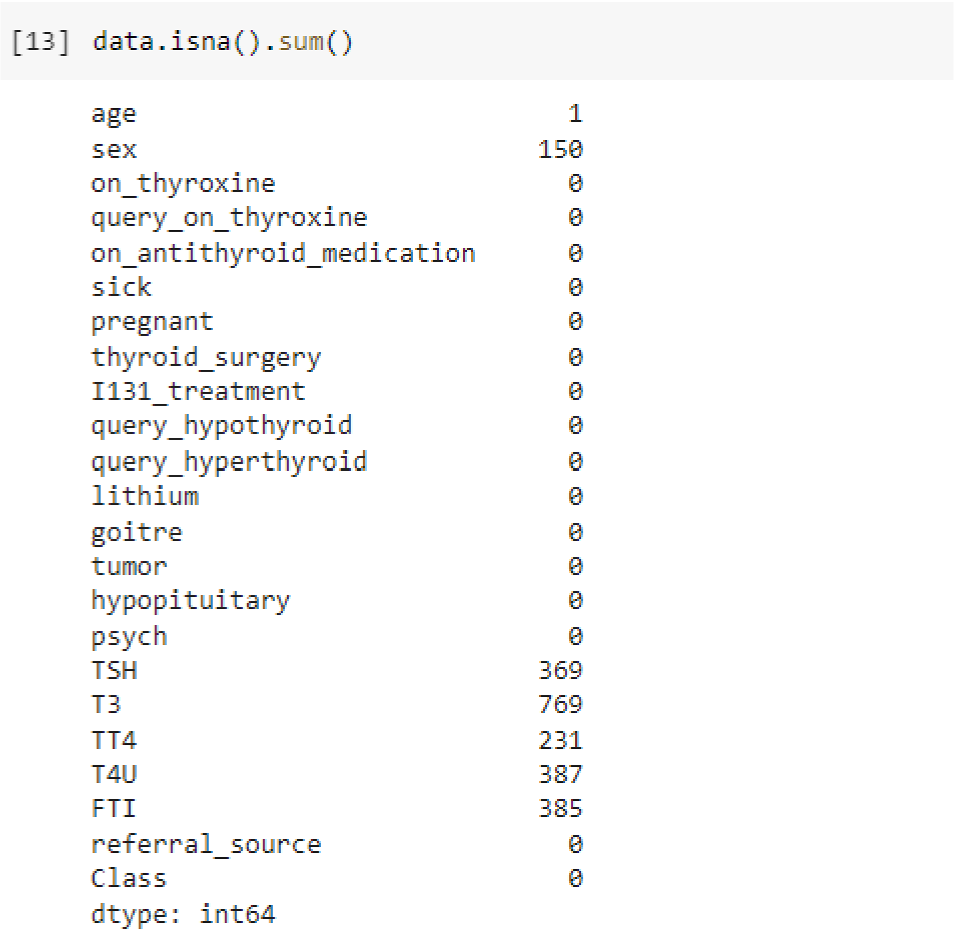
```
[ ] for column in data.columns:
        count = data[column][data[column]=='?'].count()
        if count!=0:
            print(column, data[column][data[column]=='?'].count())

    age 1
    sex 150
    TSH 369
    T3 769
    TT4 231
    T4U 387
    FTI 385
    TBG 3772
```

3.1.5 Data Preprocessing

After replacing all such values with 'nan'. Now moving forward we will deal with these missing values now.

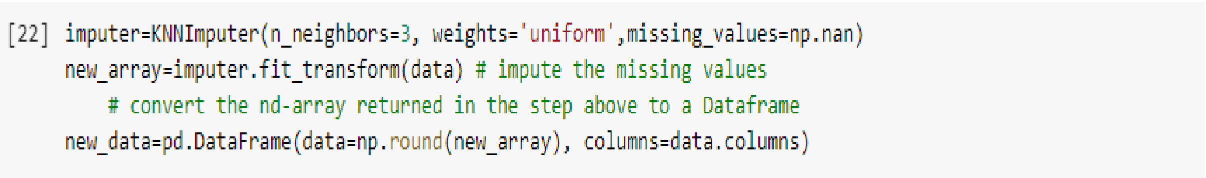
Since the values are categorical, we have to change them to numerical before we use any imputation techniques.

We can use get dummies but since most of the columns have only two distinct categories we will use mapping for them. Why? Because since there are only two categories then the two columns formed after get_dummies will both have very high correlation since they both explain the same thing. So in any case we will have to drop one of the columns. That's why let's use mapping for such columns. For columns with more than two categories we will use get dummies.

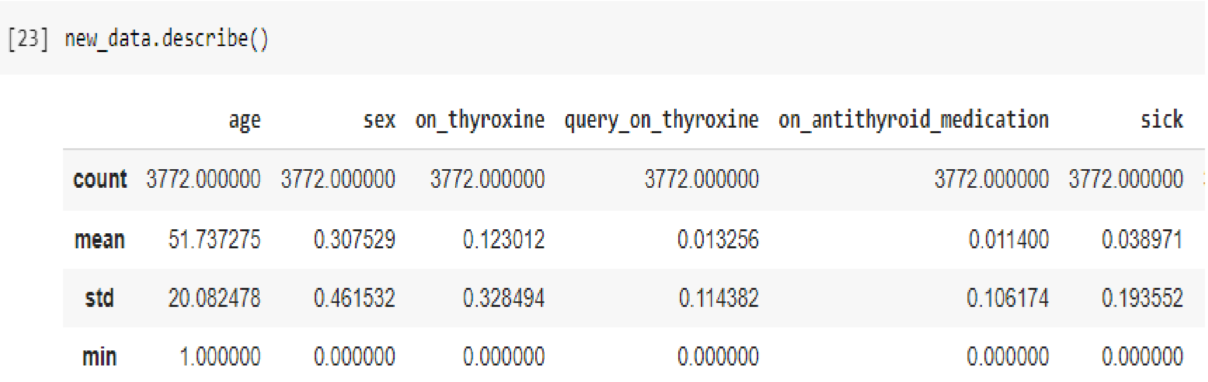


```
[13] data.isna().sum()

     age                            1
     sex                          150
     on_thyroxine                   0
     query_on_thyroxine             0
     on_antithyroid_medication      0
     sick                           0
     pregnant                       0
     thyroid_surgery                0
     I131_treatment                 0
     query_hypothyroid              0
     query_hyperthyroid             0
     lithium                        0
     goitre                         0
     tumor                          0
     hypopituitary                  0
     psych                          0
     TSH                          369
     T3                           769
     TT4                          231
     T4U                          387
     FTI                          385
     referral_source                0
     Class                          0
     dtype: int64
```

Here, we have applied KNN Imputer in order to impute the missing values present in the Dataset, in this the KNN Imputer is what it basically does is it takes the arithmetic average of the above present 3 neighbours from the missing value.

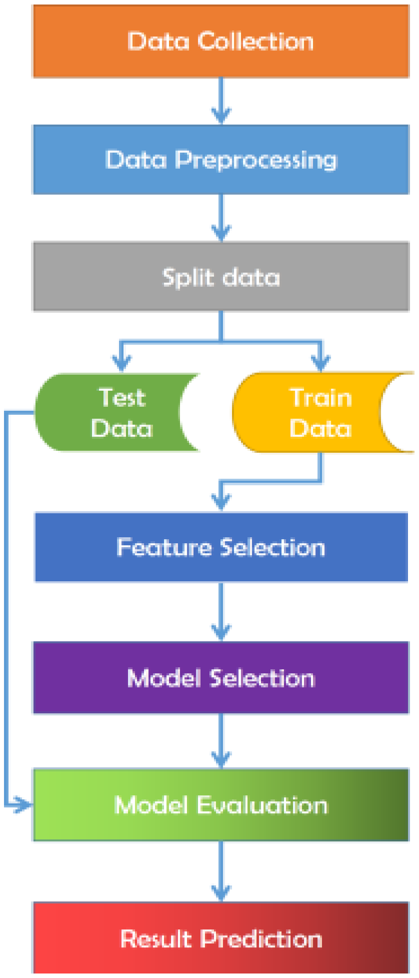


```
[22] imputer=KNNImputer(n_neighbors=3, weights='uniform',missing_values=np.nan)
     new_array=imputer.fit_transform(data) # impute the missing values
        # convert the nd-array returned in the step above to a Dataframe
     new_data=pd.DataFrame(data=np.round(new_array), columns=data.columns)
```



```
[23] new_data.describe()
```

	age	sex	on_thyroxine	query_on_thyroxine	on_antithyroid_medication	sick
count	3772.000000	3772.000000	3772.000000	3772.000000	3772.000000	3772.000000
mean	51.737275	0.307529	0.123012	0.013256	0.011400	0.038971
std	20.082478	0.461532	0.328494	0.114382	0.106174	0.193552
min	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000



3.1.6 Handling Missing Values

RandomOverSampler

In undersampling we are basically losing out on important information. How do we tackle that by using something called over sampling. In oversampling, we kind of increase the number of samples of the minority class to match up to the number of samples in the majority class.

The basic fundamental idea of oversampling is you take the minority class and try to create new samples that could match up to the length of my majority class samples. What I mean by that is if you have 100 samples in your minority class and Ten thousand samples in your majority class, then you should increase your minority class sample. From hundred to say nine thousand or ten thousand so that the length of both your majority & minority class add up. That's the bare minimum idea of how oversampling is achieved.

There are various means of doing it one of them is random over sampling and the other is SMOTE wherein you create artificial samples of your minority class we'll first look at random over sampling.

So the way random over sampling works is you have your minority class and you have 100 samples and you have your majority class. You have a thousand samples. You will pick a row randomly from this minority class and put it into the data set. So, I will first pick up say row number 99, put it up here again. I'll do this iteratively so that I match the total number of points which are there in my majority class. That is how random over sampling works.

Now comes the next method, which is called SMOTE. So the full form of SMOTE is Synthetic Minority Oversampling Technique Say, for example, you have two features. X1 and X2 We have more number of black samples compared to blue samples so there is a class imbalance that is there now. Rather than decreasing the black samples I'll be increasing the blue samples, which are my minority class samples by using SMOTE. How does this work is what I'll explain so, I'll magnify the minority class samples in the next sheet. So, I had four points in my minority class.

3.3 System Design

3.3.1 Data Validation

In the Data Validation Process, we intend to perform the different necessary steps of data validation on particular sets on the given set of training files.

3.3.1.1 Name Validation

The names of the files are checked against the schema file's name. To utilise for validation, we constructed a regex pattern using the name specified in the schema file. We check for the length of date in the file name as well as the length of time in the file name after verifying the pattern in the name. We transfer such files to "Good Data Folder" if all of the values are correct; otherwise, we move them to "Bad Data Folder."

3.3.1.2 Column Numbers

In this step, We are going to validate the number of columns present in the files, and if it doesn't match with the value given in the schema file, then the file is moved to "Bad_Data_Folder."

3.3.1.3 Name of the Columns

The name of the columns is validated and should be the same as given in the schema file. If not, then the file is moved to "Bad_Data_Folder".

3.3.1.4 Datatype of Columns

The datatype of columns is given in the schema file. This is validated when we insert the files into Database. If the datatype is wrong, then the file is moved to "Bad_Data_Folder".

3.3.1.5 Null values in columns

In this step, we check if any of the columns in a file have all the values as NULL or missing, we discard such a file and move it to "Bad_Data_Folder".

3.3.2 Insertion of data in the Database

3.3.2.1 Creating Database and Database connection

In this step, we created a database with the given name passed. If the database is already created, open the connection to the database.

3.3.2.2 Table creation in the database

In the database, a table named "Good_Data" is created for adding files into the "Good_Data _Folder" based on column names and data types specified in the schema file. If the table already exists, no new table is generated, and new files are added into the existing database, because we want to train on both new and old training files.

3.3.2.3 Insertion of files in the table

The above-created table contains all of the files in the "Good_Data_Folder." If any of the columns in a file have an incorrect data type, the file is not loaded into the database and is transferred to the "Bad_Data_Folder."

3.3.3 Model Training

3.3.1 Exporting data from Database

In this step, we will be exporting the data stored in the database to a CSV file which will be required or further used for the model training purpose.

3.3.2 Data Preprocessing

a) Drop the columns if they are useless for the model training purpose. These columns were chosen during the EDA.

b) Replace incorrect values with numpy "nan" so that the imputer may be used on them.

c) Write down the category values.

d) Inspect the columns for null values. If the null values are present, use the KNN imputer to compute them.

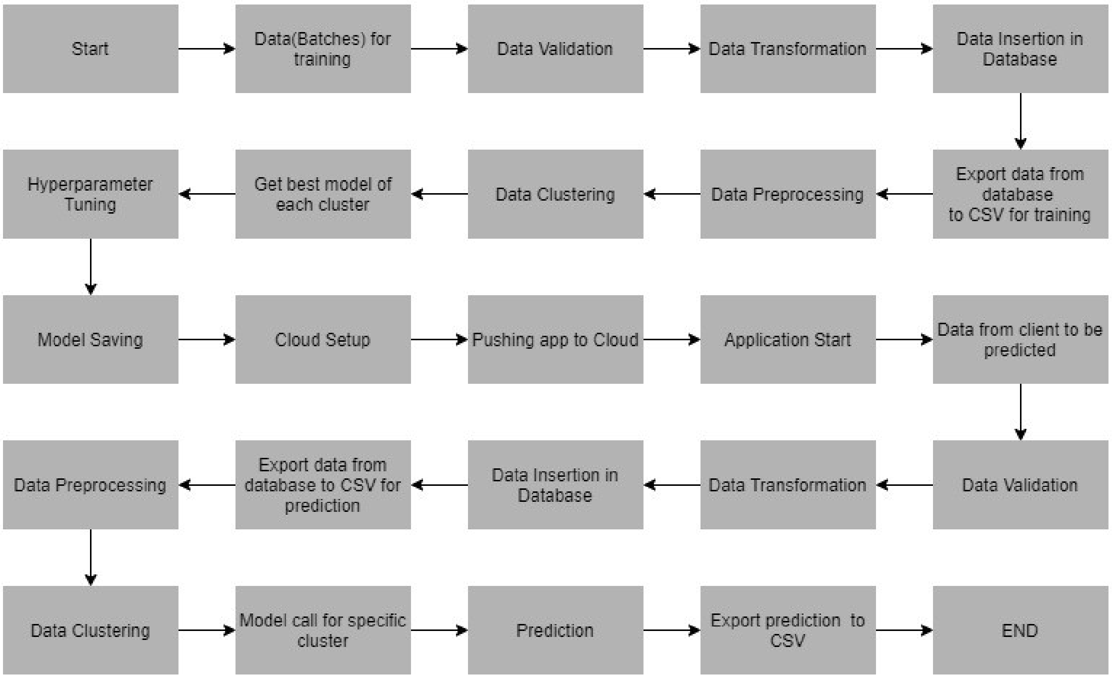
e). Use RandomOverSampler to manage the unbalanced dataset after imputing.

3.3.3 Clustering

The preprocessed data is clustered using the KMeans technique. The ideal number of clusters is determined by displaying the elbow plot, and the "KneeLocator" function is used to dynamically pick the number of clusters. Clustering is based on the use of several methods. To train data over several clusters. The K Means model is trained on preprocessed data and then stored for future prediction usage.

3.3.4 Model Selection

We determine the optimal model for each cluster once the clusters have been built. "Random Forest" and "KNN" are the two algorithms we're employing. Both methods are passed with the optimal parameters produced by GridSearch for each cluster. The AUC scores for both models are calculated, and the model with the highest score is chosen. Similarly, the model for each cluster is chosen. For prediction, all of the models for each cluster are kept.


Start
Data(Batches) for training
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for training
Data Preprocessing
Data Clustering
Get best model of each cluster
Hyperparameter Tuning
Model Saving
Cloud Setup
Pushing app to Cloud
Application Start
Data from client to be predicted
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for prediction
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV
END

3.4 Prediction Data Description

The data will be sent in batches by the client in several sets of files to a certain place. Wafer names and 590 columns of distinct sensor readings for each wafer will be included in the data.

ii) In addition to prediction files, we need a "schema" file from the client that provides all important information about the training files, such as file names, length of date value in FileName, length of time value in FileName, number of columns, column names, and datatype.

3.5 Prediction

3.5.1 Data Export from Db

The data in the stored database is exported as a CSV file to be used for prediction.

3.5.2 Data Preprocessing

a) Drop columns are useless for model training. These columns were chosen during the EDA.

b) Replace incorrect values with numpy "nan" so that the imputer may be used on them.

c) Write down the category values.

d) Inspect the columns for null values. If the null values are present, use the KNN imputer to compute them.

3.5.3 Clustering

KMeans model created during training is loaded , and clusters for the preprocessed prediction data is predicted .

3.5.4 Prediction

Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.

3.5.5 Once the prediction is made for all the clusters, the predictions along with the original names before label encoder are saved in a CSV file at a given location and the location is returned to the client.

3.3.3 Clustering

KMeans algorithm is used to create clusters in the preprocessed data. The optimum number of clusters is selected by plotting the elbow plot, and for the dynamic selection of the number of clusters, we are using the "KneeLocator" function. The idea behind clustering is to implement different algorithms. To train data in different clusters. The K Means model is trained over preprocessed data and the model is saved for further use in prediction.

3.3.4 Model Selection

After clusters are created, we find the best model for each cluster. We are using two algorithms, "Random Forest" and "KNN". For each cluster, both the algorithms are passed with the best parameters derived from GridSearch. We calculate the AUC scores for both models and select the model with the best score. Similarly, the model is selected for each cluster. All the models for every cluster are saved for use in prediction.

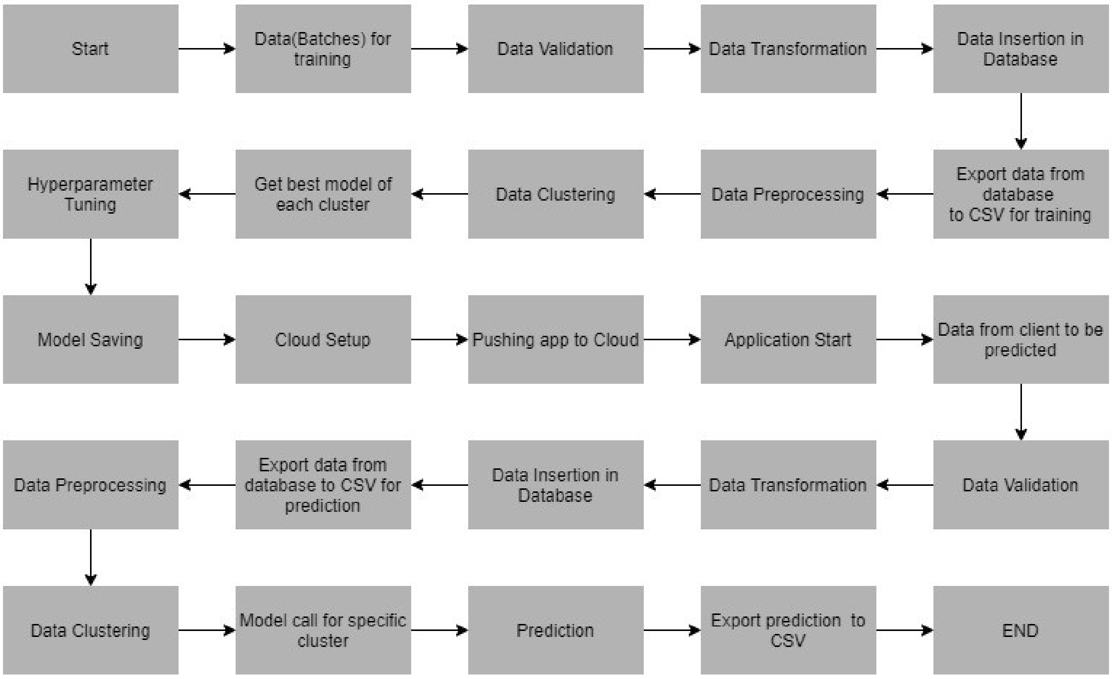

Start
Data(Batches) for training
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for training
Data Preprocessing
Data Clustering
Get best model of each cluster
Hyperparameter Tuning
Model Saving
Cloud Setup
Pushing app to Cloud
Application Start
Data from client to be predicted
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for prediction
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV
END

Figure 11 : DFD (Data Flow Diagram)

3.3.4.1 Prediction Data Description

i). Client will send the data in multiple sets of files in batches at a given location. Data will contain Wafer names and 590 columns of different sensor values for each wafer.

ii). Apart from prediction files, we also require a "schema" file from client which contains all the relevant information about the training files such as: Name of the files, Length of Date value in FileName, Length of Time value in FileName, Number of Columns, Name of the Columns and their datatype.

3.3.4.1 Model Prediction

3.3.4.2 Data Export from Db

The data in the stored database is exported as a CSV file to be used for prediction.

3.3.4.3 Data Preprocessing

a). Drop columns are not useful for training the model. Such columns were selected while doing the EDA.

b). Replace the invalid values with numpy "nan" so we can use the imputer on such values.

c). Encode the categorical values.

d). Check for null values in the columns. If present, impute the null values using the KNN imputer.

3.3.4.4 Clustering

KMeans model created during training is loaded, and clusters for the preprocessed prediction data is predicted.

3.3.4.5 Prediction

Based on the cluster number, the respective model is loaded and is used to predict the data for that cluster.

Once the prediction is made for all the clusters, the predictions along with the original names before label encoder are saved in a CSV file at a given location and the location is returned to the client.

3.4 Algorithms Used

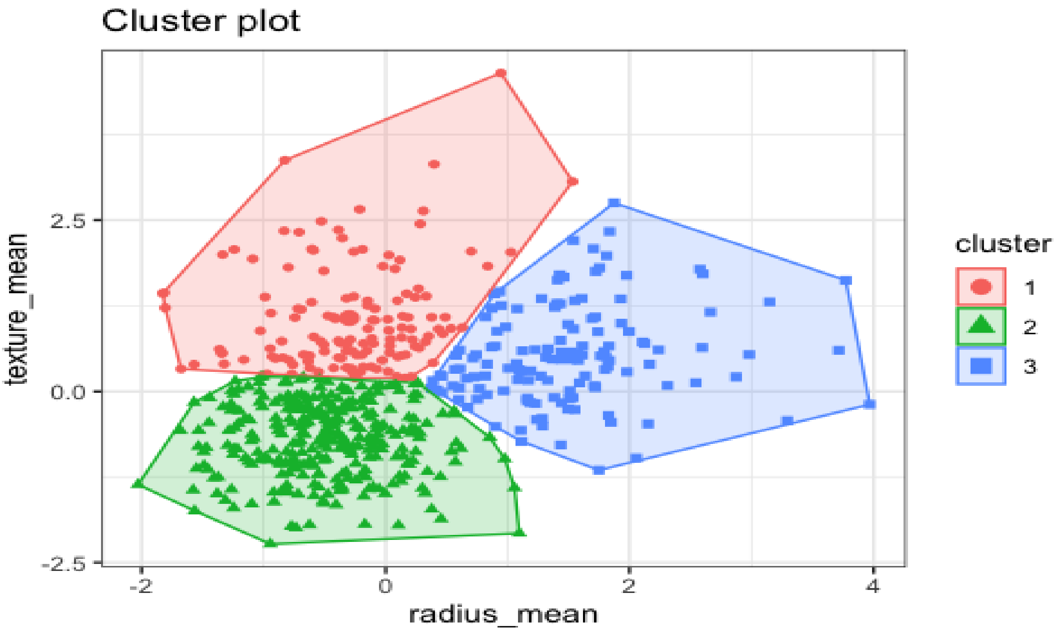
The main goal is to compare decision tree algorithms in order to find the best classification strategy for thyroid illness diagnosis. Experiments are carried out in order to compare the various types of decision tree algorithms described in the preceding section. The steps utilised in the study are listed in the subsections below.

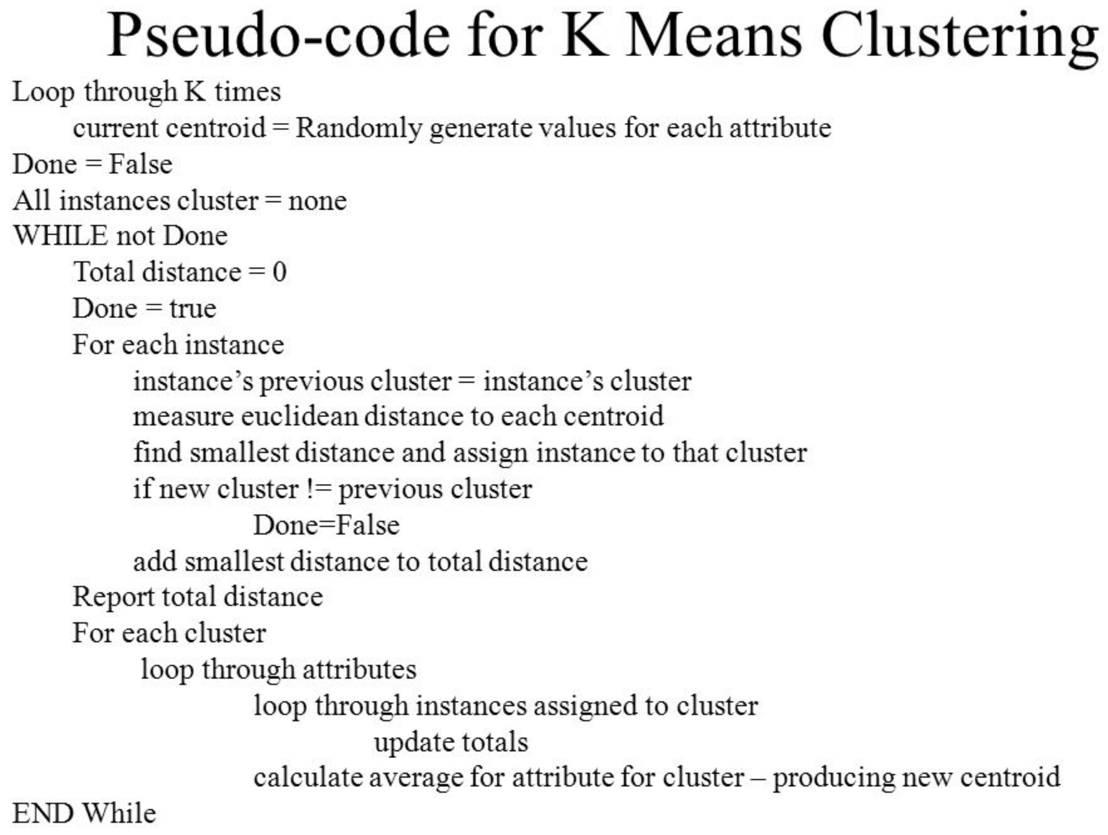
3.4.1 Training and testing process

Decision Tree, CART (Classification and Regression Trees), Random Forest, and K-nearest-neighbour were used for the experiment. The WEKA framework was used in order to implement these algorithms. The computational experiments have been performed. In the training and testing process, 10-fold cross validation was performed on the dataset.

3.4.2 K-Means clustering approach

K-means clustering is a method of vector quantization, first from signal processing, that is popular in the collection analysis in data mining. K-mean integration objectives to partition n observations into k groups each the view is of a cluster with a close description, acting as a collective prototype.





Pseudo-code for K Means Clustering

```
Loop through K times
    current centroid = Randomly generate values for each attribute
Done = False
All instances cluster = none
WHILE not Done
    Total distance = 0
    Done = true
    For each instance
        instance’s previous cluster = instance’s cluster
        measure euclidean distance to each centroid
        find smallest distance and assign instance to that cluster
        if new cluster != previous cluster
                Done=False
        add smallest distance to total distance
    Report total distance
    For each cluster
         loop through attributes
                loop through instances assigned to cluster
                        update totals
                calculate average for attribute for cluster – producing new centroid
END While
```

3.4.3 KNN (K Nearest Neighbour)

K-nearest-neighbour (KNN) Algorithm is a method for initiating laziness in the learning process. In other words, it makes no suppositions about the circulation of the information on which it is based. Regularly, the model is assessed utilising the dataset. It is valuable while working with genuine world datasets. Moreover, there are no preparation information focuses required for model age.

All preparing information is integrated into the testing system. While this speeds up arranging, it defers testing and requires a lot of time and memory. While building the model, the quantity of neighbours (K) should be indicated in KNN. For this situation, K goes about as a controlling variable for the forecast model. At the point when the quantity of classes is even, K is generally an odd number. It doesn't quickly gain from the preparation set; rather, it keeps up with the dataset and involves it for characterization. Figure portrays the functioning instrument of the KNN method.

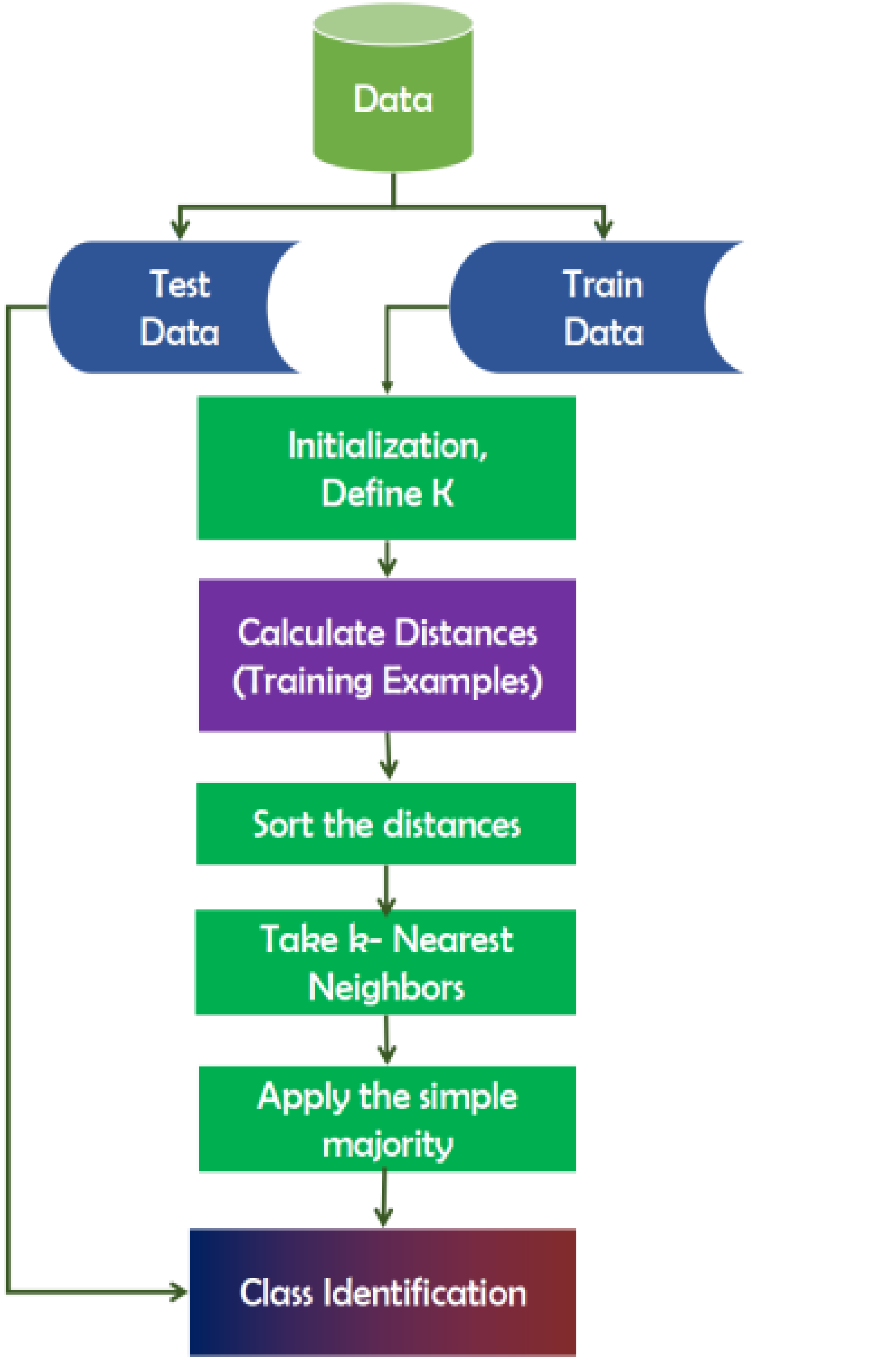

Data
Test Data
Train Data
Initialization, Define K
Calculate Distances (Training Examples)
Sort the distances
Take k- Nearest Neighbors
Apply the simple majority
Class Identification

Figure 12 Flowchart for KNN Algorithm

3.4.4 Decision Tree Algorithm

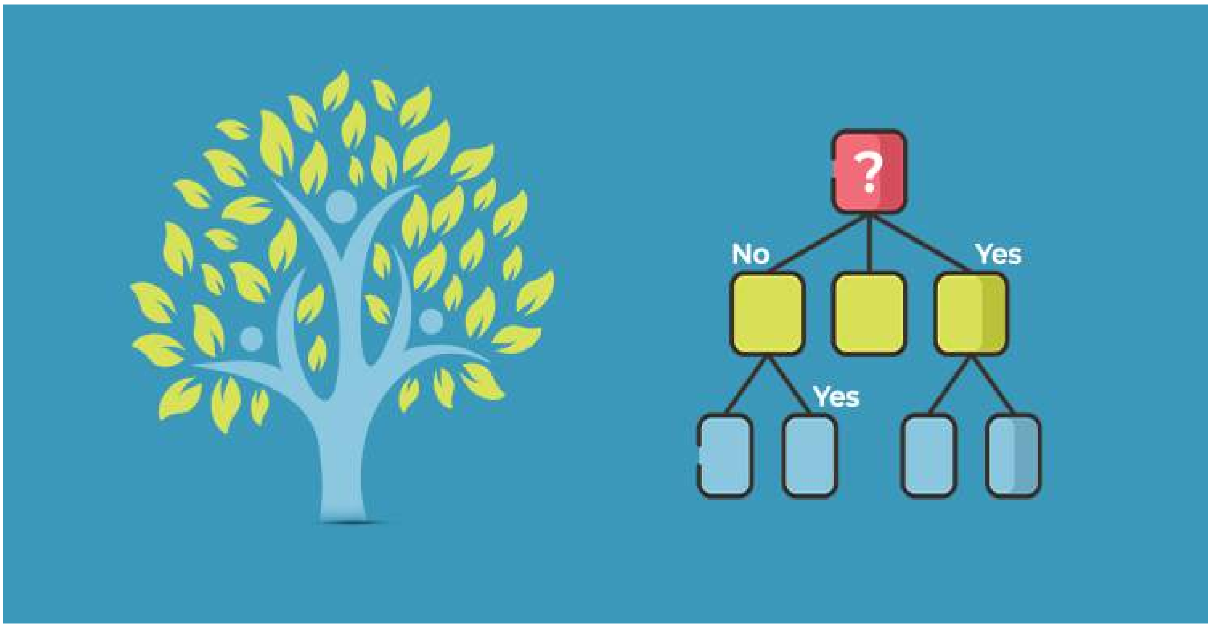


Figure 13 : Decision tree

The decision tree learning algorithm recursively learns the tree as follows:

1. Assign all training instances to the root of the tree. Set current node to root node.
2. For each attribute
    1. Partition all data instances at the node by the value of the attribute.
    2. Compute the information gain ratio from the partitioning.
3. Identify features that result in the greatest information gain ratio. Set this feature to be the splitting criterion at the current node.
    1. If the best information gain ratio is 0, tag the current node as a leaf and return.
4. Partition all instances according to attribute value of the best feature.
5. Denote each partition as a child node of the current node.
6. For each child node:
    1. If the child node is "pure" (has instances from only one class) , tag it as a leaf and return.
    2. If not, set the child node as the current node and recurse to step 2.

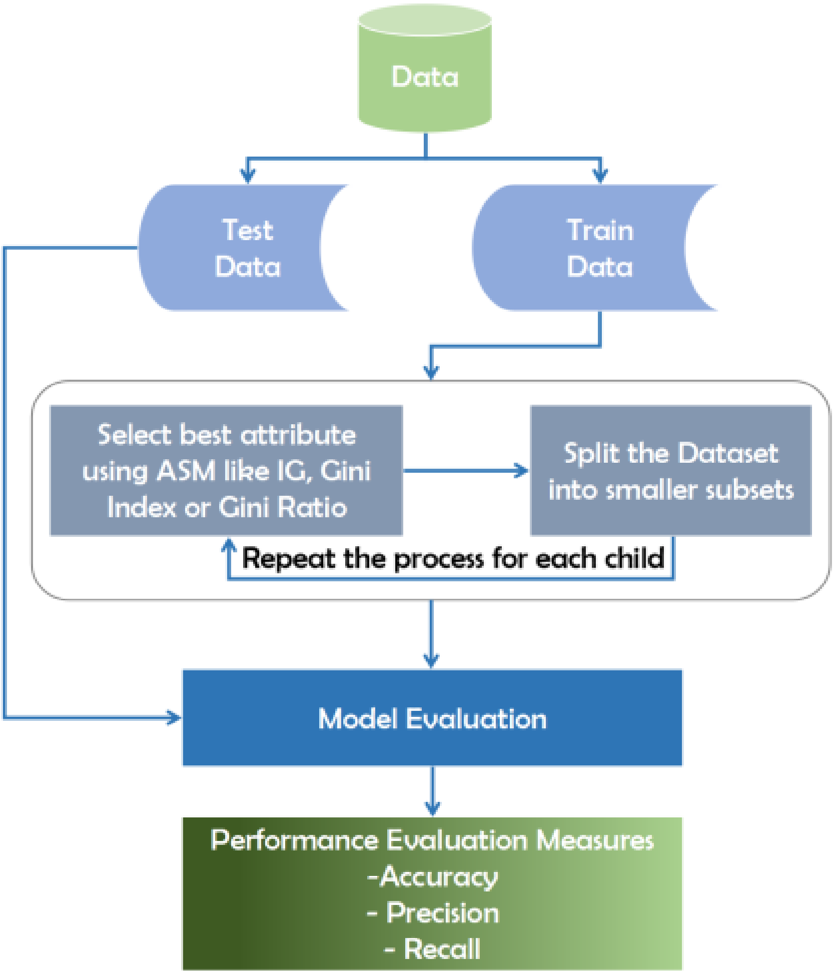

Data
Test Data
Train Data
Select best attribute using ASM like IG, Gini Index or Gini Ratio
Split the Dataset into smaller subsets
Repeat the process for each child
Model Evaluation
Performance Evaluation Measures
-Accuracy
- Precision
- Recall

Figure 14 : Flowchart for Decision Tree

3.4.5 Random Forest Algorithm

Random Forest pseudocode:

```
    1. Randomly select "k" features from total "m" features.
        1. Where k << m
2. Among the "k" features, calculate the node "d" using the best split point.
3. Split the node into daughter nodes using the best split.
4. Repeat 1 to 3 steps until the "l" number of nodes has been reached.
5. Build forest by repeating steps 1 to 4 for "n" number times to create "n" number of
   trees.
```

The beginning of the random forest algorithm starts with randomly selecting "k" features out of total "m" features. In the image, you can observe that we are randomly taking features and observations. In the next stage, we are using the randomly selected "k" features to find the root node by using the best split approach.

In the next stage, We will be calculating the daughter nodes using the same best split approach. Will the first 3 stages until we form the tree with a root node and have the target as the leaf node. Finally, we repeat 1 to 4 stages to create "n" randomly created trees. These randomly created trees form the random forest.

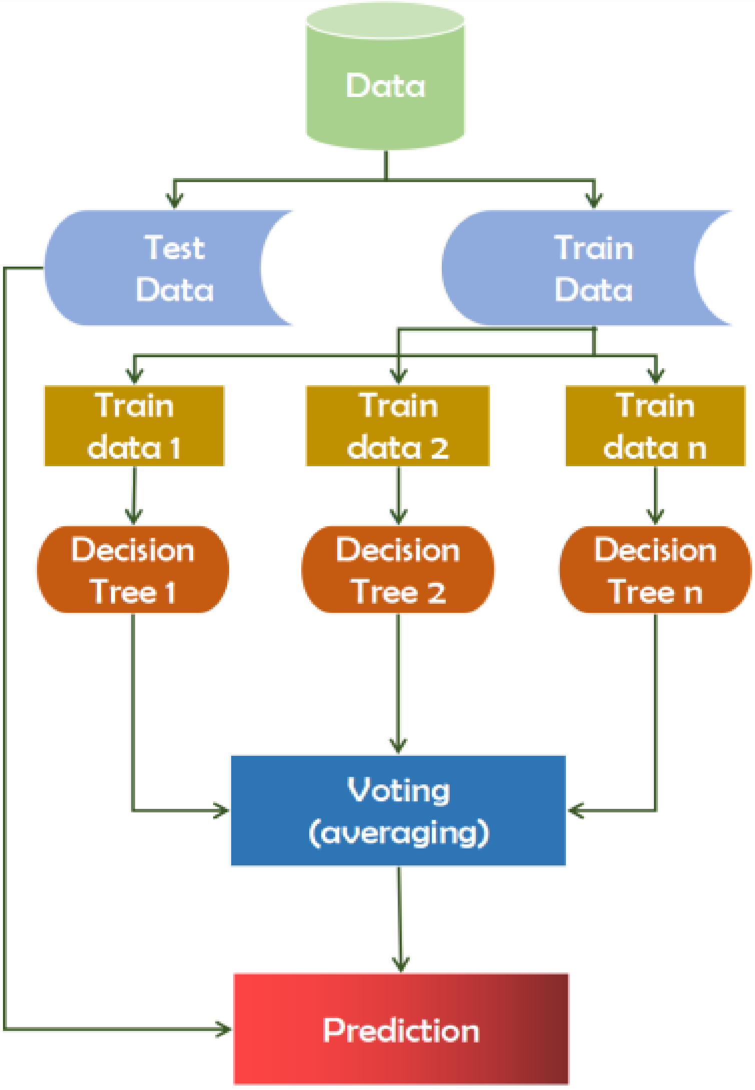
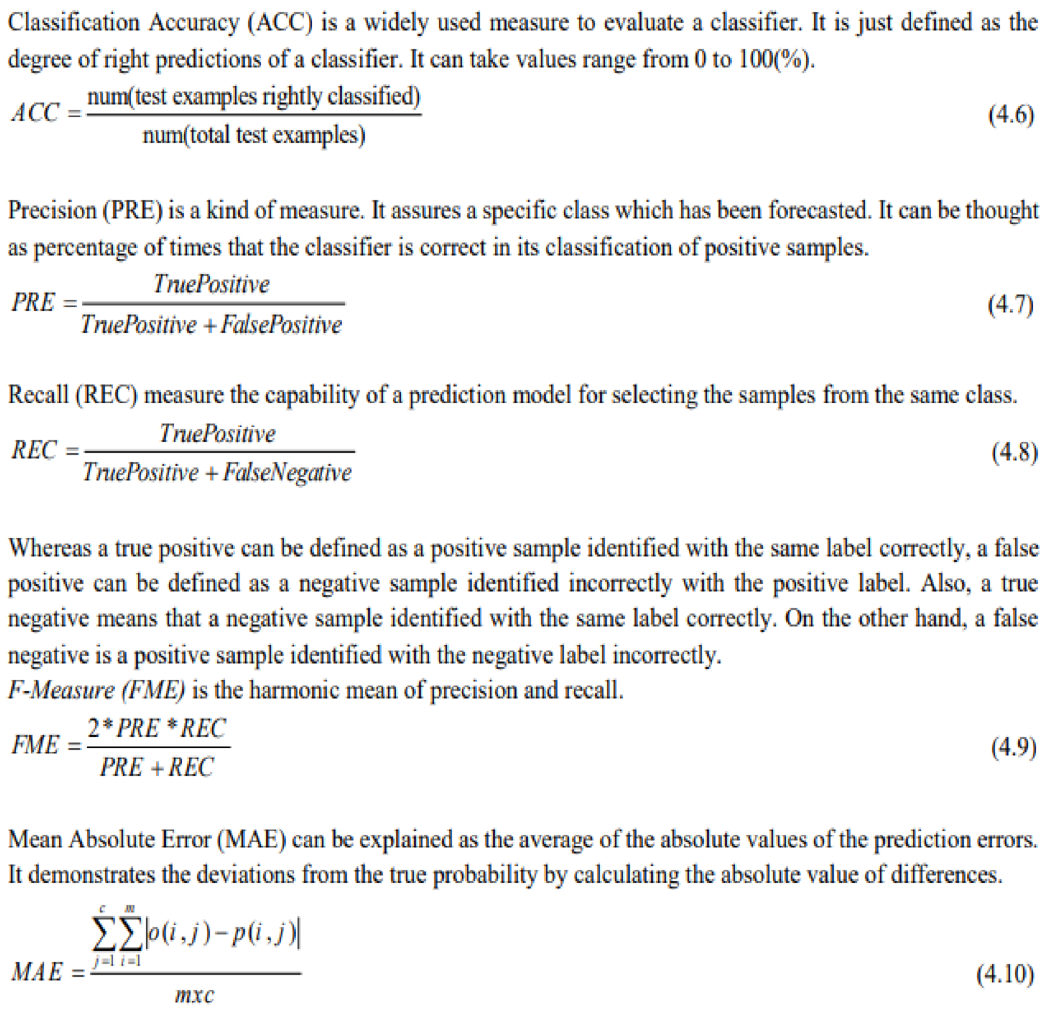


Figure 15 : Flowchart for Random Forest

3.5 Model Development

3.5.1 Mathematical / Statistical

3.5.1.1 Training and testing Process



Classification Accuracy (ACC) is a widely used measure to evaluate a classifier. It is just defined as the degree of right predictions of a classifier. It can take values range from 0 to 100(%).

$$ACC = \frac{\text{num(test examples rightly classified)}}{\text{num(total test examples)}} \quad (4.6)$$

Precision (PRE) is a kind of measure. It assures a specific class which has been forecasted. It can be thought as percentage of times that the classifier is correct in its classification of positive samples.

$$PRE = \frac{TruePositive}{TruePositive + FalsePositive} \quad (4.7)$$

Recall (REC) measure the capability of a prediction model for selecting the samples from the same class.

$$REC = \frac{TruePositive}{TruePositive + FalseNegative} \quad (4.8)$$

Whereas a true positive can be defined as a positive sample identified with the same label correctly, a false positive can be defined as a negative sample identified incorrectly with the positive label. Also, a true negative means that a negative sample identified with the same label correctly. On the other hand, a false negative is a positive sample identified with the negative label incorrectly.
F-Measure (FME) is the harmonic mean of precision and recall.

$$FME = \frac{2 * PRE * REC}{PRE + REC} \quad (4.9)$$

Mean Absolute Error (MAE) can be explained as the average of the absolute values of the prediction errors. It demonstrates the deviations from the true probability by calculating the absolute value of differences.

$$MAE = \frac{\sum_{j=1}^{c}\sum_{i=1}^{m}\left|o(i,j) - p(i,j)\right|}{mxc} \quad (4.10)$$

3.5.1.2 Root Mean Square Deviation (RMSD)

RMSD is just basically the standard deviation of the predicted errors. Residue which are the measure of the regression where the data points are, however it also shows this widespread of the residuals in the data points and also finds out the the best fit in the data .It is also used in forecasting ,regression analysis to get the verified results of the experiments . Better the performance lower will be The RMSE value.

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

RMSD = root-mean-square deviation

i = variable i

N = number of non-missing data points

x_i = actual observations time series

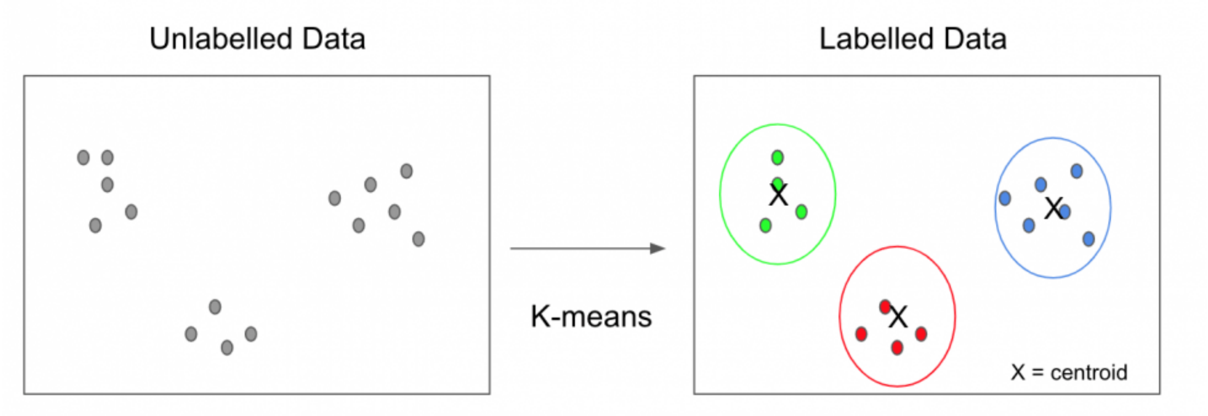
$\hat{x}_i$ = estimated time series

3.5.1.3 K-Means Clustering

K-implies bunching is a strategy for vector quantization, first from signal handling, that is famous in the assortment examination in information mining. K-mean coordination goals to segment n perceptions into k gatherings each the view is of a group with a nearby depiction, going about as an aggregate model.

Given the observation set (x1, x2, ..., xn), where each observation is a real d-dimensional vector, the addition of k means the marked division into k sets (k ≤ n) S = {S1, S2 ,⋯, Sk} to reduce the total number of squares within a set :

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}_j \in S_i} \left\| \mathbf{x}_j - \boldsymbol{\mu}_i \right\|^2$$



3.6 SDLC (Software Development Life Cycle)

The Software Development Life Cycle (SDLC), also known as the Systems Architecture, Information Systems, and Network Engineering Software Development Life Cycle (SDLC), is the process of creating or updating systems, as well as the concepts and techniques used to create them. All software development approaches are supported by the SDLC paradigm engineering. These approaches serve as the foundation for developing and maintaining a software development information system.

3.6.1 Existing System: Thyroid disease is a major cause of medical diagnosis and estimate, and it is also a major cause of death. A medical axiom is being challenged. Thyroid hormone releases are responsible for metabolic control. Thyroid hormones are released to manage hyperthyroidism and hypothyroidism, two common thyroid illnesses in the body's metabolic rate.

To make data rudimentary enough for analytics to illustrate the likelihood of patients having thyroid disease, data cleaning procedures were applied. Energy level, Weakness, Breathing are all disadvantages.

3.6.2 System Proposed: Machine learning is important in the prediction process, and material from UCI machine learning archives is utilised in paper research and model classifications for thyroid illness.

In order to solve complicated learning concerns like medical diagnostics and statistical tasks, a good knowledge base that can be centred and used as a hybrid paradigm must be kept.

We also offered other machine learning and thyroid diagnostic methodologies.

The estimated chance of a patient having thyroid illness was calculated using various machine learning classification algorithms, Logic regression, Decision Trees, Random Forest and K-NN.

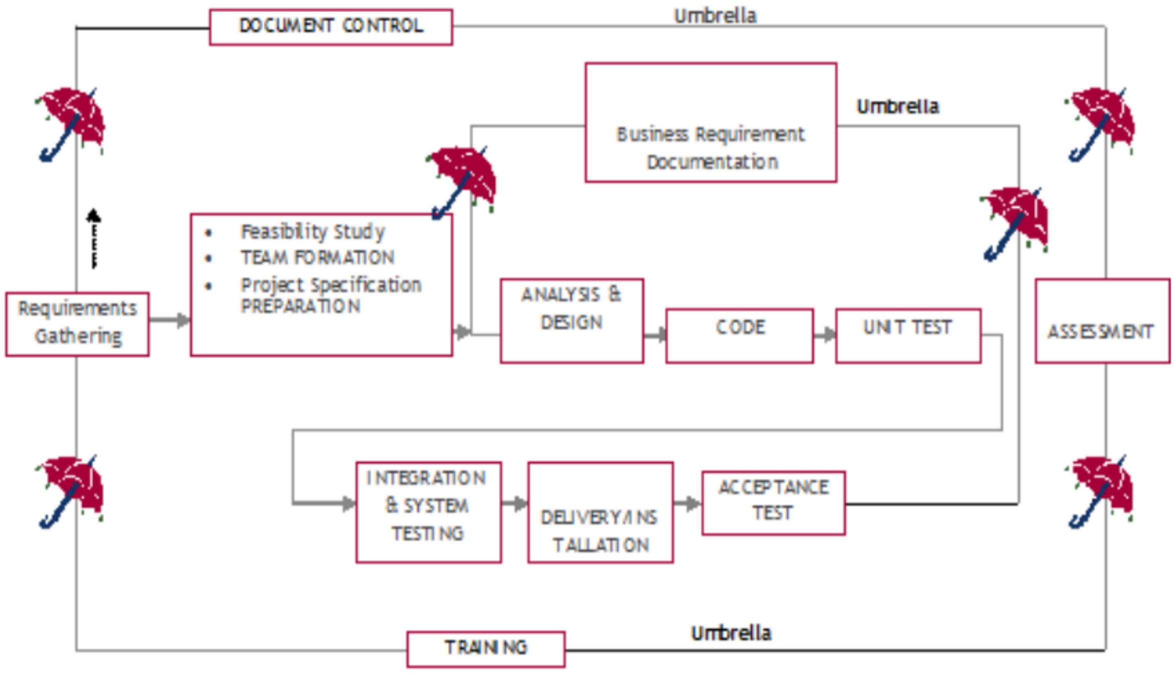


Fig Block Diagram of SDLC

CHAPTER 4

PERFORMANCE ANALYSIS

4.1 Data Flow Diagram

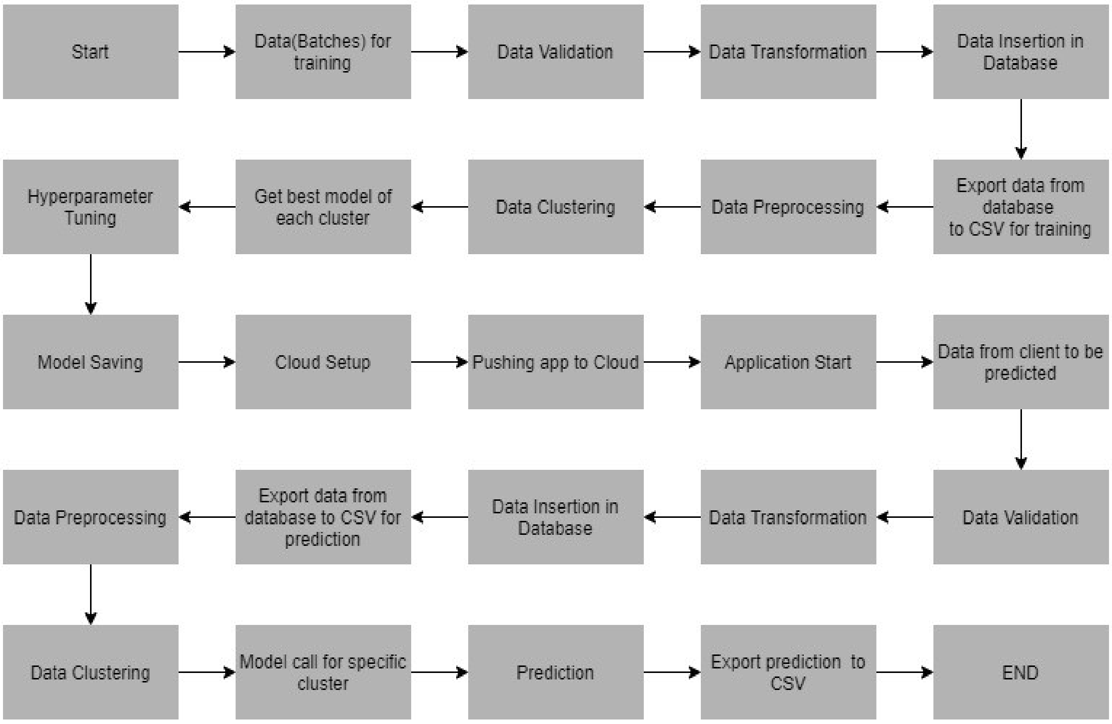
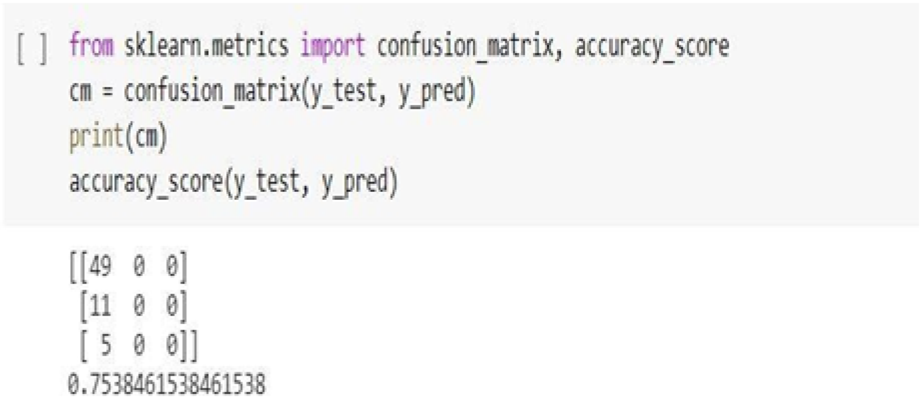

Start
Data(Batches) for training
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for training
Data Preprocessing
Data Clustering
Get best model of each cluster
Hyperparameter Tuning
Model Saving
Cloud Setup
Pushing app to Cloud
Application Start
Data from client to be predicted
Data Validation
Data Transformation
Data Insertion in Database
Export data from database to CSV for prediction
Data Preprocessing
Data Clustering
Model call for specific cluster
Prediction
Export prediction to CSV
END

Figure 17 : DFD (Data Flow Diagram)

4.2 Computational Analysis of System Developed

These are the following accuracies obtained while performing operations on my project using Logistic Regression, Decision Tree, and KNN Algorithm.

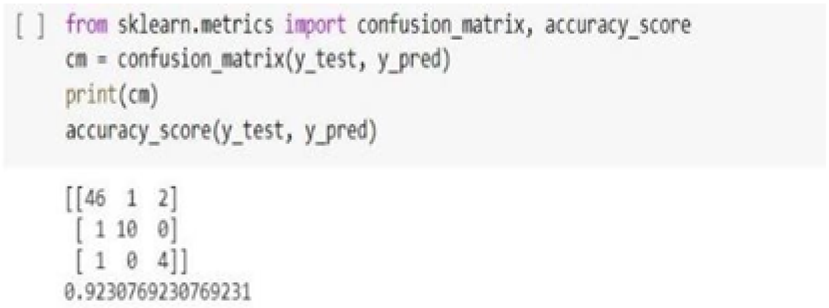
4.2.1 Using Logistic Regression



```
[ ] from sklearn.metrics import confusion_matrix, accuracy_score
    cm = confusion_matrix(y_test, y_pred)
    print(cm)
    accuracy_score(y_test, y_pred)
```

```
[[49  0  0]
 [11  0  0]
 [ 5  0  0]]
0.7538461538461538
```

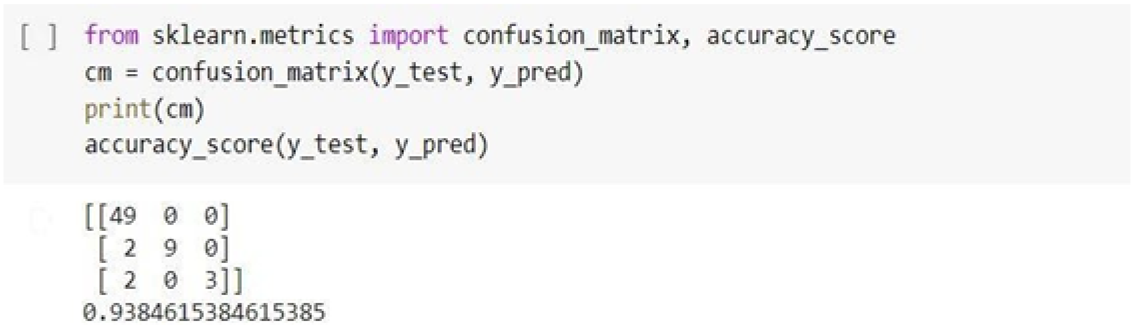
4.2.2 Using Decision Tree



```
[ ] from sklearn.metrics import confusion_matrix, accuracy_score
    cm = confusion_matrix(y_test, y_pred)
    print(cm)
    accuracy_score(y_test, y_pred)
```

```
[[46  1  2]
 [ 1 10  0]
 [ 1  0  4]]
0.9230769230769231
```

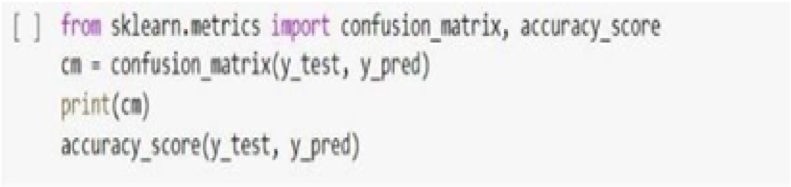
4.2.3 Using KNN Algorithm



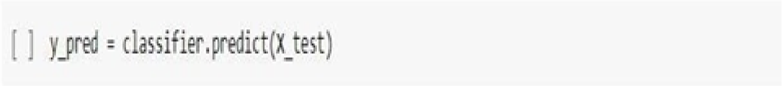
```
[ ] from sklearn.metrics import confusion_matrix, accuracy_score
    cm = confusion_matrix(y_test, y_pred)
    print(cm)
    accuracy_score(y_test, y_pred)
```

```
[[49  0  0]
 [ 2  9  0]
 [ 2  0  3]]
0.9384615384615385
```

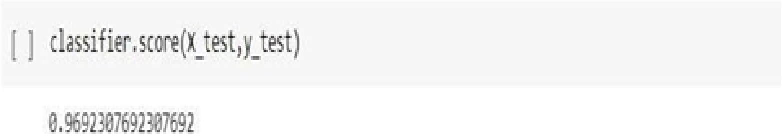
4.2.4 Using Random Forest Algorithm



```
[ ] from sklearn.metrics import confusion_matrix, accuracy_score
    cm = confusion_matrix(y_test, y_pred)
    print(cm)
    accuracy_score(y_test, y_pred)
```



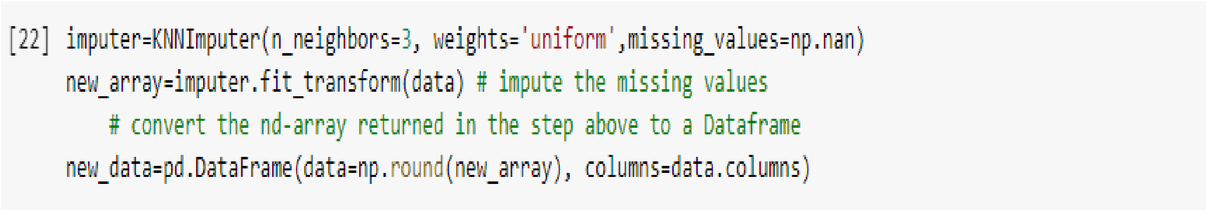
```
[ ] y_pred = classifier.predict(X_test)
```



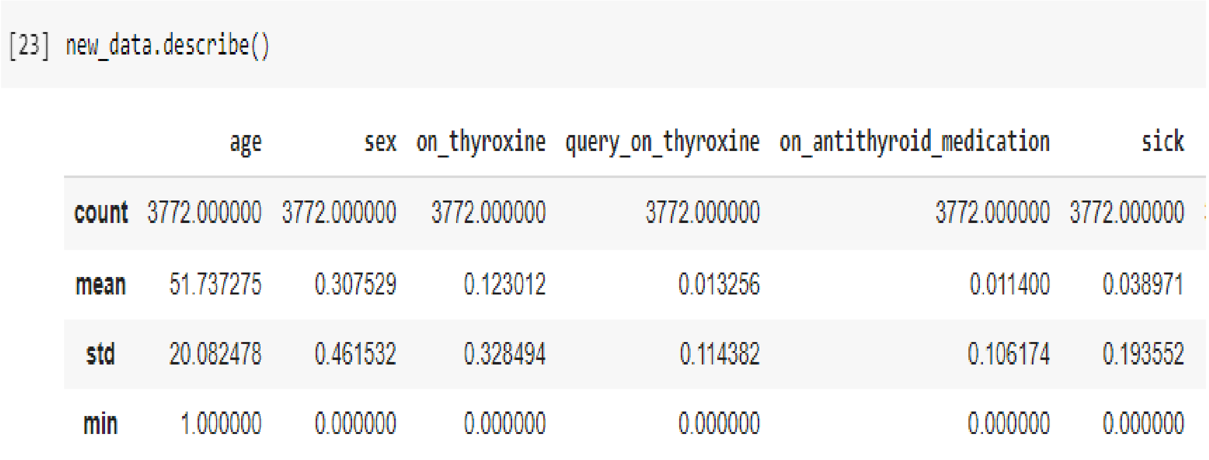
```
[ ] classifier.score(X_test,y_test)

    0.9692307692307692
```

4.3 Experimental Results / Outputs at various Stages

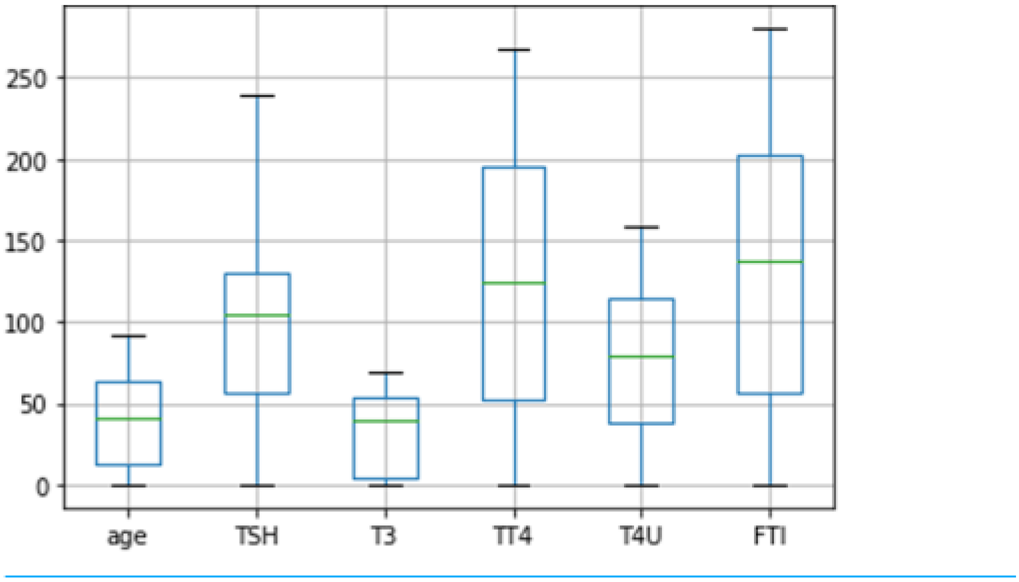
Here, we have applied KNN Imputer in order to impute the missing values present in the Dataset, in this the KNN Imputer is what it basically does is it takes the arithmetic average of the above present 3 neighbours from the missing value.



```
[22] imputer=KNNImputer(n_neighbors=3, weights='uniform',missing_values=np.nan)
     new_array=imputer.fit_transform(data) # impute the missing values
        # convert the nd-array returned in the step above to a Dataframe
     new_data=pd.DataFrame(data=np.round(new_array), columns=data.columns)
```

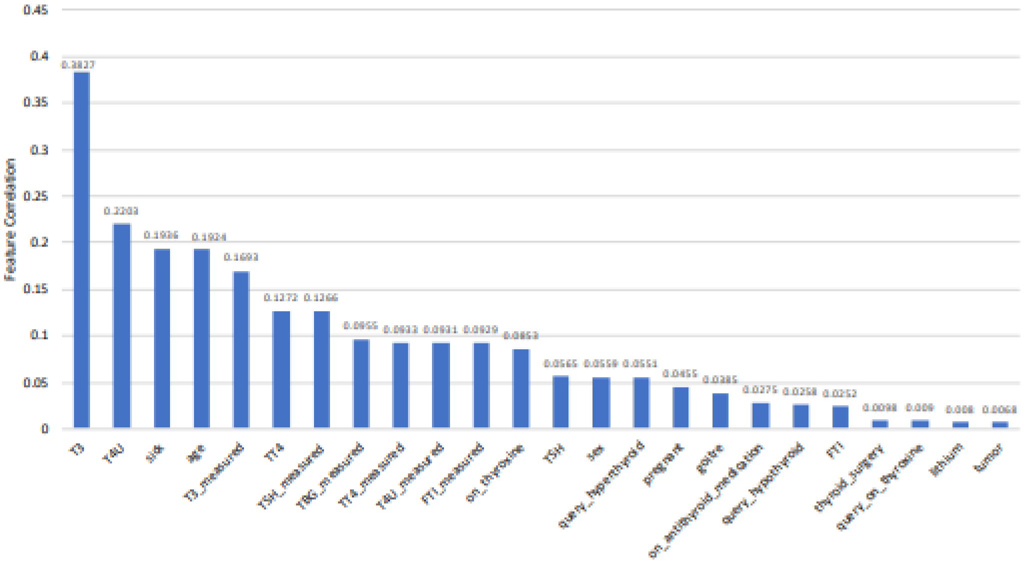


```
[23] new_data.describe()

               age          sex  on_thyroxine  query_on_thyroxine  on_antithyroid_medication          sick
count  3772.000000  3772.000000   3772.000000         3772.000000                3772.000000   3772.000000
mean     51.737275     0.307529      0.123012            0.013256                   0.011400      0.038971
std      20.082478     0.461532      0.328494            0.114382                   0.106174      0.193552
min       1.000000     0.000000      0.000000            0.000000                   0.000000      0.000000
```



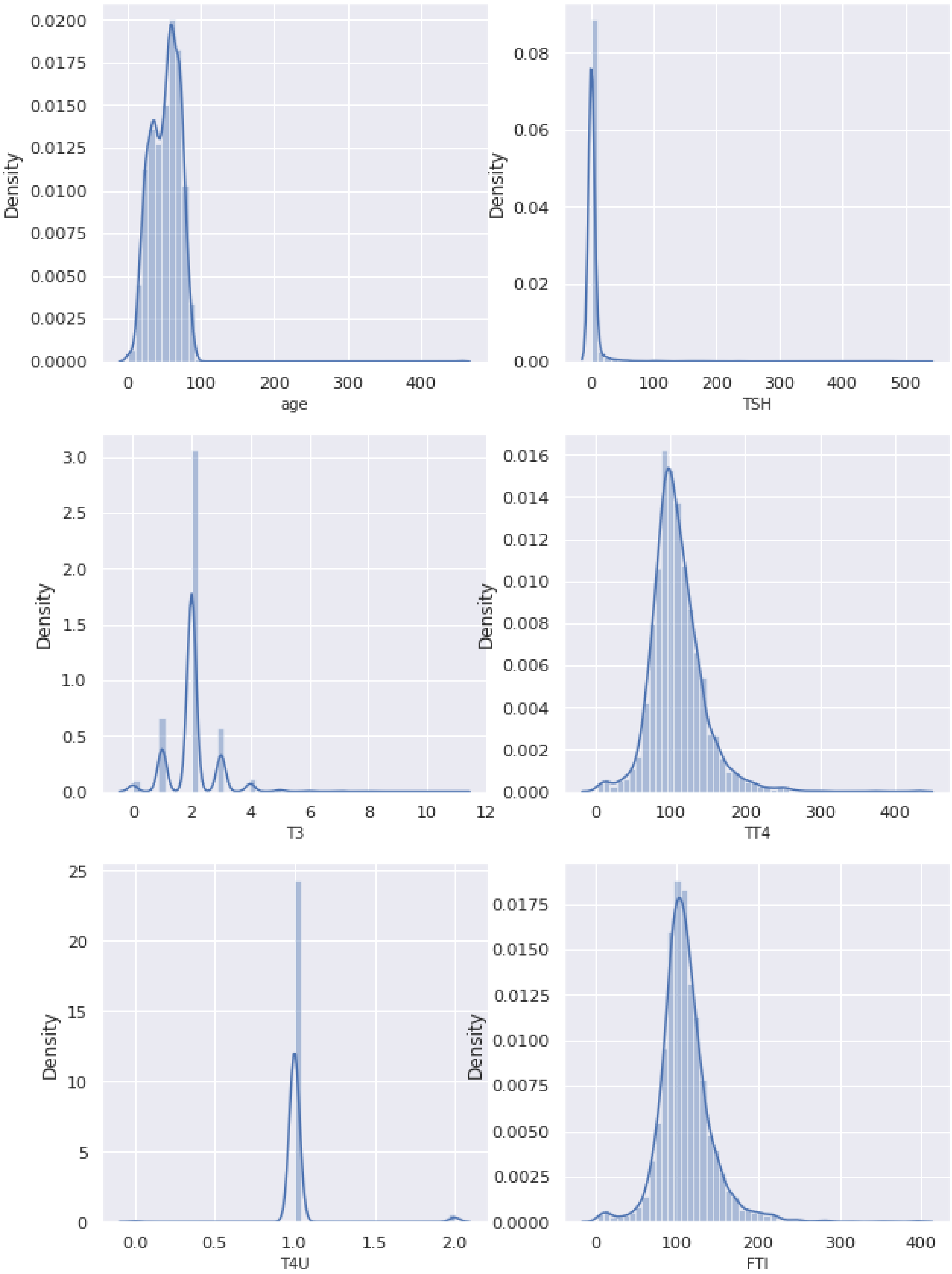
Graph 2 : Box Plot Diagram For some Dataset Features

The Correlation scores of every features of the Dataset **:-**



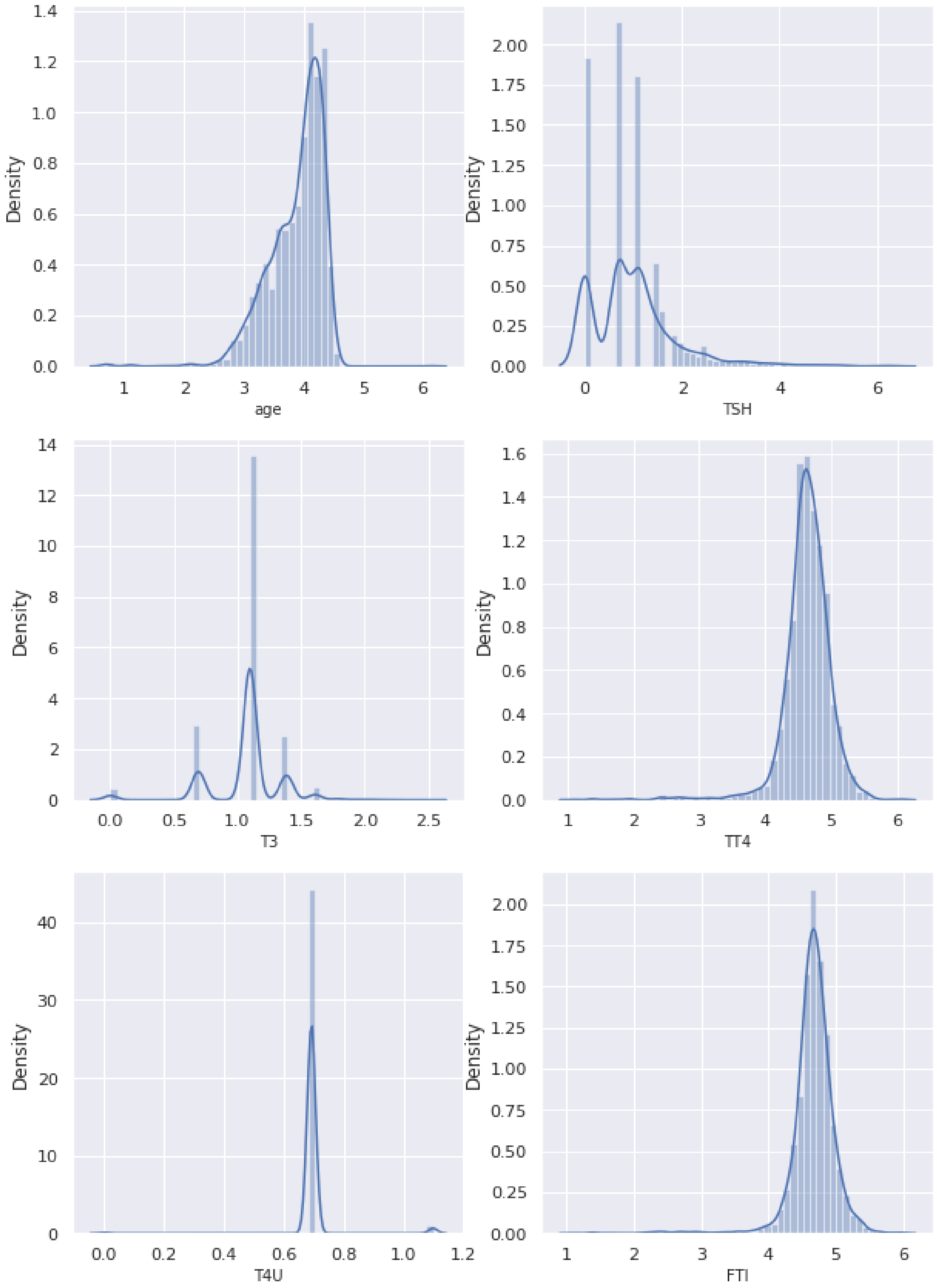
Graph 3 : Correlation of features of Dataset

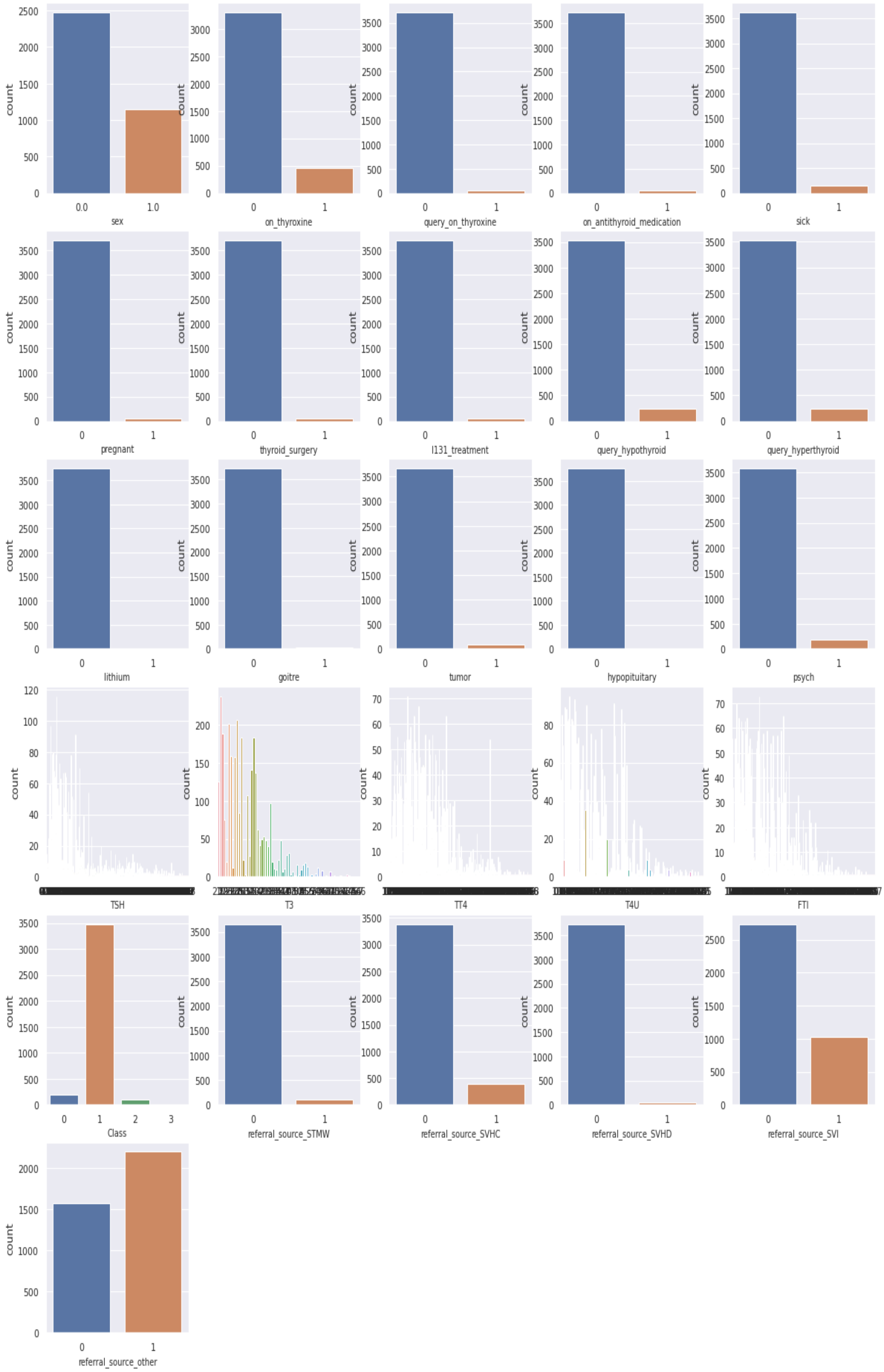
Now, let us check if the distribution of our data is continuous or not.



Graph 4 : Distribution of features of Dataset

The graphs for age, TSH and T3 look heavily skewed towards the left. Let's do some transformations to the data and see if it improves the plot.Before doing log transformation , let's add 1 to each value in the column to handle exceptions when we try to find a log of '0'.



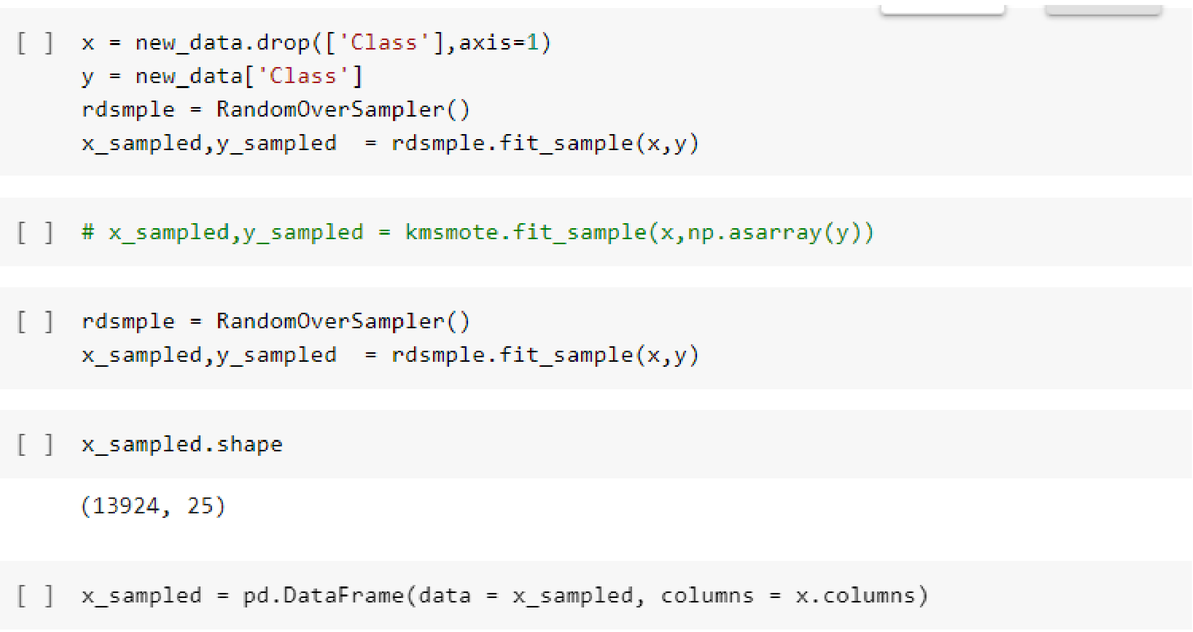

count
2500
2000
1500
1000
500
0
0.0
1.0
sex
3000
on_thyroxine
0
1
query_on_thyroxine
on_antithyroid_medication
sick
3500
pregnant
thyroid_surgery
I131_treatment
query_hypothyroid
query_hyperthyroid
lithium
goitre
tumor
hypopituitary
psych
120
100
80
60
40
20
TSH
200
150
50
T3
70
50
30
10
TT4
T4U
FTI
Class
2
3
referral_source_STMW
referral_source_SVHC
referral_source_SVHD
referral_source_SVI
referral_source_other

The above graph describes how balanced our dataset in terms of given target classes is, which means how the data is distributed for every column for every individual id present in the dataset.

We can clearly see that the dataset is highly imbalanced.

So, In order to balance the dataset, we will use a python library known as imbalanced-learn to deal with imbalanced data. Imbalanced learning has an algorithm called RandomOverSampler.

After performing all the required EDA operations , now our dataset is balanced and it looks like the following.



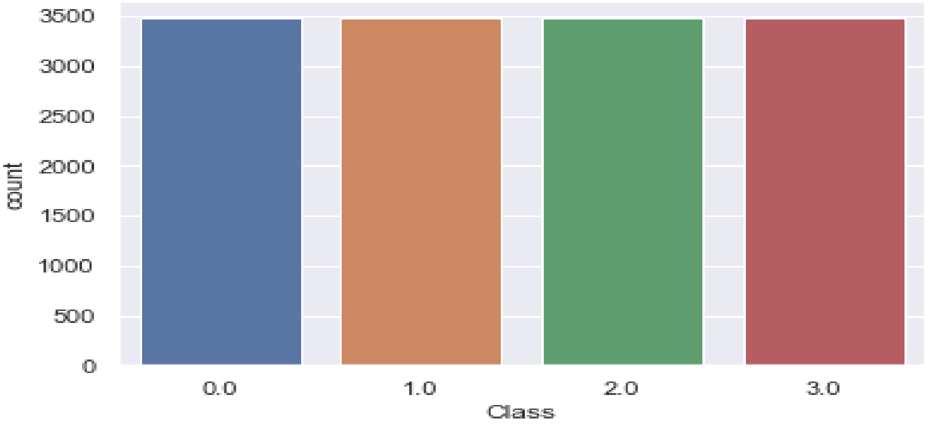
```
[ ]  x = new_data.drop(['Class'],axis=1)
     y = new_data['Class']
     rdsmple = RandomOverSampler()
     x_sampled,y_sampled  = rdsmple.fit_sample(x,y)
```

```
[ ]  # x_sampled,y_sampled = kmsmote.fit_sample(x,np.asarray(y))
```

```
[ ]  rdsmple = RandomOverSampler()
     x_sampled,y_sampled  = rdsmple.fit_sample(x,y)
```

```
[ ]  x_sampled.shape

     (13924, 25)
```

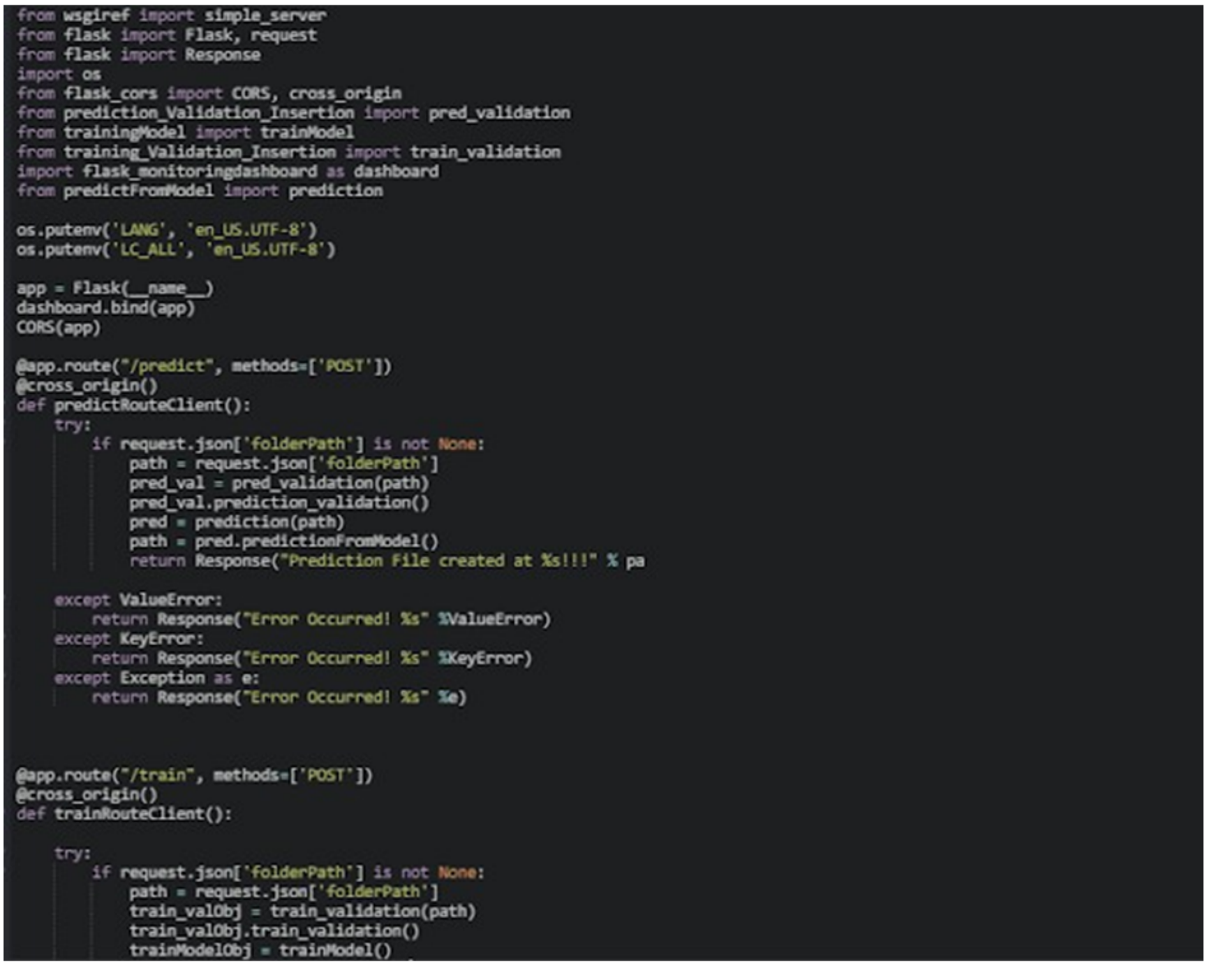
```
[ ]  x_sampled = pd.DataFrame(data = x_sampled, columns = x.columns)
```



Graph 5 : Balanced features of Dataset

4.4 Snapshots of Code at various stages of Project

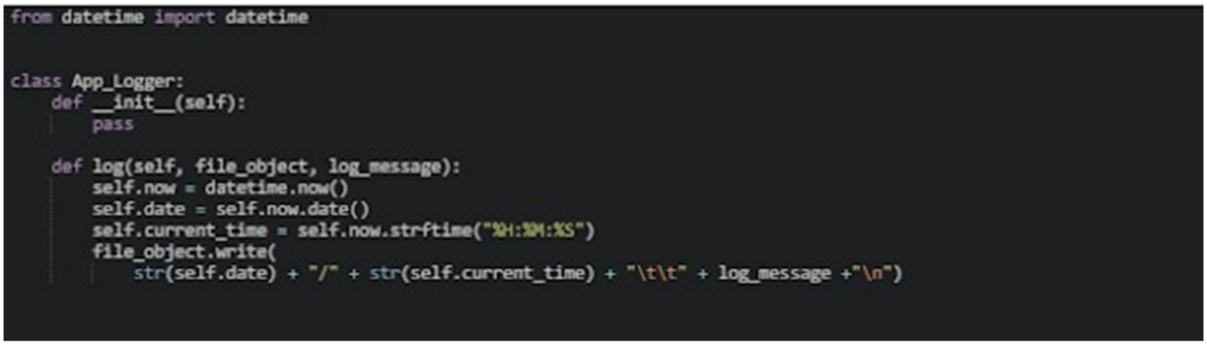
4.4.1 Main.py



```
from wsgiref import simple_server
from flask import Flask, request
from flask import Response
import os
from flask_cors import CORS, cross_origin
from prediction_Validation_Insertion import pred_validation
from trainingModel import trainModel
from training_Validation_Insertion import train_validation
import flask_monitoringdashboard as dashboard
from predictFromModel import prediction

os.putenv('LANG', 'en_US.UTF-8')
os.putenv('LC_ALL', 'en_US.UTF-8')

app = Flask(__name__)
dashboard.bind(app)
CORS(app)

@app.route("/predict", methods=['POST'])
@cross_origin()
def predictRouteClient():
    try:
        if request.json['folderPath'] is not None:
            path = request.json['folderPath']
            pred_val = pred_validation(path)
            pred_val.prediction_validation()
            pred = prediction(path)
            path = pred.predictionFromModel()
            return Response("Prediction File created at %s!!!" % pa

    except ValueError:
        return Response("Error Occurred! %s" %ValueError)
    except KeyError:
        return Response("Error Occurred! %s" %KeyError)
    except Exception as e:
        return Response("Error Occurred! %s" %e)



@app.route("/train", methods=['POST'])
@cross_origin()
def trainRouteClient():

    try:
        if request.json['folderPath'] is not None:
            path = request.json['folderPath']
            train_valObj = train_validation(path)
            train_valObj.train_validation()
            trainModelObj = trainModel()
```

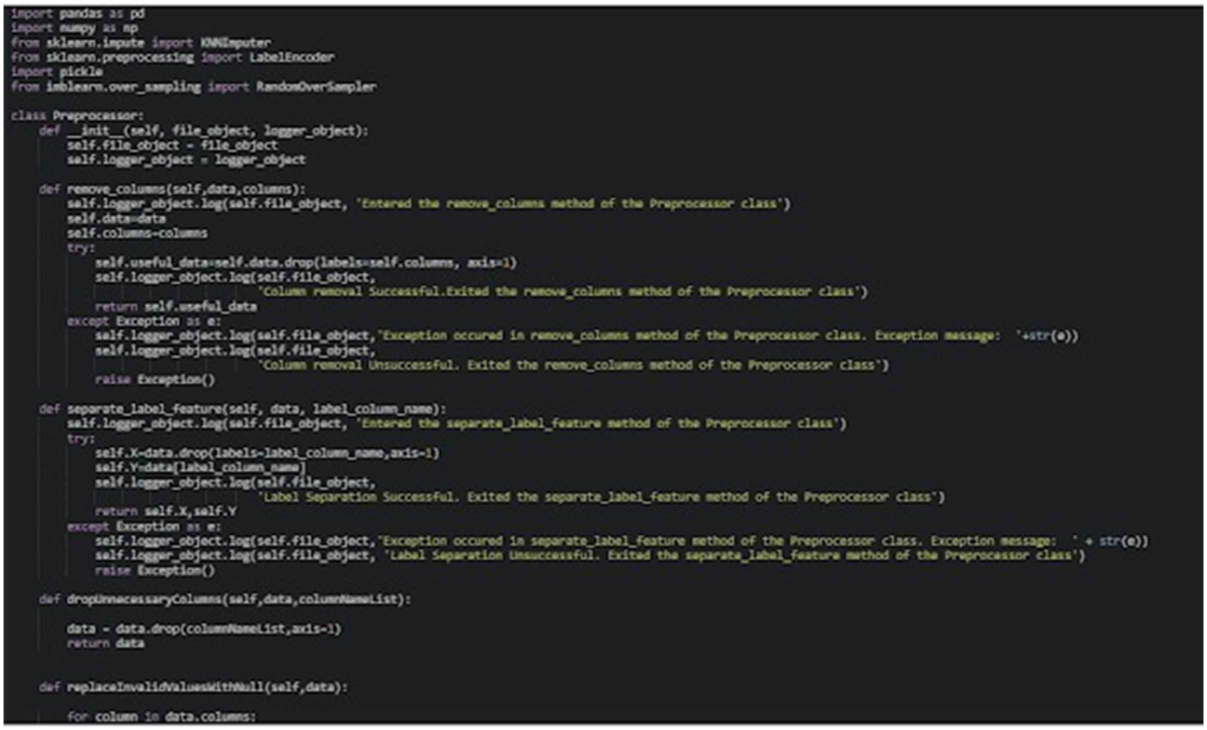
4.7.2 Application Logging

Logger.py



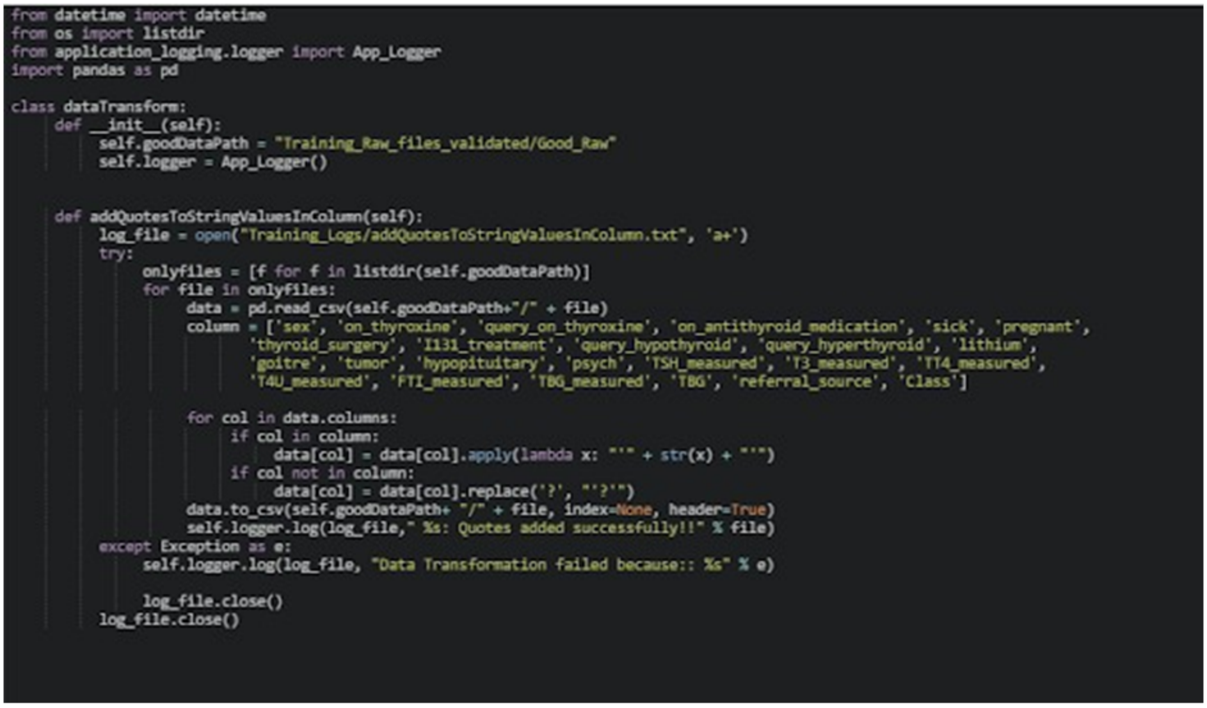
```
from datetime import datetime


class App_Logger:
    def __init__(self):
        pass

    def log(self, file_object, log_message):
        self.now = datetime.now()
        self.date = self.now.date()
        self.current_time = self.now.strftime("%H:%M:%S")
        file_object.write(
            str(self.date) + "/" + str(self.current_time) + "\t\t" + log_message +"\n")
```

4.7.3 Data Preprocessing



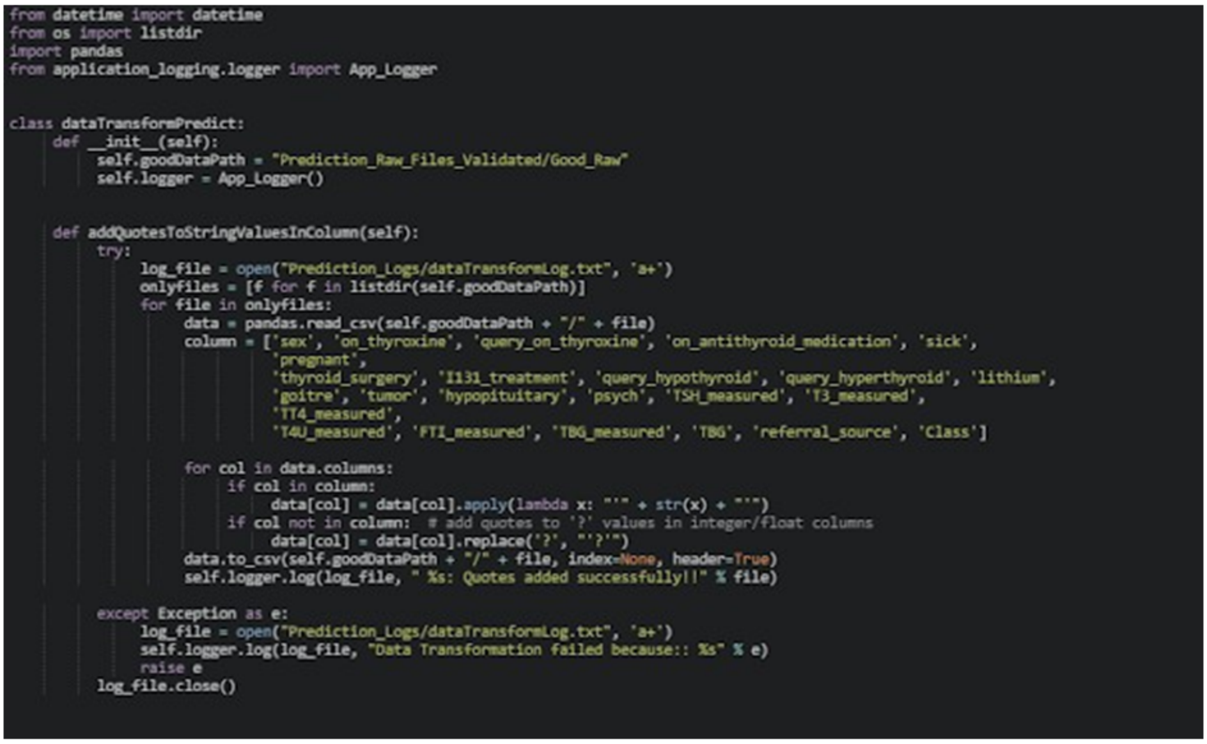
```python
import pandas as pd
import numpy as np
from sklearn.impute import KNNImputer
from sklearn.preprocessing import LabelEncoder
import pickle
from imblearn.over_sampling import RandomOverSampler

class Preprocessor:
    def __init__(self, file_object, logger_object):
        self.file_object = file_object
        self.logger_object = logger_object

    def remove_columns(self,data,columns):
        self.logger_object.log(self.file_object, 'Entered the remove_columns method of the Preprocessor class')
        self.data=data
        self.columns=columns
        try:
            self.useful_data=self.data.drop(labels=self.columns, axis=1)
            self.logger_object.log(self.file_object,
                                   'Column removal Successful.Exited the remove_columns method of the Preprocessor class')
            return self.useful_data
        except Exception as e:
            self.logger_object.log(self.file_object,'Exception occured in remove_columns method of the Preprocessor class. Exception message:  '+str(e))
            self.logger_object.log(self.file_object,
                                   'Column removal Unsuccessful. Exited the remove_columns method of the Preprocessor class')
            raise Exception()

    def separate_label_feature(self, data, label_column_name):
        self.logger_object.log(self.file_object, 'Entered the separate_label_feature method of the Preprocessor class')
        try:
            self.X=data.drop(labels=label_column_name,axis=1)
            self.Y=data[label_column_name]
            self.logger_object.log(self.file_object,
                                   'Label Separation Successful. Exited the separate_label_feature method of the Preprocessor class')
            return self.X,self.Y
        except Exception as e:
            self.logger_object.log(self.file_object,'Exception occured in separate_label_feature method of the Preprocessor class. Exception message:  ' + str(e))
            self.logger_object.log(self.file_object, 'Label Separation Unsuccessful. Exited the separate_label_feature method of the Preprocessor class')
            raise Exception()

    def dropUnnecessaryColumns(self,data,columnNameList):

        data = data.drop(columnNameList,axis=1)
        return data


    def replaceInvalidValuesWithNull(self,data):

        for column in data.columns:
```

4.7.4 Data Transformation Training



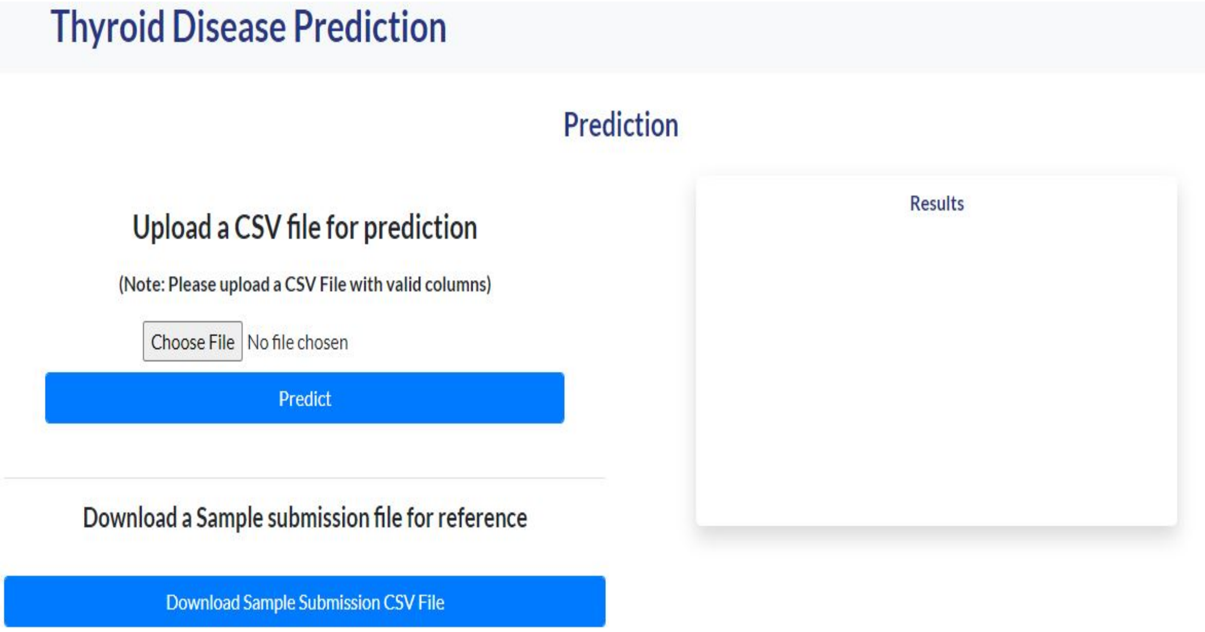
```python
from datetime import datetime
from os import listdir
from application_logging.logger import App_Logger
import pandas as pd

class dataTransform:
    def __init__(self):
        self.goodDataPath = "Training_Raw_files_validated/Good_Raw"
        self.logger = App_Logger()


    def addQuotesToStringValuesInColumn(self):
        log_file = open("Training_Logs/addQuotesToStringValuesInColumn.txt", 'a+')
        try:
            onlyfiles = [f for f in listdir(self.goodDataPath)]
            for file in onlyfiles:
                data = pd.read_csv(self.goodDataPath+"/" + file)
                column = ['sex', 'on_thyroxine', 'query_on_thyroxine', 'on_antithyroid_medication', 'sick', 'pregnant',
                          'thyroid_surgery', 'I131_treatment', 'query_hypothyroid', 'query_hyperthyroid', 'lithium',
                          'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH_measured', 'T3_measured', 'TT4_measured',
                          'T4U_measured', 'FTI_measured', 'TBG_measured', 'TBG', 'referral_source', 'Class']

                for col in data.columns:
                    if col in column:
                        data[col] = data[col].apply(lambda x: "'" + str(x) + "'")
                    if col not in column:
                        data[col] = data[col].replace('?', "'?'")
                data.to_csv(self.goodDataPath+ "/" + file, index=None, header=True)
                self.logger.log(log_file," %s: Quotes added successfully!!" % file)
        except Exception as e:
            self.logger.log(log_file, "Data Transformation failed because:: %s" % e)

            log_file.close()
        log_file.close()
```

4.7.5 Data Transformation Prediction



```python
from datetime import datetime
from os import listdir
import pandas
from application_logging.logger import App_Logger


class dataTransformPredict:
    def __init__(self):
        self.goodDataPath = "Prediction_Raw_Files_Validated/Good_Raw"
        self.logger = App_Logger()


    def addQuotesToStringValuesInColumn(self):
        try:
            log_file = open("Prediction_Logs/dataTransformLog.txt", 'a+')
            onlyfiles = [f for f in listdir(self.goodDataPath)]
            for file in onlyfiles:
                data = pandas.read_csv(self.goodDataPath + "/" + file)
                column = ['sex', 'on_thyroxine', 'query_on_thyroxine', 'on_antithyroid_medication', 'sick',
                          'pregnant',
                          'thyroid_surgery', 'I131_treatment', 'query_hypothyroid', 'query_hyperthyroid', 'lithium',
                          'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH_measured', 'T3_measured',
                          'TT4_measured',
                          'T4U_measured', 'FTI_measured', 'TBG_measured', 'TBG', 'referral_source', 'Class']

                for col in data.columns:
                    if col in column:
                        data[col] = data[col].apply(lambda x: "'" + str(x) + "'")
                    if col not in column:  # add quotes to '?' values in integer/float columns
                        data[col] = data[col].replace('?', "'?'")
                data.to_csv(self.goodDataPath + "/" + file, index=None, header=True)
                self.logger.log(log_file, " %s: Quotes added successfully!!" % file)

        except Exception as e:
            log_file = open("Prediction_Logs/dataTransformLog.txt", 'a+')
            self.logger.log(log_file, "Data Transformation failed because:: %s" % e)
            raise e
        log_file.close()
```

4.7.6 Output



Thyroid Disease Prediction

Prediction

Upload a CSV file for prediction

(Note: Please upload a CSV File with valid columns)

No file chosen

Download a Sample submission file for reference

4.5 Use-Case Diagram

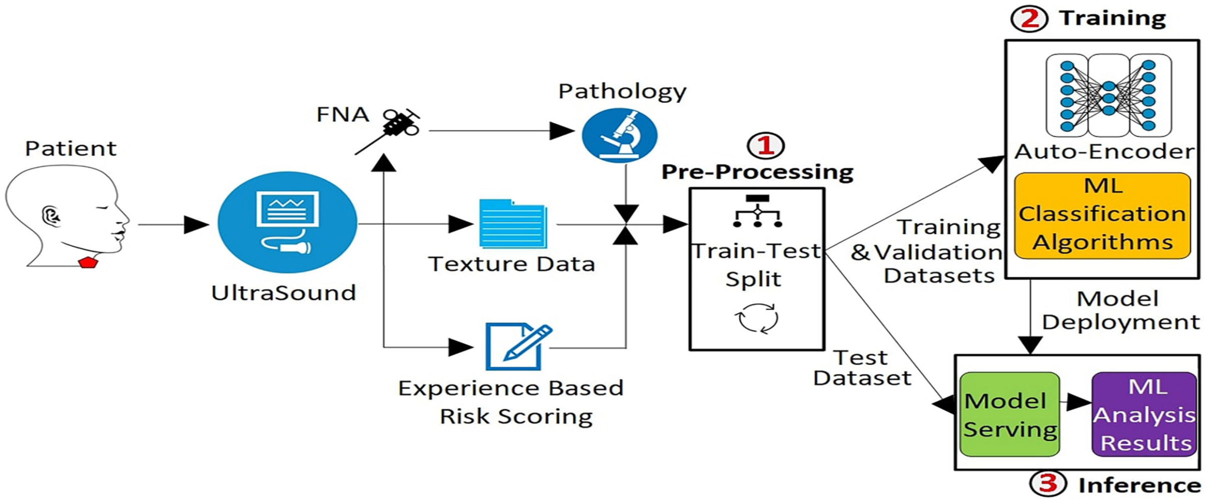


Figure 18 : Use Case Diagram

4.5 Workflow Diagram

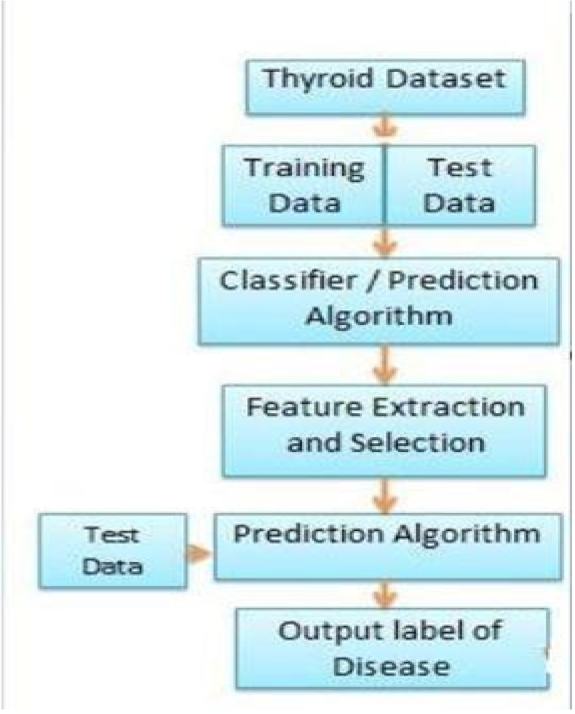


Figure 19 : Workflow Diagram

4.7 Thyroid State Conditions

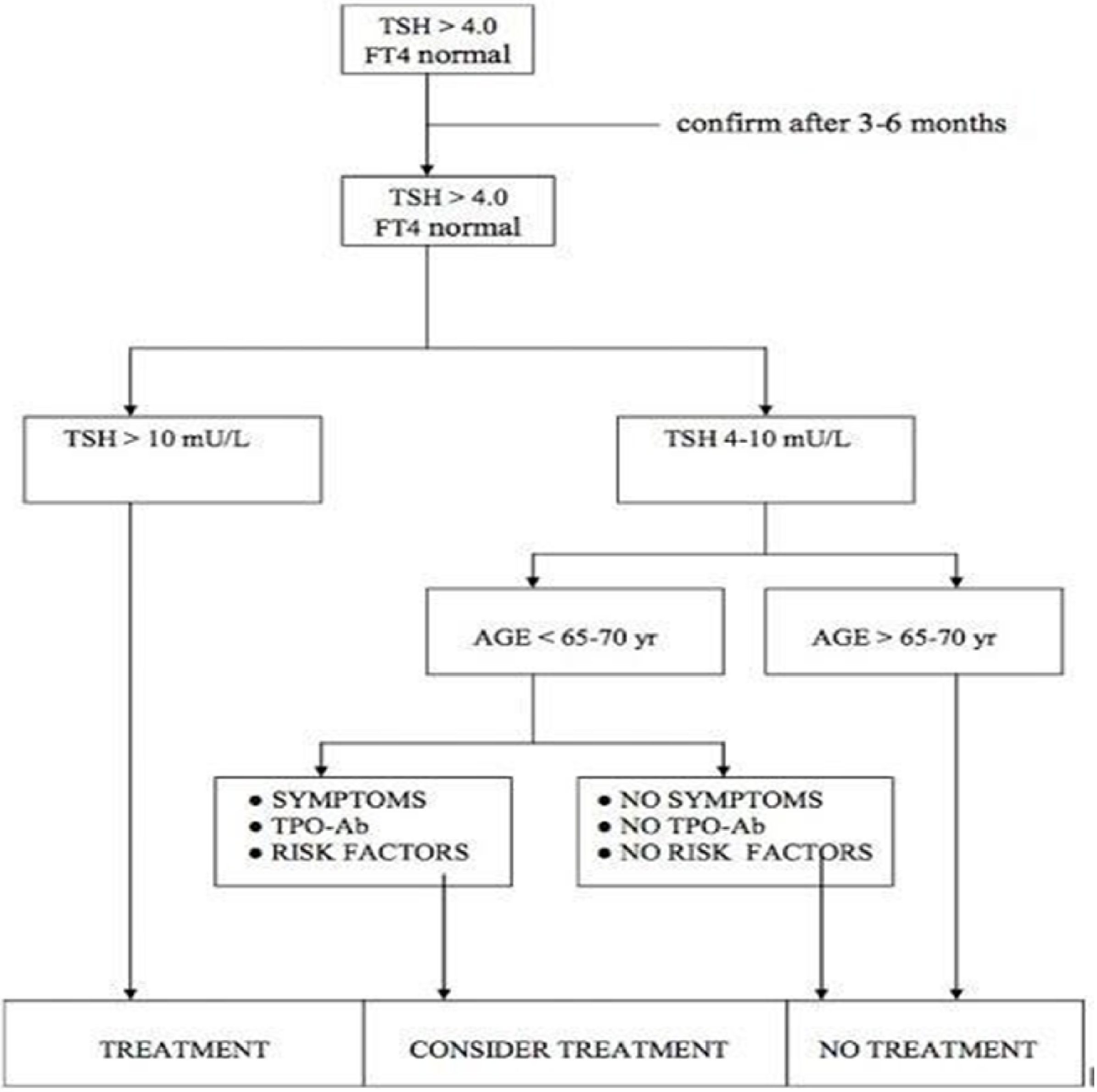
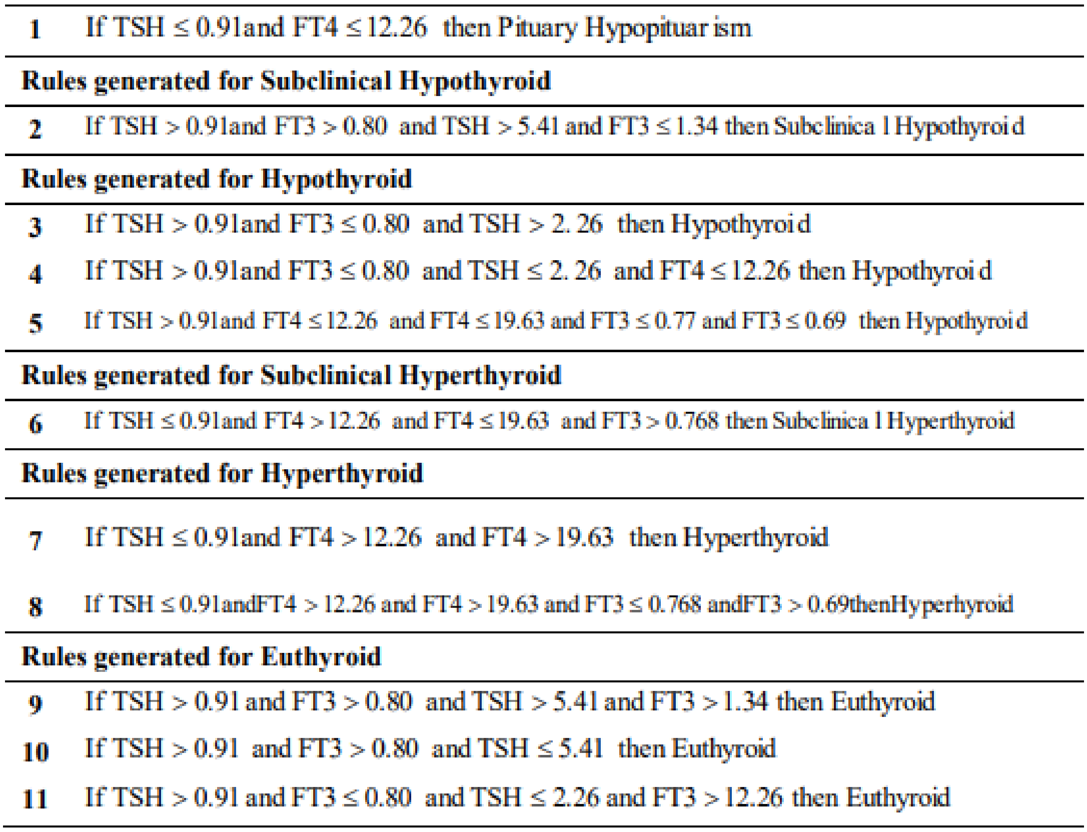


Figure 20 : Thyroid State Conditions Diagram



1	If $TSH \le 0.91$and $FT4 \le 12.26$ then Pituary Hypopituarism
Rules generated for Subclinical Hypothyroid	
2	If $TSH > 0.91$and $FT3 > 0.80$ and $TSH > 5.41$ and $FT3 \le 1.34$ then Subclinical Hypothyroid
Rules generated for Hypothyroid	
3	If $TSH > 0.91$and $FT3 \le 0.80$ and $TSH > 2.26$ then Hypothyroid
4	If $TSH > 0.91$and $FT3 \le 0.80$ and $TSH \le 2.26$ and $FT4 \le 12.26$ then Hypothyroid
5	If $TSH > 0.91$and $FT4 \le 12.26$ and $FT4 \le 19.63$ and $FT3 \le 0.77$ and $FT3 \le 0.69$ then Hypothyroid
Rules generated for Subclinical Hyperthyroid	
6	If $TSH \le 0.91$and $FT4 > 12.26$ and $FT4 \le 19.63$ and $FT3 > 0.768$ then Subclinical Hyperthyroid
Rules generated for Hyperthyroid	
7	If $TSH \le 0.91$and $FT4 > 12.26$ and $FT4 > 19.63$ then Hyperthyroid
8	If $TSH \le 0.91$andFT4 > 12.26 and $FT4 > 19.63$ and $FT3 \le 0.768$ andFT3 > 0.69thenHyperhyroid
Rules generated for Euthyroid	
9	If $TSH > 0.91$ and $FT3 > 0.80$ and $TSH > 5.41$ and $FT3 > 1.34$ then Euthyroid
10	If $TSH > 0.91$ and $FT3 > 0.80$ and $TSH \le 5.41$ then Euthyroid
11	If $TSH > 0.91$ and $FT3 \le 0.80$ and $TSH \le 2.26$ and $FT3 > 12.26$ then Euthyroid

Table 2 : Thyroid State Condition Table

CHAPTER 5

CONCLUSION

5.1 Conclusion

The study goes on to look at some uncommon machine learning algorithms that can be used to diagnose thyroid problems. Numerous simple analyses for accurate and professional thyroid illness diagnosis have been developed and employed in recent years. Various technologies utilised in both publications demonstrate varying accuracy, according to the study.

The neural network outperforms alternative tactics, according to the majority of scholarly articles. On the other hand, because of the fact that the decision tree and the assistance vector machine performed admirably. There is no doubt that professionals all across the world have obtained an understanding of the situation.

Although there has been significant progress in detecting thyroid problems, it is advised that the number of criteria utilised by patients be increased. Thyroid diseases can be difficult to diagnose.More features indicate that a patient must execute a wider range of health-related tasks. Cost-effective as well as time-consuming examinations. As a result, several thyroid algorithms and prediction models have been developed. Thyroid illness must be established, with a minimal number of criteria for a person to diagnose thyroid disease and a way to save time and money for the patient.

This study uses multiple machine learning techniques to predict thyroid risk, including neural networking classifiers, tree-based classifiers, and statistical classifiers. These algorithms are used to predict thyroid risk using a specific dataset, that is, Thyroid Dataset. Precision, recall, F1 score, and accuracy are all used to evaluate the algorithms.

We used a decision tree, Random Forest and KNN algorithm to train our dataset and improve our accuracy in predicting thyroid illness. Based on the user's input, the machine is taught to determine whether the person is normal, hyperthyroid, or hypothyroid. As a consequence, when a user inputs information into a web app, the information is processed in the backend (model) and the result is shown on the screen. Our goal was to provide society with an efficient and precise method of machine learning that might be employed in illness detection applications.

Further research might be conducted by applying image processing of ultrasonic scanning of thyroid pictures to predict thyroid nodules and cancer that are not visible in blood test results.Doing this can ensure that both the types of thyroid disease can be predicted effectively and efficiently.

5.2 Future Scope

Future work of my project is described as follows :-

The current project resided mainly on classification accuracy as the main criteria for measuring the performance of proposed tools. However, future work will focus on other criteria such as classification speed and computational cost.

In order to properly generalise our findings, we will need to broaden the amount of data and variables considered in the future. With more data, the training process is more likely to yield more effective classifiers, as well as a more accurate assessment of the shown performance. Finally, the presence of any secondary thyroid illness connected to the patient might be studied in order to determine if there is a specific extra thyroid condition that can impact hypothyroidism.

In fact, it is not uncommon for people to have many thyroid diseases at the same time. Other factors, such as classification speed and computing cost, will be the subject of future research.

Medical expert systems have advanced dramatically in recent years, and the systems presently available are mature enough for targeted deployment in practice. Expert systems, on the other hand, may be gradually integrated into hospital information systems to improve health-care delivery.

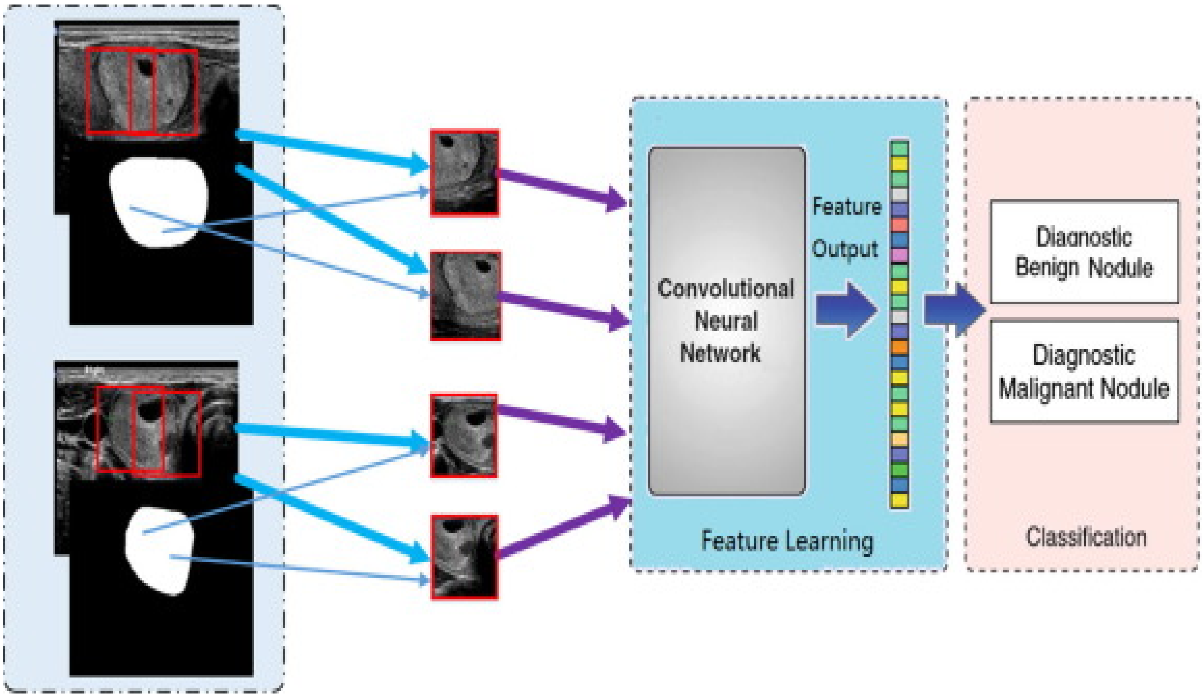


Figure 20 : Future CNN Scope

We can also incorporate Image Classification Techniques to our project with the inculcation of Image Classification algorithms using neural networks through algorithms like ANN, CNN, R- CNN , etc. This image classification technique can prove a boost or very beneficial in Thyroid detection as it can seamlessly double check the predicted value with the diagnosis and match with the predicted value from the image classification technique, providing much effectiveness and better accuracy.

References

A). Thyroid Disease Dataset

Website : UCI Repository
https://archive.ics.uci.edu/ml/datasets/thyroid+disease

B). Journals / Periodicals

[1]. Aversano L, Bernardi ML, Cimitile M, Iammarino M, Macchia PE, Nettore IC, Verdone C. 2021. Thyroid disease treatment prediction with machine learning approaches. Procedia Computer Science
Available : https://doi.org/10.1016/j.procs.2021.08.106

[2] Bastias A, Horvath E, Baesler F, Silva C. 2011. Predictive model based on neural networks to assist the diagnosis of malignancy of thyroid nodules. In: Proceedings of the 41st international conference on computers & industrial engineering.

[3] Kumar V, Webb J, Gregory A, Meixner DD, Knudsen JM, Callstrom M, Fatemi M, Alizad A. 2020. Automated segmentation of thyroid nodule, gland, and cystic components from ultrasound images using deep learning. IEEE Access
Available : https://ieeexplore.ieee.org/document/9044381

[4] Ankita Tyagi and Rikitha Mehra "Interactive Thyroid Disease Prediction System Using Machine Learning Techniques" 5th IEEE International Conference on Parallel, Distributed and Grid Computing(PDGC-2018), 20-22 Dec, 2018, Solan, India.

[5] Vander JB, Gaston EA, Dawber TR. The significance of nontoxic thyroid nodules: final report of a 15-year study of the incidence of malignancy. Ann Intern Med 1968.

[6] . G. Zhang, L.V. Berardi, "An investigation of neural networks in thyroid function diagnosis," Health Care Management Science, 1998, pp. 29-37. Available: http://www.endocrineweb.com/ thyroid.html

[7] Perros P, editor; British Thyroid Association and Royal College of Physicians. Report of the Thyroid Cancer Guidelines Update Group. London: Royal College of Physicians; 2007. Guidelines for the management of thyroid cancer. Availablehttp://www.british-thyroid-association.org/news/Docs/Thyroid_cancer_guidelines_2007.pdf.

[9] International agency for research on cancer, world health organisation GLOBOCAN 2012 estimated cancer incidence , mortality and prevalence worldwide in 2012.

http://globocan.iarc.fr/Pages/fact_sheets_population.aspx

[10] Machine Learning Stanford University, https://www.coursera.org/learn/machine- learning, Instructors,Andrew Ng,Associate Professor, Stanford University; Chief Scientist, Baidu; Chairman and Co-founder, Coursera.