JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING
DHARMAPUR, MAHABOOBNAGAR – 509001
AFFILIATED TO JAWAHARLALNEHRU TECHNOLOGICAL UNIVERSITY,
KUKATPALLY, HYDERABAD – 500072, TELANGANA, INDIA.

"THYROID DISEASE CLASSIFICATION AND PREDICTION USING MACHINE LEARNING"

A Dissertation on MAJOR Project submitted to the Jawaharlal Nehru Technological University, Hyderabad in partial fulfillment of the requirement for the award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted By

L.SRIROOPA	(19361A0584)
D.SREEJA	(19361A0581)
K.RAHUL SHASHANK	(19361A05A1)
B.RUKMINI	(19361A0563)
T.SAI KUMAR	(18361A0577)

Under the Guidance of

Ms. Nazia Tabassum

Asst. Professor

MAY-2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING
DHARMAPUR, MAHABOOBNAGAR – 509001.
Web: www.jpnce.ac.in, Phone: 8886680021.
2022-2023

DEPARTMENTOFCOMPUTERSCIENCEAND ENGINEERING

JAYAPRAKASH NARAYAN COLLEGE OF ENGINEERING

DHARMAPUR, MAHABOOBNAGAR – 509001

(Affiliated to J.N.T.U.H, Approved by A.I.C.T.E)

CERTIFICATE

This is to Certify that the MAJOR Project on "**THYROID PREDICTION AND CLASSIFICATION USING MACHINE LEARNING**" is a bonafide work Done by L.SRIROOPA (19361A0584),D.SREEJA (19361A0581), K.RAHULSHASHANK (19361A05A1), B.RUKMINI (19361A0563), T.SAIKUMAR (18361A0577) in partial fulfillment of the requirement of the award for the degree of Bachelor of Technology in "**Computer Science & Engineering**" J.N.T.U, Hyderabad during the year 2022-2023.

Project Guide **H.O.D.**

Ms. Nazia Tabassum
Asst. Professor,
Dept. Of C.S.E

Dr. K.Guru Raghavendra Reddy
Asst. Professor& head,
Dept. of C.S.E.

External Examiner:

Department of Computer Science Engineering
Jayaprakash Narayan College of Engineering
Mahabubnagar - 509001, Telangana, India.

Ms. Nazia Tabassum
Assistant Professor
Department of Computer Science Engineering
Jayaprakash Narayan College of Engineering, Mahabubnagar

13.05. 2023

Supervisor's Certificate

This is to certify that the work in the thesis entitled **"THYROID PREDICTION AND CLASSIFICATION USING MACHINE LEARNING"** by L.SRIROOPA (19361A0584), D.SREEJA (19361A0581), K.RAHULSHASHANK (19361A05A1) , B.RUKMINI (19361A0563) , T.SAIKUMAR (18361A0577) is a record of a work carried out by them under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science Engineering**.

(Ms. Nazia Tabassum)

Supervisor

ACKNOWLEDGEMENT

We owe a debt of gratitude to **Ms. NAZIA TABASSUM**, **Asst. Professor**, CSE, JPNCE for her admirable guidance an inspirational both theoretically and practically and most importantly for the drive to complete projects successfully. Working under such an eminent guide was our privilege.

We express our sincere thanks to **Dr. K.GURU RAGHAVENDRA REDDY, Asst. Professor** and HOD, CSE, JPNCE of all kinds of consideration, support and encouragement in carrying out this project successfully.

We would also like to thank **Dr. A. SUJEEVAN KUMAR, Principal**, JPNCE for his cooperation and encouragement.

We are grateful to the department of Computer Science and Engineering for providing us with excellent lab and library facilities.

We thank our parents for the love, care and moral support with which we would have been able to complete this project. It has been a constant source of inspiration for all our academic endeavor.

PROJECT ASSOCIATES

1. L.SRIROOPA	(19361A0584)
2. D.SREEJA	(19361A0581)
3. K.RAHUL SHASHANK	(19361A05A1)
4. B.RUKMINI	(19361A0563)
5. T.SAI KUMAR	(18361A0577)

ABSTRACT

The Thyroid is butterfly-shaped endocrine gland which is situated at the base of the human neck. The vital tasks performed by thyroid gland are blood circulation, body temperature control, muscle strength and brain functioning. Any damage or improper functioning of the gland may seriously affect the normal human body functioning. Thyroid disease diagnosis is not a simple task. The normal traditional way includes a proper medical examination and many blood samples for blood tests. Therefore, there is a necessity for a model which detects the thyroid disease at a very early stage. The main goal is to recognize the disease at the early stages with a very high correctness.

The main motive of this project is to develop a model using binary classification which uses Decision Tree ID3 and Naive Bayes Algorithms for the prediction of thyroid disease. If thyroid is present then Naïve Bayes algorithm is applied to check for the thyroid stage in the patient. These algorithms give various levels of precision and accuracy. The thyroid dataset is taken from Kaggle. The Database mainly includes the thyroid patient records having all the necessary patient details in it. These classification methods make the treatment of the thyroid patient simple by reducing further complex procedures with an affordable price.

CONTENT

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Background of study

Thyroid is a butterfly-shaped gland, which is located at the bottom of the throat responsible for producing two active thyroid hormones, levothyroxine (T4) and triiodothyronine (T3) that affect some functions of the body such as: stabilizing body temperature, blood pressure, regulating the heart rate etc. Reverse T3 (RT3) is manufactured from thyroxine (T4), and its role is to block the action of T3. An abnormal function of the thyroid implies the occurrence of hyperthyroidism and hypothyroidism, two of the common thyroid affections.

Hypothyroidism (underactive thyroid or low thyroid) means that the thyroid gland doesn't produce enough of certain important hormones. Without an adequate treatment, hypothyroidism can cause various health problems such as: obesity, joint pain, infertility and heart disease. Hyperthyroidism (overactive thyroid) refers to a condition in which the thyroid gland produces too much of the hormone thyroxin. In this case, the body's metabolism is accelerating significantly, causing sudden weight loss, a rapid or irregular heartbeat, sweating, and nervousness or irritability.

The main factors that affect the thyroid function. It is obvious that factors such as stress, infection, toxins, trauma and certain medication are directly responsible for the improper production of thyroid hormones.

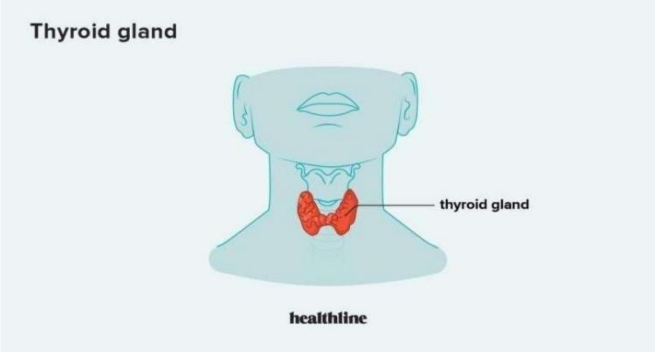


Figure 1.1: Thyroid gland which is present in human neck

Symptoms identification and the early detection of abnormal values of thyroid hormones after clinical investigation will help in establishing the proper diagnostic and to prescribe the right medication. The patient must periodically evaluate his clinical state in order to receive the treatment as long as he needs it.

Thyroid disease diagnosis is not a simple task. It involves many procedures.The normal traditional way includes a proper medical examination and many blood sample For blood tests. Therefore, there is a necessity for a model which detects the thyroid disease d at a very early stage of development.

In medical field machine learning plays an important role for thyroid disease diagnosis as it has various classification models based on which developer can train the model with proper train data set of the thyroid patient and can predict and give the results in an accurate manner with higher degree of correctness. Some recent studies from Mumbai have suggested that congenital hypothyroidism is common in India.The disease occurs in 1 part of 2640 new born children, when compared to the worldwide average range of 1 in 3800 considered. Congenital hypothyroidism can lead to serious complications if not detected in early stages. Therefore, the proposed model serves the goal in early detection of thyroid disease

.

Based on the obtained test values the health care staff can easily examine the condition of the patient and also skip further clinical examinations if not necessary.Hence, this approach proves to be very much beneficial to the health care field. A proper train data set results into an accurate predicting model therefore reducing the overall cost of the thyroid patient treatment and also saving the time. Classification algorithms are most suitable in decision making and also solving the real world problems.

1.2 OVERVIEW

In the Literature Survey, different architectures using Data Mining and Machine Learning techniques are used and evaluated based on their performance of classifying.

A model has been proposed for different Thyroid prediction techniques using data mining approaches. They have considered different data set attributes for prediction and have explained the classification techniques in data mining like Decision Tree, Back propagation Neural Network, SVM and density based clustering. They have analyzed the correlation of T3, T4 and TSH with hyperthyroidism and hypothyroidism.

Thyroid data preprocessing mainly by applying the decision tree algorithm. They have first calculated the mean values of T3, T4 and TSH and considered as the preprocessing stage. Later on they have applied machine learning based feature selection and feature construction. Further they have applied classification based J48 algorithm which is a continuation of ID3 algorithm and calculated the results .

A model has been proposed to study various classification based machine learning algorithms. They have considered train data set from UCI Machine Learning repository and compared and analyzed the performance metric of decision tree, support vector machine and K nearest neighbor .

In, the authors proposed a work which identified Feature Selection Algorithms To improve Thyroid Disease Diagnosis. The main purpose of this research is to analyze the use of filter based (F-Score) and wrapper based (Recursive Feature Elimination) feature selection algorithms on its effect on disease identification and classification. Four classifiers also used such as Multi layer Perceptron, Back Propagation Neural Network, Support Vector Machine, and Extreme Learning Machine. The wrapper-based algorithm produced maximum efficiency and produced a maximum accuracy of 98.14% with an ELM classifier . A general empirical study on various disease diagnosis like Diabetes, Breast Cancer,Heart disease, Thyroid prediction and have compared the accuracy rate by applying SVM, Decision tree and Artificial Neural Networks .

A model has been proposed for Thyroid Prediction System based on data mining classification algorithm. They have used random forest approach to predict the results using weak open source tool used for data mining. Using this tool, they have applied random forest algorithm with 25 thyroid data attributes and predicted the results accordingly. A training model consisting of 21 thyroid causing attributes. They have proposed partial swarm optimization to optimize the support vector machine parameters .A study has been conducted on diagnosis of the thyroid disease using different data mining approaches. They have explained the major cause of the thyroid disease and have also given description about Decision Tree, Naïve Bayes classification and SVM .

Different classification methods are compared. A comparison on various classification methods used to diagnose thyroid disease. They have compared by using Artificial Neural Networks, Radial Based Function, Learning Vector Quantization, Back Propagation Algorithm and Artificial Immune recognition system and concluded the comparison results. Among that they found out that Multi layer Perceptron has the highest accuracy of 96.74% .

A study on different data mining techniques to detect thyroid disease. They have done study on Linear Discriminant analysis, K-fold cross validation, and Decision tree. They have analyzed various splitting rules for the attributes of Decision tree. They have also compared the obtained values .

A model to predict Thyroid Disease using various machine learning techniques. They have considered Logistic Regression and Support Vector Machine as the main Thyroid detection models. They have concluded that these two proposed classifier methods are the best when the number of classes increases in the thyroid prediction model .

There has been a lot of work done to diagnose the discrete diseases in thyroid. Many authors have used various kinds of data mining technique. The authors proved to obtain an adequate approach and certainty to find out the diseases analogous to the thyroid by the work that includes various datasets and algorithms linked with the work that is to be done in the future perspective to accomplish effective and better results. The intent of the paper interprets various techniques of data mining mechanisms and the statistical attributes that is been popularized in the latter years for interpretation of thyroid diseases with the certainty by various authors to attain various prospects and for various approaches. There are various algorithms of machine learning counting random forest, decision tree, naïve Bayes, SVM and ANN that are extensively used in the frequent diseases and in the prognostic problems. There are few functions that are comprised of diseases related to heart diseases.

1.3 MOTIVATION

The motivation behind using machine learning for thyroid classification and prediction is driven by the potential benefits it offers in the field of health care. Here are some key motivations:

1. Improved Diagnostic Accuracy: Thyroid disorders can present with diverse symptoms, and accurately diagnosing them can be challenging. Machine learning algorithms have the potential to analyze large amounts of data and identify patterns that may not be apparent to human clinicians alone. By leveraging machine learning, we can enhance diagnostic accuracy and reduce the chances of misdiagnosis or missed diagnoses.

2. Personalized Treatment: Different thyroid disorders require different treatment approaches. Machine learning algorithms can help identify patterns and factors that influence treatment response, allowing for personalized treatment plans. This can lead to better patient outcomes and improved quality of life.

3. Time and Cost Efficiency: Machine learning algorithms have the ability to analyze data at a faster pace than humans. By automating the analysis process, health care professionals can save time and focus on other critical tasks. Additionally, machine learning can potentially reduce health care costs by optimizing resource allocation and treatment planning.

4. Early Detection and Intervention: Identifying thyroid disorders at an early stage is crucial for effective treatment. Machine learning models can analyze various patient factors and historical data to identify early signs and predict the risk of developing thyroid disorders. This allows for timely interventions and preventive measures

5. Research and Insights: Machine learning can be applied to large-scale datasets to identify novel correlations and associations in thyroid disorders. These insights can contribute to a better understanding of the disease and support further research and development in the field.

6. Support for Health care Professionals: Machine learning models can serve as decision support tools for health care professionals. They can provide additional information and recommendations based on data analysis, aiding clinicians in making informed decisions and improving patient care.

Overall, the motivation behind using machine learning for thyroid classification and prediction is to enhance diagnostic accuracy, personalize treatment plans, improve efficiency, enable early detection, and provide valuable insights to support health care professionals in delivering better patient care.

1.4 IMPORTANCE OF THYROID CLASSIFICATION AND PREDICTION

1. Accurate Diagnosis: Thyroid disorders can manifest with diverse symptoms, and their diagnosis can be complex. Machine learning models can analyze a wide range of patient data, including clinical, laboratory, and imaging information, to provide more accurate and reliable diagnoses. This can help avoid misdiagnosis and ensure appropriate treatment plans are implemented.

2. Personalized Treatment: Different thyroid disorders require specific treatment approaches. Machine learning models can analyze patient data and identify patterns that influence treatment response, enabling personalized treatment plans.

3. Early Detection and Prevention: Timely detection of thyroid disorders is crucial for effective management. Machine learning models can analyze patient data and identify early signs or risk factors for thyroid disorders. This enables early intervention, facilitating prompt treatment and potentially preventing the progression of the disease or the development of complications.

4. Improved Patient Management: Thyroid classification and prediction models can assist healthcare professionals in monitoring and managing patients with thyroid disorders. By analyzing patient data over time, these models can detect trends, track disease progression, and provide insights for optimizing treatment strategies. This can enhance patient management and improve long-term outcomes.

5. Resource Optimization: Efficient allocation of healthcare resources is essential for delivering quality care. Machine learning models can assist in identifying patients at higher risk or in need of more intensive monitoring or intervention. This helps optimize resource allocation, prioritize patient care, and improve the efficiency of healthcare systems.

6. Research and Insights: By analyzing large-scale datasets, machine learning techniques can uncover new patterns, correlations, and insights related to thyroid disorders. These findings can contribute to a better understanding of the disease, its underlying mechanisms, and potential risk factors. They can also support further research, development of new treatments, and refinement of clinical guidelines.

In summary, thyroid classification and prediction using machine learning techniques have the potential to improve diagnostic accuracy, enable personalized treatment, facilitate early detection and prevention, enhance patient management, optimize resource allocation, and contribute to scientific knowledge. These benefits can ultimately lead to better patient care, improved outcomes, and advancements in the field of thyroid disorders.

1.5 PROBLEM

1. Data Availability and Quality: Obtaining a comprehensive and high-quality dataset for thyroid classification can be challenging. The availability of diverse and well-annotated data, including patient demographics, laboratory results, medical history, and imaging data, is crucial for training accurate machine learning models. However, acquiring such data may require collaboration with healthcare institutions, adherence to privacy regulations, and overcoming issues related to data completeness, missing values, and inconsistencies.

2. Class Imbalance: Imbalanced datasets, where one class (e.g., normal thyroid) is significantly more prevalent than others (e.g., different types of thyroid disorders), can lead to biased models. In thyroid classification, certain disorders may be less frequent, making it challenging for the model to learn and accurately predict these minority classes. Techniques like data augmentation, oversampling, or under sampling can be employed to address class imbalance.

3. Interpretable Models: Machine learning models often operate as black boxes, making it difficult to interpret the reasoning behind their predictions. In healthcare, interpretability is crucial for gaining trust from healthcare professionals and patients. Developing models that provide explanations or employing interpretable algorithms like decision trees or rule-based systems can help address this issue.

4. Generalization and External Validation: Machine learning models need to demonstrate good performance not only on the training dataset but also on unseen data from different sources or time periods. Models should be validated externally to assess their generalizability and ensure they can perform accurately in real-world scenarios. Robust evaluation strategies, such as cross-validation or independent validation on external datasets, are necessary to assess model performance.

5. Over fitting and Model Complexity: Complex machine learning models can over fit the training data, resulting in poor generalization to new data. It is crucial to balance model complexity and generalization performance. Regularization techniques, such as L1 or L2 regularization, can help mitigate over fitting. Additionally, feature selection or dimensionality reduction methods can be employed to reduce noise and improve model efficiency.

6. Ethical Considerations and Bias: Machine learning models should be developed with ethical considerations in mind, ensuring fairness and avoiding bias. Biases can emerge from biased training data or reflect existing biases in healthcare practices. Careful attention should be paid to the representativeness of the training data, fairness in predictions across different population groups, and mitigation of any potential bias or discrimination in the models.

Addressing these challenges and considering these considerations is essential for developing accurate, reliable, and ethical machine learning models for thyroid classification and prediction. Collaboration between data scientists, healthcare professionals, and domain experts is crucial to overcome these problems and ensure the successful implementation of machine learning in thyroid healthcare.

1.6 PROPOSED SYSTEM

In this system, the thyroid dataset was taken as input. The input data was taken from the dataset repository. Then, data preprocessing step has to be implemented. In this step, have missing values are to be handled for avoid wrong prediction. If there is any missing values in our input data then replace the missing values by zero or Null values. Next, implement the label encoding for convert the strings into numeric integer value.

Next, data splitting has to be implemented. In this step, splitting the data into test and train is done. Then implement the machine learning algorithms such as decision tree for predicting the disease and classifying the disease into minor or critical or major by using the **Naïve bayes**. Finally, the experimental results shows that the performance metrics such as accuracy, precision, recall and f1 score.

2. LITERATURE REVIEW

1. Title: "Machine Learning Approaches for Thyroid Disease Diagnosis: A Review"

   Authors: Zhang, Y., Meng, F., & Jin, Y.

   Published: 2020

   Summary: This review provides an overview of various machine learning techniques used for thyroid disease diagnosis. It discusses the application of different algorithms, including decision trees, support vector machines, and artificial neural networks, for classifying thyroid disorders based on patient data. The review highlights the strengths and limitations of each approach and discusses future research direction.

2. Title: "Thyroid Disease Classification Using Support Vector Machines"

   Authors: Suresh, S., & Dhanapal, R.

   Published: 2016

   Summary: This study focuses on the application of support vector machines (SVM) for thyroid disease classification. It explores the impact of different kernel functions and SVM parameters on classification performance. The results show that SVM achieves high accuracy in distinguishing between normal and abnormal thyroid function, demonstrating its potential as a reliable diagnostic tool.

3. Title: "Thyroid Disease Diagnosis Using Deep Learning Models"

   Authors: Ma, Y., Li, B., & Wang, X.

   Published: 2019

   Summary: This paper investigates the use of deep learning models, specifically convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for thyroid disease diagnosis. It explores the effectiveness of these models in extracting features from medical images and time-series data, respectively. The study shows promising results, indicating the potential of deep learning for accurate thyroid disease classification.

4. Title:"Thyroid Disease Classification Using Ensemble Learning Technique”

   Authors: Raza, K., et al.

   Published:2017

   Summary: This research focuses on ensemble learning techniques for thyroid disease classification. It compares the performance of different ensemble methods, such as random forests, gradient boosting, and bagging, in predicting thyroid disorders. The study demonstrates that ensemble models can enhance classification accuracy by combining the strengths of multiple base classifier.

5. Title: "Prediction of Thyroid Disease Using Machine Learning Techniques"

   Authors: Praveena, K., & Ramesh, A.

   Published: 2019

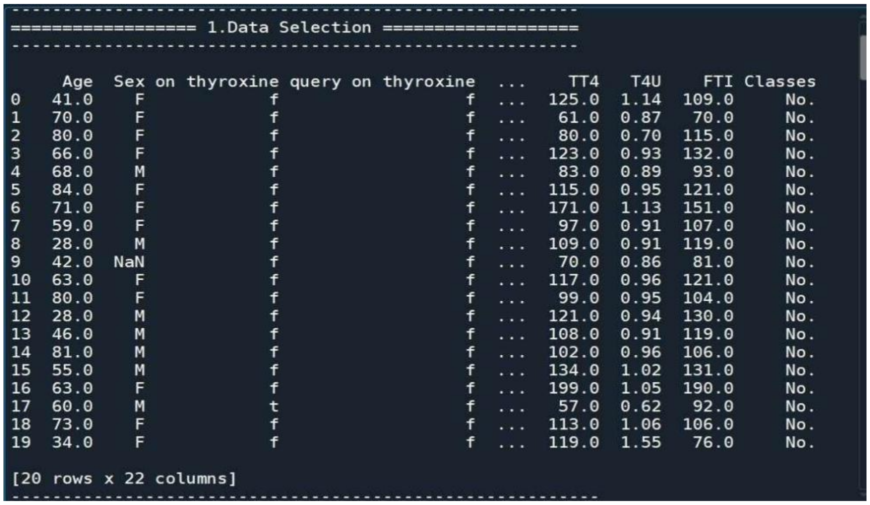
   Summary: This study investigates the prediction of thyroid disease using machine learning algorithms. It explores the use of decision trees, logistic regression, and k-nearest neighbors (KNN) for predicting thyroid disorders based on patient data. The results highlight the effectiveness of these algorithms in accurate disease prediction and emphasize the importance of feature selection for improving prediction performance.

3.METHODOLOGY

3.1 MODULES

3.1.1 Data collection

- The input data was collected from dataset repository like UCI, Kaggle or GitHub.
- In our process, the thyroid disease dataset is used.
- The dataset which contains the information about the patient details such as
- In python, Model have to read the dataset by using the panda's packages.
- Our dataset, is in the form of '.csv' file extension.

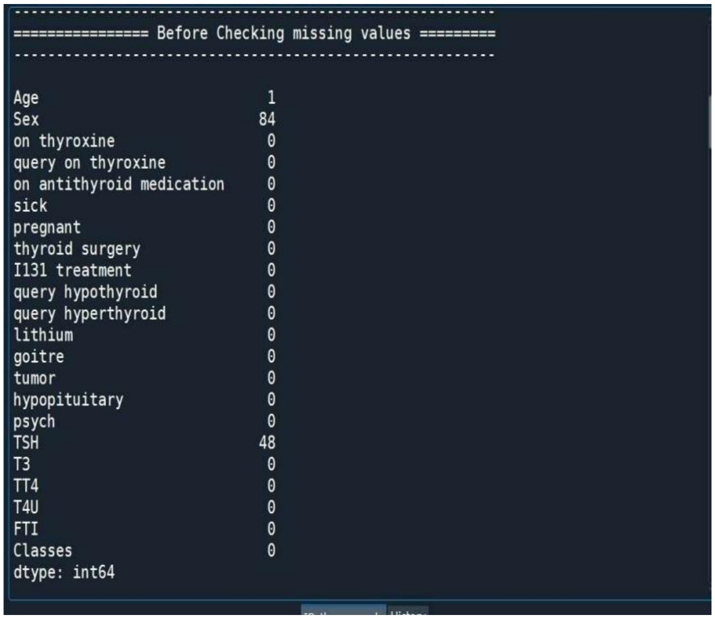


```
----------------------------------------------------------
================== 1.Data Selection ====================
----------------------------------------------------------

     Age  Sex on thyroxine query on thyroxine  ...    TT4   T4U    FTI Classes
0   41.0    F            f                  f  ...  125.0  1.14  109.0     No.
1   70.0    F            f                  f  ...   61.0  0.87   70.0     No.
2   80.0    F            f                  f  ...   80.0  0.70  115.0     No.
3   66.0    F            f                  f  ...  123.0  0.93  132.0     No.
4   68.0    M            f                  f  ...   83.0  0.89   93.0     No.
5   84.0    F            f                  f  ...  115.0  0.95  121.0     No.
6   71.0    F            f                  f  ...  171.0  1.13  151.0     No.
7   59.0    F            f                  f  ...   97.0  0.91  107.0     No.
8   28.0    M            f                  f  ...  109.0  0.91  119.0     No.
9   42.0  NaN            f                  f  ...   70.0  0.86   81.0     No.
10  63.0    F            f                  f  ...  117.0  0.96  121.0     No.
11  80.0    F            f                  f  ...   99.0  0.95  104.0     No.
12  28.0    M            f                  f  ...  121.0  0.94  130.0     No.
13  46.0    M            f                  f  ...  108.0  0.91  119.0     No.
14  81.0    M            f                  f  ...  102.0  0.96  106.0     No.
15  55.0    M            f                  f  ...  134.0  1.02  131.0     No.
16  63.0    F            f                  f  ...  199.0  1.05  190.0     No.
17  60.0    M            t                  f  ...   57.0  0.62   92.0     No.
18  73.0    F            f                  f  ...  113.0  1.06  106.0     No.
19  34.0    F            f                  f  ...  119.0  1.55   76.0     No.

[20 rows x 22 columns]
------------------------------------------------------------
```

Figure 3.1: Dataset after data selection

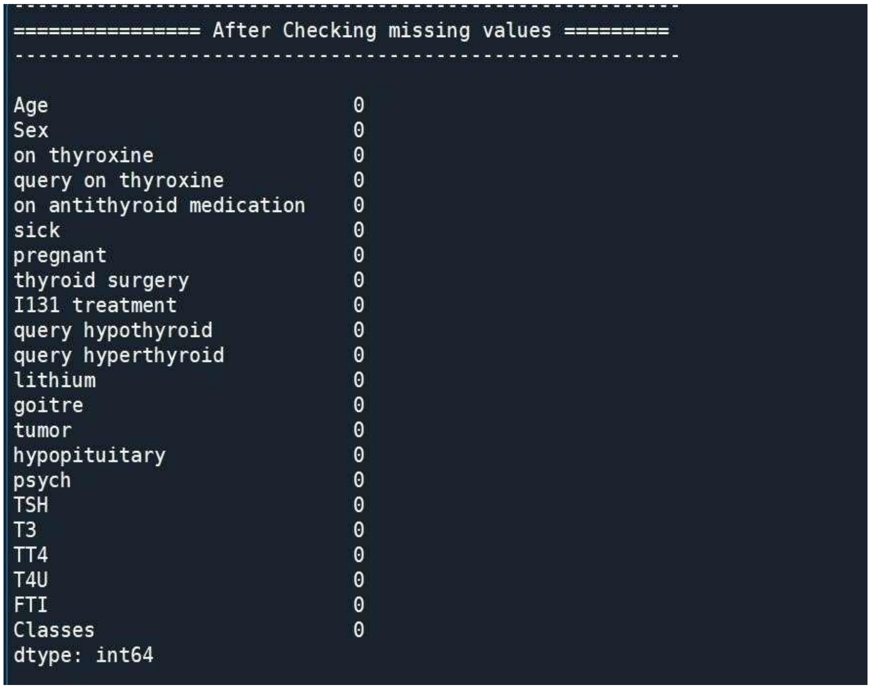
3.1.2 DATA PREPROCESSING:

- Data pre-processing is the process of removing the unwanted data from the dataset.
- Pre-processing data transformation operations are used to transform the dataset into a structure suitable for machine learning.
- This step also includes cleaning the dataset by removing irrelevant or corrupted data that can affect the accuracy of the dataset, which makes it more efficient.
- Data processing is one of the most common tasks in many ML applications. This technique is used to transform raw data into a useful and efficient format. To do the analysis, the dataset needs to be cleaned, standardized, and noise-free. The entire process is known as text preprocessing.
- Missing data removal
- Missing data removal: In this process, the null values such as missing values and Nan values are replaced by 0.
- Missing and duplicate values were removed and data was cleaned of any abnormalities.



```
---------------------------------------------------------
=============== Before Checking missing values =========
---------------------------------------------------------

Age                          1
Sex                         84
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                         48
T3                           0
TT4                          0
T4U                          0
FTI                          0
Classes                      0
dtype: int64
```

Figure 3.2: Data before applying pre-processingg techniques.



```
---------------------------------------------------------
=============== After Checking missing values =========
---------------------------------------------------------

Age                         0
Sex                         0
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                         0
T3                          0
TT4                         0
T4U                         0
FTI                         0
Classes                     0
dtype: int64
```

Figure 3.3: Data after applying the data preprocessing technique

i. Handling Missing Values: Identify missing values in the dataset and decide on an appropriate strategy for handling them. This can involve removing rows or columns with missing values, imputing missing values with statistical measures (e.g., mean, median, mode), or using advanced imputation techniques such as K-nearest neighbors or regression imputation.

ii. Outlier Detection and Treatment: Identify outliers in the dataset that may be erroneous or abnormal data points. Outliers can have a significant impact on the model's performance. Consider using statistical techniques like Z-score, box plots, or clustering algorithms to detect outliers. Decide whether to remove outliers or apply techniques like Winsorization or data transformation to minimize their influence.

iii. Data Normalization/Scaling: Normalize or scale numerical features to ensure they have a similar scale and distribution. Common techniques include min-max scaling (scaling features between a specified range), z-score normalization (standardizing features to have zero mean and unit variance), or logarithmic transformations to handle skewed distributions.

iv. Encoding Categorical Variables: Convert categorical variables into numerical representations that can be processed by machine learning algorithms. This can be done through techniques like one-hot encoding (creating binary columns for each category), label encoding (assigning numerical labels to categories), or ordinal encoding (assigning numerical labels based on the order of categories).

v. Handling Class Imbalance: Address class imbalance if present in the target variable. Techniques like oversampling the minority class, undersampling the majority class, or using synthetic minority oversampling technique (SMOTE) can be employed to balance the classes. Ensure that the chosen technique does not introduce bias or distort the original distribution of the data.

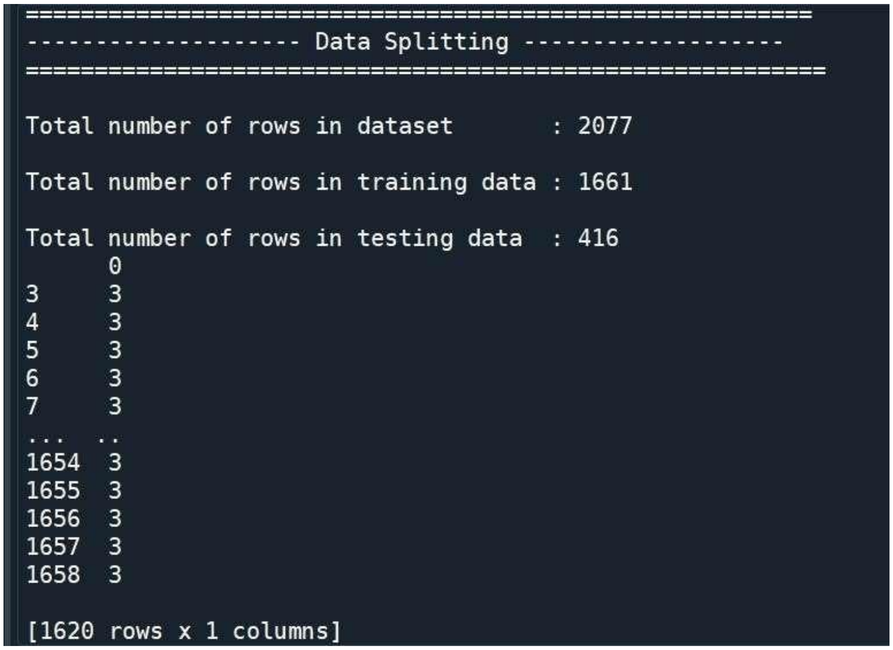
vi. Dimensionality Reduction: Consider reducing the dimensionality of the feature space to improve model efficiency and reduce noise. Techniques such as principal component analysis (PCA) or feature selection algorithms (e.g., correlation analysis, mutual information) can be used to select the most informative features or create new ones that capture most of the variation in the data.

vii. Train-Validation-Test Split: Split the preprocessed dataset into separate training, validation, and testing subsets. The training set is used to train the machine learning model, the validation set is used for hyper parameter tuning and model selection, and the testing set is used to assess the final performance of the model on unseen data.

It is important to carefully implement these preprocessing steps, as they can significantly impact the performance and generalizability of the machine learning model. The specific preprocessing techniques and steps may vary depending on the characteristics of the dataset and the requirements of the thyroid classification and prediction task.

3.1.3 DATA SPLITTING:

- During the machine learning process, data are needed so that learning can take place.
- In addition to the data required for training, test data are needed to evaluate the performance of the algorithm in order to see how well it works.
- In our process, Model considered 70% of the disease dataset to be the training data and the remaining 30% to be the testing data.
- Data splitting is the act of partitioning available data into two portions, usually for cross validator purposes.
- One Portion of the data is used to develop a predictive model and the other to evaluate the model's performance.
- Separating data into training and testing sets is an important part of evaluating data mining models.
- Typically, when user separate a data set into a training set and testing set, most of the data is used for training, and a smaller portion of the data is used for testing.



```
==========================================================
------------------- Data Splitting ------------------
===========================================================

Total number of rows in dataset       : 2077

Total number of rows in training data : 1661

Total number of rows in testing data  : 416
      0
3     3
4     3
5     3
6     3
7     3
...  ..
1654  3
1655  3
1656  3
1657  3
1658  3

[1620 rows x 1 columns]
```

Figure 3.4: Data splitting

3.1.4 Models

3.1.4.1 Support vector machine

In machine learning, Support Vector Machines are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Support vector machines (SVMs) are a set of supervised learning methods used for classification, regression and outlier's detection.

The advantages of support vector machines are: Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples. In Support Vector Regression, the straight line that is required to fit the data is referred to as hyper plane.

The objective of a support vector machine algorithm is to find a hyper plane in an n-dimensional space that distinctly classifies the data points. The data points on either side of the hyper plane that are closest to the hyper plane are called Support Vectors. These influence the position and orientation of the hyper plane and thus help build the SVM.

Hyper parameters in SVM

1.Hyper plane

Hyper planes are decision boundaries that are used to predict the continuous output. The data points on either side of the hyper plane that are closest to the hyper plane are called Support Vectors. These are used to plot the required line that shows the predicted output of the algorithm.

2.Kernel

A kernel is a set of mathematical functions that takes data as input and transform it into the required form. These are generally used for finding a hyper plane in the higher dimensional space. The most widely used kernels include Linear, Non-Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid. By default, RBF is used as the kernel. Each of these kernels is used depending on the dataset.

3.Boundary Lines

These are the two lines that are drawn around the hyper plane at a distance of ε (epsilon). It is used to create a margin between the data points. Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyper plane that has the maximum number of points.

3.1.4.2 Decision Tree

Decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node (e.g., Outlook) has two or more branches (e.g., Sunny, Overcast and Rainy), each representing values for the attribute tested. Leaf node (e.g., Hours Played) represents a decision on the numerical target. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving regression and classification problems too.

The core Algorithm for building decision trees called ID3 by J.R Quinlan which employs a top down, greedy search through the space of possible branches with no backtracking. The ID3 algorithm can be used to construct a decision tree for regression by replacing Information Gain with Standard Deviation Reduction.

3.1.4.3 K nearesst neighbor regression

K nearest neighbors is a simple algorithm that stores all available cases, and predicts the numerical target based on a similarity measure (example distance functions).KNN has been used 26 in statistical estimation and pattern recognition already in the beginning of 1970's as a non parametric technique.

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data .

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

Algorithm

A simple implementation of KNN Regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the k nearest neighbor. KNN regression uses the same distance functions as KNN classification.

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.
Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready.

K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems'-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

3.1.4.4 Linear regression

An interface requirement is a system requirement that involves an interaction with another system. The format of the interface requirement is such that it includes a reference (pointer) to the specific location in the definition document that defines the interface.

Regression analysis is a statistical method that helps us to understand the relationship between dependent and one or more independent variables, In Machine Learning lingo, Linear Regression (LR) means simply finding the best fitting line that explains the variability between the dependent and independent features very well or we can say it describes the linear relationship between independent and dependent features, and in 28 linear regression, the algorithm predicts the continuous features (e.g. Salary, Price), rather than deal with the categorical features (e.g. cat, dog).

Linear regression is probably one of the most important and widely used regression techniques. It’s among the simplest regression methods. One of its main advantages is the ease of interpreting results.

There are two types of regression analysis, they are

1. Simple Linear Regression

2. Multiple Linear Regressions.

1. **Simple Linear Regression**

Simple linear regression is used to find out the best relationship between a single input variable (predictor, independent variable, input feature, and input parameter) & output variable (predicted, dependent variable, output feature, and output parameter) provided that both variables are continuous in nature. This relationship represents how an input variable is related to the output variable and how it is represented by a straight line. Simple Linear Regression algorithm has mainly two objectives:

- Model the relationship between the two variables. Such as the relationship between Income and expenditure, experience and Salary, etc.

• Forecasting new observations. Such as Weather forecasting according to temperature, Revenue of a company according to the investments in a year, etc.

2.Multiple Linear Regression

Multiple Linear Regression is an extension of Simple Linear regression as it takes more than one predictor variable to predict the response variable. Multiple Linear Regression is one of the important regression algorithms which model the linear relationship between a single dependent continuous variable and more than one independent variable.

- For MLR, the dependent or target variable(Y) must be the continuous/real, but the predictor or independent variable may be of continuous or categorical form.
- Each feature variable must model the linear relationship with the dependent variable.
- MLR tries to fit a regression line through a multidimensional space of data-points.

3.1.4.4 Naive Bayes

Naive Bayes algorithm is a supervised learning algorith

m, which is based on Bayes theorem and used for solving classification problems .It is mainly used in text classification that includes a high-dimensional training dataset. Naive Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions. It is a probabilistic classifier, which means it predicts on the basis of the probability of an object. Some popular examples of Naive Bayes Algorithm are spam filtration, Sentimental analysis, and classifying articles. Working of Naive Bayes

1. Convert the given dataset into frequency tables.

2. Generate Likelihood table by finding the probabilities of given features.

3. Now, use Bayes theorem to calculate the posterior probability.

3.2 Interface requirements

An interface requirement is a system requirement that involves an interaction with another system. The format of the interface requirement is such that it includes a reference (pointer) to the specific location in the definition document that defines the interface.

The dataset we are utilizing comes from Kaggle website, which is open to everyone for free. Data set contains some null values and categorical values should be encoded to understand for ML Algorithm

3.2.1 Training and testing the model on data

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The training data must contain the correct answer, which is known as a target. The learning algorithm finds patterns in the training data that map the input data attributes to the target (the answer that you want to predict), and it outputs an ML model that captures these patterns.

Train the classifier using ‘training data set’, tune the parameters and then test the performance of your classifier on unseen ‘test data set’. An important point to note is that during training the classifier only the training is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

Train the classifier using ‘training data set’, tune the parameters and then test the performance of your classifier on unseen ‘test data set’. An important point to note is that during training the classifier only the training is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

Training set: The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning that is to fit the parameters of the classifier.

A training set is a portion of a data set used to fit (train) a model for prediction or classification of values that are known in the training set, but unknown in other (future) data. The training set is used in conjunction with validation and/or test sets that are used to evaluate different models.

Training set (or a training data) is the initial data used to train machine learning models. Training datasets are fed to machine learning algorithms to teach them how to make predictions or perform a desired task.

Test set: A set of unseen data used only to assess the performance of a fullyspecified classifier. The test set is a separate set of data used to test the model after completing the training. It provides an unbiased final model performance metric in terms of accuracy, precision, etc.

A test set is a portion of a data set used in data mining to assess the likely future performance of a single prediction or classification model that has been selected from among competing models, based on its performance with the validation set.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built.

3.2.2 Evaluation

Evaluation allows us to test models against data that has never been used for training. This metric allows us to see how the model might perform against data that it has not yet seen. This is meant to be representative of how the model might perform in the real world.

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future. Evaluation is the process through which we qualify the quality of a system's predictions. To do this, we measure the newly trained model performance on a new and independent dataset. This model will compare labeled data with its own predictions.

3.2.3 Predictions

Prediction refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days. The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.

The word "prediction" can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you're using machine learning to determine the next best action in a marketing campaign. Other times, though, the "prediction" has to do with, for example, whether or not a transaction that already occurred was fraudulent. In that case, the transaction already happened, but you're making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.

3.3 CLASSIFICATION:

- In machine learning, classification refers to a predictive modelling problem where a class label is predicted for a given example of input data.
- Classification is the task of predicting a discrete class label. Regression is the task of predicting a continuous quantity.
- In machine learning, classification is a supervised learning concept which basically categorizes a set of data into classes.
- Before classification, data should be split into test and train.
- Most of data's are used for training and smaller portion of the data's are used for testing.
- Training data is used for evaluate the model and testing data is used for predictive the model.
- After data splitting, implement the classification algorithm.
- Model has two different machine learning algorithms such decision tree for predicting the disease and naïve bayes for classifying the disease into minor or major or critical.

A decision tree is a flowchart-like structure in which each internal node represents a test on a feature (e.g. whether a coin flip comes up heads or tails), each leaf node represents a class label (decision taken after computing all features) and branches represent conjunctions of features that lead to those class.

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e., every pair of features being classified is independent of each other.

4.SYSTEM DESIGN

4.1 REQUIREMENTS

A software requirements specification (SRS) is a description of a software system to be developed. It lays out functional and nonfunctional requirements, and may include a set of use cases that describe user interactions that the software must provide. In order to fully understand one's project, it is very important that they come up with a SRS listing out their requirements, how are they going to meet it and how will they complete the project. It helps the team to save upon their time as they are able to comprehend how are going to go about the project. Doing this also enables the team to find out about the limitations and risks early on. Requirement is a condition or capability to which the system must conform. Requirement Management is a systematic approach towards eliciting, organizing and documenting the requirements of the system clearly along with the applicable attributes. The elusive difficulties of requirements are not always obvious and can come from any number of sources.

4.2 Functional requirements

A Functional Requirement is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender's necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behaviour of the system. Algorithms used in our project are – Random Forest, Support Vector Machine, K-Nearest neighbor, Decision Tree, Logistic Regression, Naïve Bayes.

4.2.1 Benefits of Functional requirements

- A functional requirement document helps you to define the functionality of a system or one of its subsystems.

- Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.

- Errors caught in the Functional requirement gathering stage are the cheapest to fix.

- Support user goals, tasks, or activities

4.2.2 Interface requirements

The dataset we are utilizing comes from Kaggle website, which is open to everyone for free. Data set contains some null values and category values should be encoded to understand for ML algorithm.

Characteristics of data set

- Size of the dataset
- Dimensionality
- Imbalanced classes
- Missing data
- Categorial and Numerical Variables
- Temporal or Sequential Data

4.3 Non-Functional requirements

Non-functional needs are those that aren't directly related to the system's specific functionality. They could be linked to emergent features like reliability, usability, and so on.

Non-Functional Requirement (NFR) specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs. Nonfunctional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users is> 10000. They specify the criteria that can be used to judge the operation of a system rather than specific behaviors. They may relate to emergent system properties such as reliability, response time and store occupancy. Non-functional requirements arise through the user needs, because of budget constraints, organizational policies.

4.3.1 Benefits of Non-functional requirements

- The nonfunctional requirements ensure the software system follows legal and compliance rules.
- They ensure the reliability, availability, and performance of the software system.
- They ensure good user experience and ease of operating the software.
- They help in formulating security policy of the software.
- They ensure good user experience, ease of operating the software, and minimize the cost factor.
- They serve as constraints or restrictions on the design of the system across the backlogs. • They ensure usability and effectiveness of the entire system.

4.3.2 Reliability Requirement

The system should accurately perform when a farmer gives inputs to the algorithm without causing any error.

4.3.3 Usability Requirement

The system is designed for a user-friendly environment so that the users do not face any difficulty.

4.3.4 Implementation Requirements

The system is implemented using python by making use of Machine Learning algorithms such as K- Nearest Neighbor, Multivariate Linear Regression, Support Vector Machine, Naive Bayes Classifier, and Random Forest.

4.4 System configuration

4.4.1 Hardware requirements

- Operating system: windows 7 & above
- Processor: 2 gigahertz (GHz)
- RAM: 4 GB or more

4.4.2 Software requirements

- Language: Python
- IDE- Anaconda, Jupiter Notebook

Python is a simple, general purpose, high level, and object-oriented programming language. It is an interpreted scripting language also. Guido Van Rossum is known as the founder of Python programming. It is a general purpose, dynamic, high-level, and interpreted programming language. It supports Object Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. It is easy to learn yet powerful and versatile scripting language, which makes it attractive for Application Development.

Python's syntax and dynamic typing with its interpreted nature make it an ideal language for scripting and rapid application development. Python can be used on a server to create web applications. It can be used alongside software to create workflows. It can connect to database systems. It can also read and modify files. Python can be used to handle big data and perform complex mathematics. Python can be used for rapid prototyping, or for production-ready software development.

Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.). Python has a simple syntax similar to the English language. Python has syntax that allows developers to write programs with fewer lines than some other programming languages. Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick. Python can be treated in a procedural way, an object-oriented way or a functional way.

Python supports multiple programming patterns, including object-oriented, imperative, and functional or procedural programming styles. Python is not intended to work in a particular area, such as web programming. That is why it is known as multipurpose programming language because it can be used with web, enterprise, 3D CAD, etc. We don't need to use data types to declare variable because it is dynamically typed so we can write a=10 to assign an integer value in an integer variable. Python makes the development and debugging fast because there is no compilation step included in Python development, and edit-test-debug cycle is very fast.

4.4.2.1 Libraries: NumPy, pandas, scikit-learn, Matplotlib

NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed. This tutorial explains the basics of NumPy such as its architecture and environment. It also discusses the various array functions, types of indexing, etc. An introduction to Matplotlib is also provided. All this is explained with the help of examples for better understanding. NumPy is a Python package. It stands for 'Numerical Python'. It is a library consisting of multidimensional array objects and a collection of routines for processing of array.

Pandas are defined as an open-source library that provides high-performance data manipulation in Python. The name of Pandas is derived from the word Panel Data, which means an Econometrics from Multidimensional data. Data analysis requires lots of processing, such as restructuring, cleaning or merging, etc. There are different tools are available for fast data processing, such as NumPy, Skippy, Python, and Panda. But we prefer Pandas because working with Pandas is fast, simple and more expressive than other tools. Pandas is built on top of the NumPy package, means NumPy is required for operating the Pandas. Before Pandas, Python was capable for data preparation, but it only provided limited support for data analysis. So, Pandas came into the picture and enhanced the capabilities of data analysis. It can perform five significant steps required for processing and analysis of data irrespective of the origin of the data, i.e., load, manipulate, prepare, model, and analyze. Scikit-learn are an open-source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection and evaluation, and many other utilities.

Scikit-learn provide dozens of built-in machine learning algorithms and models, called estimators. Each estimator can be fitted to some data using its fit method. In scikit-learn, preprocessors and transformers follow the same API as the estimator objects they actually all inherit from the same Base Estimator class. The transformer objects don't have a predict method but rather a transform method that outputs a newly transformed sample matrix. Scikit-learn are probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, and clustering and dimensionality reduction. Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc.

4.4.2.2 Software: Anaconda

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of opensource packages and libraries. Search our cloud-based repository to find and install over 7,500 data science and machine learning packages. With the conda-install command, you can start using thousands of open-sourceConda, R, Python and many other packages. Individual Edition is an open source, flexible solution that provides the utilities to build, distribute, install, update, and manage software in a cross-platform manner. Conda makes it easy to manage multiple data environments that can be maintained and run separately without interference from each other.

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing, that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for things other than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages.

Anaconda Navigator is a desktop graphical user interface (GUI) included in Anaconda distribution that allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages on Anaconda Cloud or in a local Anaconda Repository, install them in an environment, run the packages and update them. It is available for Windows, macOS and Linux.

Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPIas well as conda package and virtual environment manager. It also includes a GUI, Anaconda navigator as a graphical alternative to the command line interface.

The big difference between conda and the pip package manager is in how package dependencies are manages, which is a significant challenge for python data science and the reason conda exists. In contrast, conda analyses the current environment including everything currently installed and together with any version limitations specified.

Open source packages can be installed from the Anaconda repository, Anaconda cloud, or the users own private repository or mirror using the conda install command. The default installation of Anaconda 2 includes python 2.7 and anaconda 3 includes python 3.7 however, it is possible to create new environments that include any version of python packaged with anaconda.

4.4.2.3 Visual Studio Code

Visual Studio Code (VS Code) is a free, open-source, and cross-platform code editor developed by Microsoft. It is designed for building and debugging modern web and cloud applications, and supports multiple programming languages, including JavaScript, TypeScript, Python, and C++.

VS Code offers a rich set of features, such as intelligent code completion, debugging, integrated terminal, Git control, and customizable extensions. It also has a clean and intuitive user interface, making it easy to use for developers of all levels.VS Code also has integrated support for live collaboration, allowing multiple developers to work on the same codebase in real-time. This makes it an ideal tool for team-based projects and remote collaboration.

VS Code also has integrated support for live collaboration, allowing multiple developers to work on the same codebase in real-time. This makes it an ideal tool for team-based projects and remote collaboration. Overall, VS Code is a versatile code editor that is well-suited for a wide range of development projects. Whether you're working on a small personal project or a large-scale enterprise application, it offers the tools and features you need to get the job done effectively.

4.5 UML Diagrams

UML is an acronym that stands for Unified Modeling Language. Simply UML is a modern approach to modeling and documenting software. Infact, it is one of the most popular business process modeling techniques. It is based on diagrammatic representation of software components.

4.5.1 Class diagram

Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. A collection of class diagrams represents the whole system. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. It is also known as A structural diagram.

Class Diagram is used to show what the logical entities are involved in the project. Class diagrams contain class name, attributes (also referred to as data fields) and behaviors (also referred to as member functions. The class diagrams are widely used in the modeling of object-oriented systems because they are the only UML diagrams, which can be mapped directly with object-orientedlanguages. Class diagram shows a collection of classes, interfaces, associations, collaborations, and constraints. Here the association exists between user, developer and algorithm. Here attributes, dataset and parameters and methods are like prediction and submit.

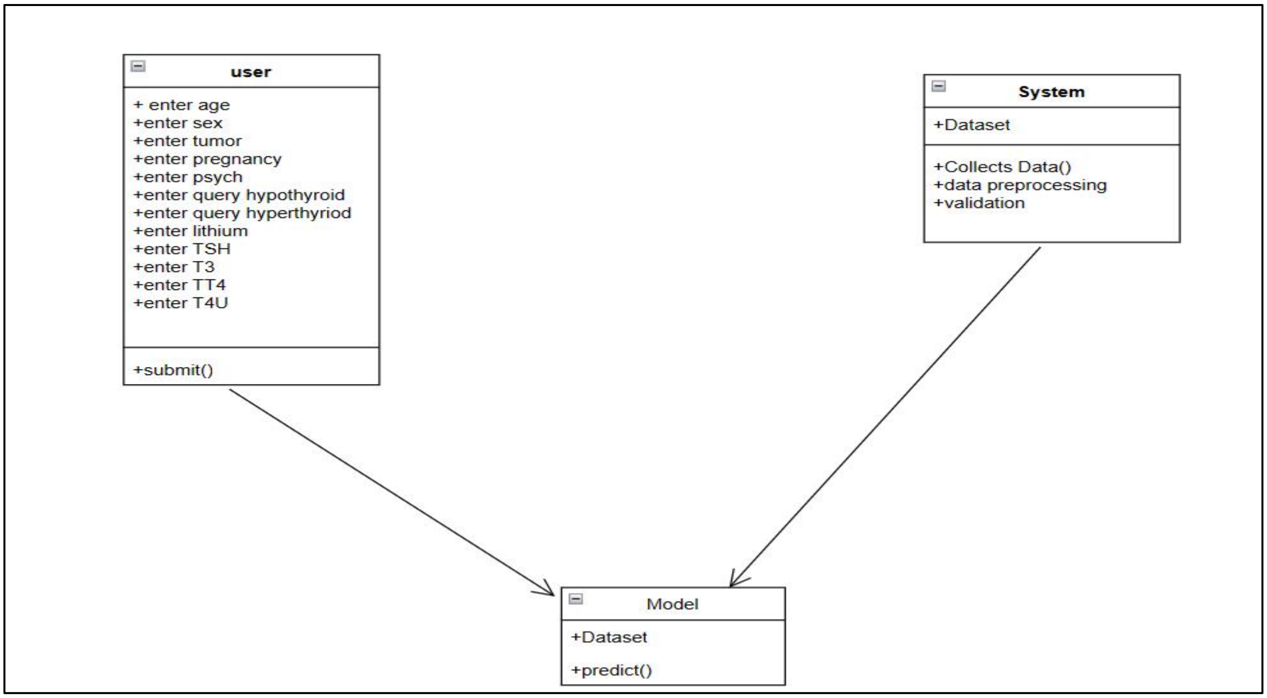


Fig:4.1 Class diagram

4.5.2 Use case diagram

A use case diagram is used to represent the dynamic behavior of a system. Itencapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system. The use cases are represented by either circles or ellipses.

When The Requirements of a system are analyzed, the functionalities are captured in use cases. We Can Say that use cases are nothing but the system functionalities written in an organized manner. The second thing which is relevant to use cases is the actors. Actors can be defined as something that interacts with the system.

Actors can be a human user, some internal applications, or may be some external applications. When we are planning to draw a use case diagram, we should have the following items identified.

Developer collects and prepares the dataset, then the dataset is preprocessed and given to the algorithm, then the data is divided into test data and train data. Through the train data algorithm learns and predicts the output. Use-case diagrams illustrate and define the context and requirements of either an entire system or the important parts of the system. You can model a complex system with a single use-case diagram, or create many use-case diagrams to model the components of the system. You would typically develop use-case diagrams in the early phases of a project and refer to them throughout ddevelopment process.

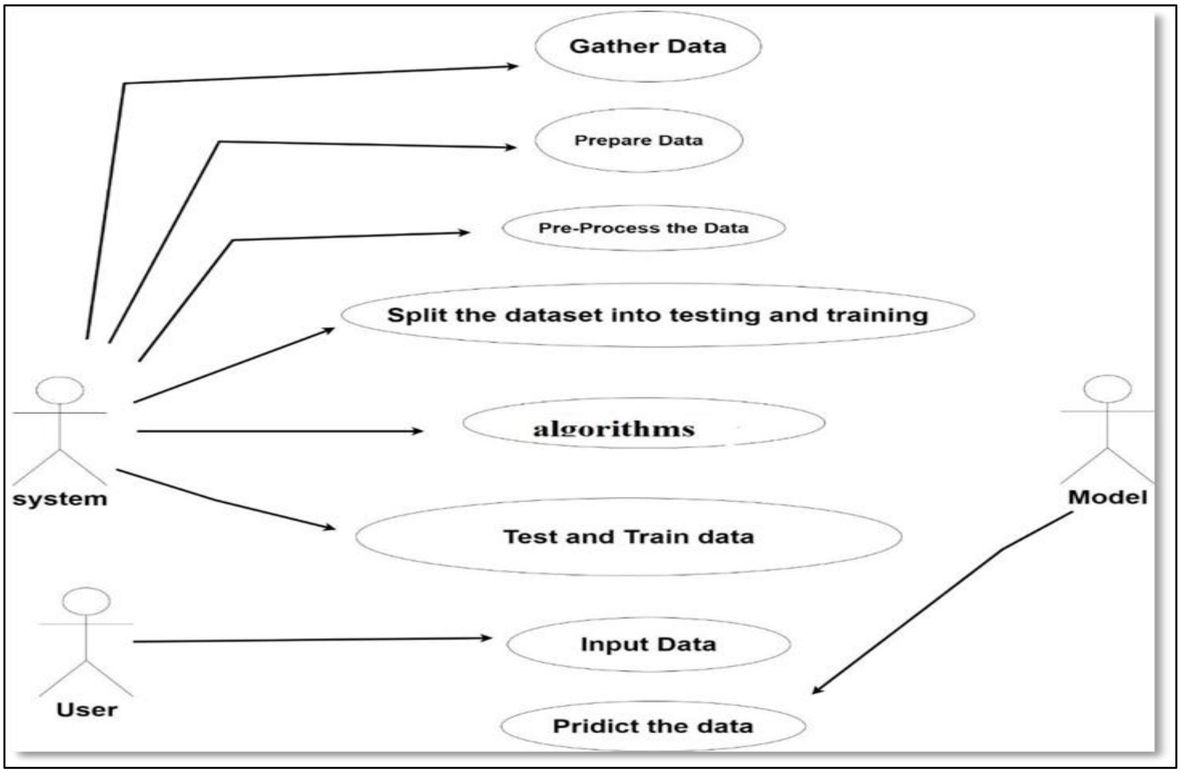


Fig:4.2: Use case Diagram

4.5.3 Sequence diagram

The sequence diagram represents the flow of messages in the system and is also termed as an event diagram. It helps in envisioning several dynamic scenarios. It portrays the communication between any two lifelines as a time-ordered sequence of events, such that these lifelines took part at the run time. A sequence diagram is a type of interaction diagram because it describes how & in what order a group of objects works together.

In UML, the lifeline is represented by a vertical bar, whereas the message flow is represented by a vertical dotted line that extends across the bottom of the page. A sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering. It simply depicts the interaction between the objects in a sequential order. It incorporates the iterations as well as branching Purpose of a Sequence Diagram To model high-level interaction among active objects within a system.

To model interaction among objects inside a collaboration realizing a use case. It either models' generic interactions or some certain instances of interaction.

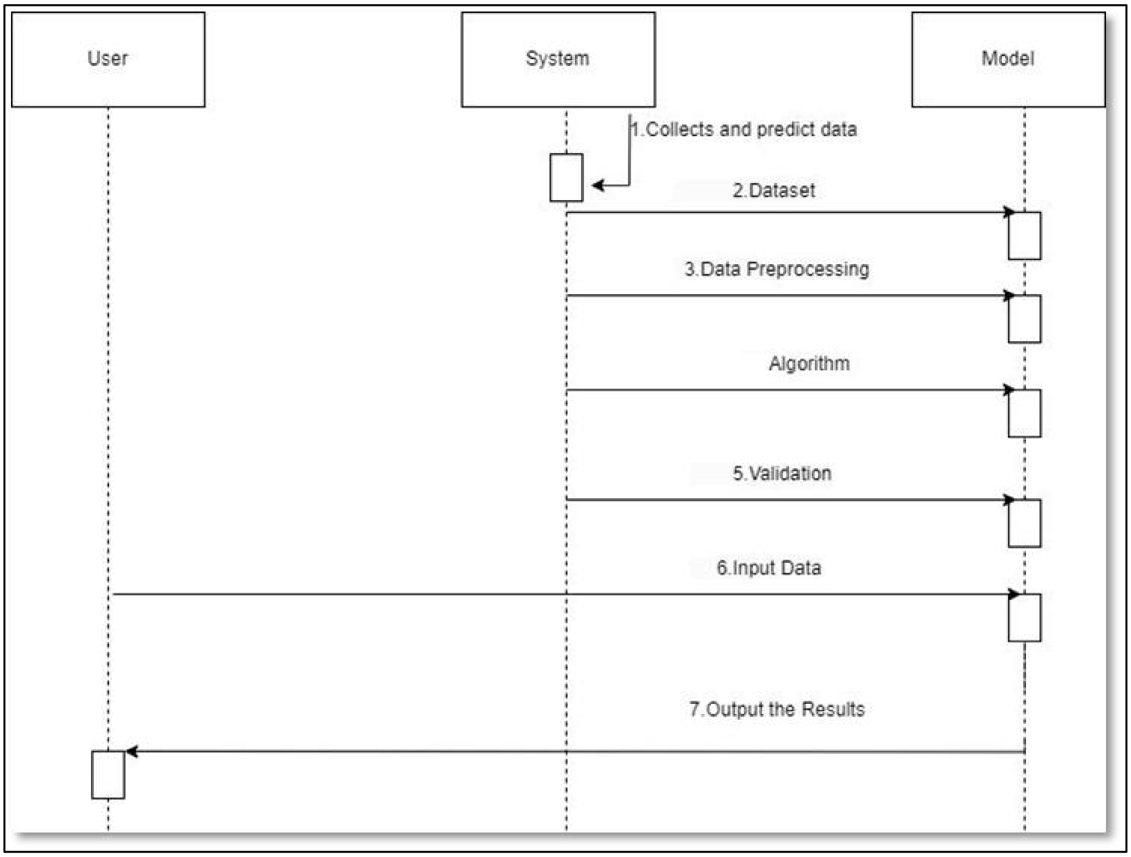


Fig:4.3: Sequence Diagram

4.5.4 Activity diagram

An activity diagram is a behavioral diagram that is it depicts the behavior of the system. An activity diagram portrays the control flow from the starting point to finish point showing the various decision paths that exit while the activity is being executed. Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purpose of activity diagrams is similar to the other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but the activity diagram is used to show the message flow from one activity to another.

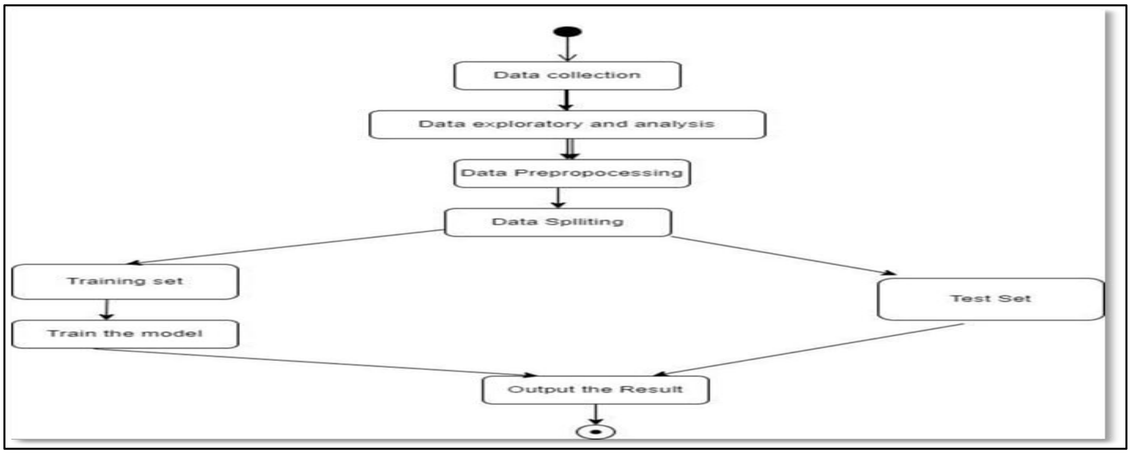


Fig:4.4: Activity Diagram

5. IMPLEMENTATION

5.1 Firstly we need to import required libraries.

```
import NumPy as np
import pandas as pd
```

NumPy is a python package used for performing the various numerical computations and processing of single dimensional and multi-dimensional array elements.

Pandas provide high performance and data manipulation.

5.2 Importing dataset

The dataset is downloaded from Kaggle Website.

```
dataset=pd.read("thyroid.csv")
print(len(dataset))-
```

This is to print the length of the dataset.

```
df= pd.DataFrame(dataset)
```

CSV files are used to store a large number of variables or data. They are incredibly simplified spreadsheets - think Excel- only the content is stored in plain text. And the CSV module is a built-in function that allows Python to parse these types of files.

5.3 Preparing data for training

Initially , we have to find out whether there are any null values in the dataset using the command,

df.isnull().values.any(), the it will return true or false.

If it returns true, then we have to remove the null values from the dataset using the command,

df.dropna(inplace=True),

it will delete all the null values from the dataset

where ever it presents.

5.4 Dividing data into training and testing data sets

We divided the data into 80% training and 20% testing. Train test split is a technique , which can be used for classification or regression problems and can be used for supervised machine learning algorithm. The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred as the training data set. The Second subset is not used to train the model, instead the input element of the dataset is provided to the model, and then predictions are made and compared to the expected values. from sklearn.model_selection import train_test_split

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)

5.5 Feature scaling

```
from sklearn. Preprocessing import StandardScaler
sc=StandardScaler()
#x_train=sc.fit_transform(x_train)
#x_test=sc.transform(x_test)
```

5.6 Training the Algorithm

We train random forest algorithm to solve the regression problem (calculating yield).

1. Support Vector Machine

```
reg = SVR(kernel = 'rbf')
reg.fit(x_train,y_train)
y_pred=reg.predict(x_test)
```

2. Decision tree

```
de=DecisionTreeRegressor(random_state=0)
de.fit(x_train,y_train)
y_pred=de.predict(x_test)
```

3. K-Nearest Neighbor

```
RegModel = KNeighborsRegressor(n_neighbors=2)
#Printing all the parameters of KNN
print(RegModel)
#Creating the model on Training Data
KNN=RegModel.fit(x_train,y_train)
4546
prediction=KNN.predict(x_test)
```

4. Linear regression

```
model = LinearRegression()
model.fit(x, y)
model = LinearRegression().fit(x_train, y_train)
y_pred = model.predict(x_test)
```

5.7 Evaluating the Algorithm

The metrics used to evaluate an algorithm are Mean Absolute Error, Mean Squared Error and Root Mean Squared Error.

```
from sklearn import metrics
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test,y_pred))
print('Mean Squared Error:',metrics.mean_squared_error(y_test,y_pred))
print('Root Mean Squared
Error:',np.sqrt(metrics.mean_squared_error(y_test,y_pred)))
```

5.7.1 Mean Absolute Error

In machine learning, absolute error refers to the magnitude of difference between the prediction of an observation and the true value of that observation. MAE takes the average of absolute errors for a group of predictions and observations as a measurement of the magnitude of errors for the entire group.Mean Absolute Error calculates the average difference between the calculated values and actual values. It is also known as scale-dependent accuracy as it calculates error in observations taken on the same scale. It is used as evaluation metrics for regression models in machine learning .

With any machine learning project, it is essential to measure the performance of the model. What we need is a metric to quantify the prediction error in a way that is easily understandable to an audience without a strong technical background. For regression problems, the Mean Absolute Error (MAE) is just such a metric. The mean absolute error is the average difference between the observations (true values) and model output (predictions). The sign of these differences is ignored so that cancellations between positive and negative values do not occur. If we didn’t ignore the sign, the MAE calculated would likely be far lower than the true difference between model and data.

5.7.2 Mean Square Error

The mean squared error tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line these distances are the “errors” and squaring them., It’s called the mean squared error. The smaller the means squared error, the closer you are to finding the line of best fit. Depending on your data, it may be impossible to get a very small value for the mean squared error.

The mean squared error (MSE) tells us how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line and squaring them. The squaring is necessary to remove any negative signs.

The Mean Squared Error or Mean Squared Deviation of an estimator measures the average of error squares i.e., the average squared difference between the estimated values and

true value. It is a risk function, corresponding to the expected value of the squared error loss. It is always non – negative and values close to zero are better. The MSE is the second moment of the error (about the origin) and thus incorporates both the variance of the estimator and its bias.

5.7.3 Root Mean Square Error

RMSE is a measure of how spread out these residuals are. In other words, it tells you how concentrated the data is around the line of best fit. The square root of the Mean Square Error is Root Mean Square 4748.

RMSE is the standard deviation of the errors which occur when a prediction is made on a dataset. This is the same as MSE (Mean Squared Error) but the root of the value is considered while determining the accuracy of the model. Root mean square error or root mean square deviation is one of the most commonly used measures for evaluating the quality of predictions.

It shows how far predictions fall from measured true values using Euclidean distance. Based on a rule of thumb, it can be said that RMSE values between 0.2 and 0.5 shows that the model can relatively predict the data accurately. In addition, Adjusted R-squared more than 0.75 is a very good value for showing the accuracy. In some cases, Adjusted R-squared of 0.4 or more is acceptable as well.

6. TESTING

Generally testing can be done either manually or automatically. In the crop yield prediction system Manual testing is done. During testing, validation of this application is done, and it is checked for any defects or errors. If the project contains any error in it, it generates wrong output. To avoid this, manual testing is done. In order to get the correct output, correct input must be given.

Machine Learning models would also need to be tested as conventional software development from the quality assurance perspective. Machine Learning represents a class of software that learns from a given set of data and then makes predictions on the new data set based on its learning. In other words, the Machine Learning models are trained with an existing data set in order to make the prediction on a new data set

.

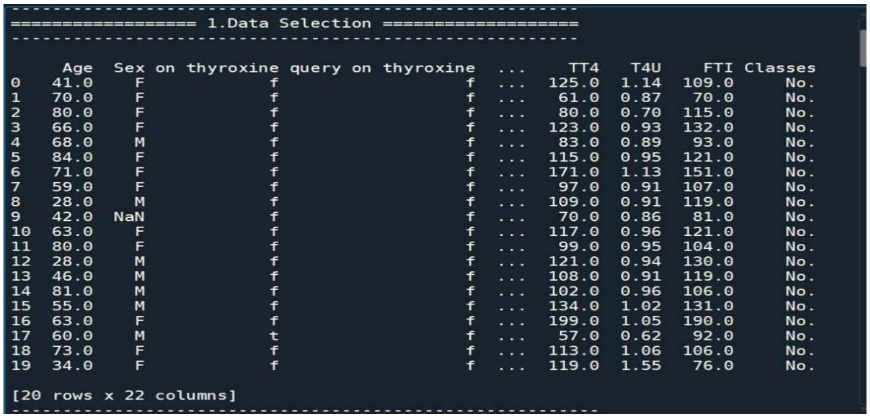
We have considered the dataset and divided the dataset to training set and testing set. Training set (or a training data) is the initial data used to train machine learning algorithms to teach them how to make predictions or perform a desired task.

A test set is a portion of a data set used in data mining to assess the likely future performance of a single prediction or classification model that has been selected from among competing models, based on its performance with the validation set.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred as the training dataset. The second subset is not used to train the model, instead the input element of the dataset is provided to the model, then predictions are made and compared to the expected values.The procedure has one main configuration parameter, which is size of the train and test sets. This is mostly expressed as percentage between 0 and 1.52

In a dataset, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. We have divided our dataset into 80% training and 20% testing data, as it resulted in accurate results, we have divided in such manner. Train test split is a technique which can be used for classification or regression problems and can be used for an y supervised machine learning algorithm.

6.1 Testing different Algorithms with different values



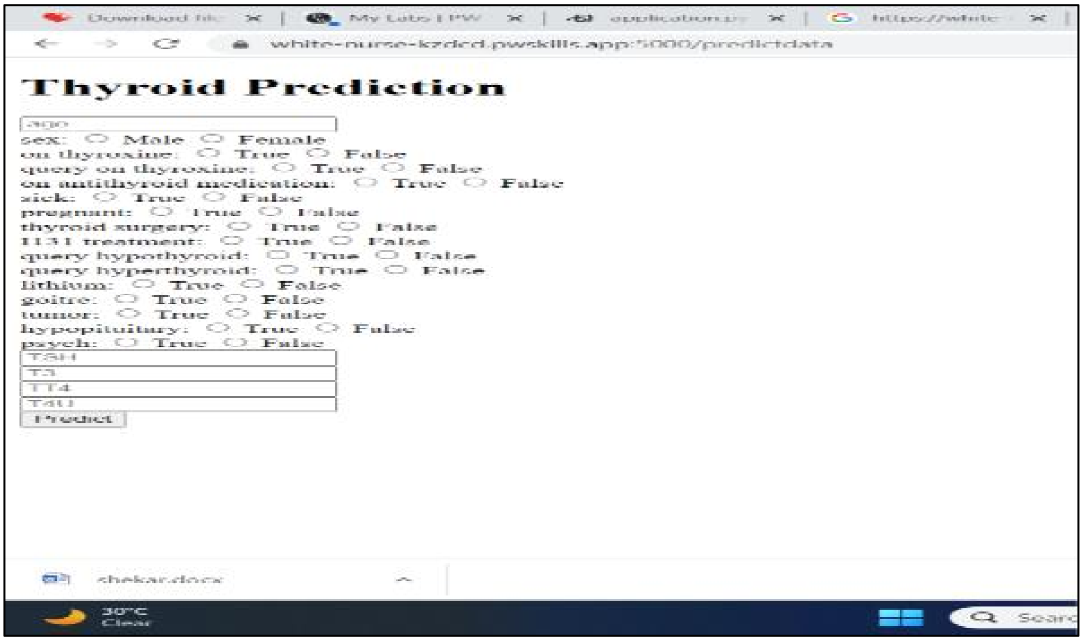
```
-------------------------------------------------------
================== 1.Data Selection ===================
-------------------------------------------------------

     Age  Sex on thyroxine query on thyroxine  ...    TT4  T4U    FTI Classes
0   41.0    F            f                  f  ...  125.0 1.14  109.0     No.
1   70.0    F            f                  f  ...   61.0 0.87   70.0     No.
2   80.0    F            f                  f  ...   80.0 0.70  115.0     No.
3   66.0    F            f                  f  ...  123.0 0.93  132.0     No.
4   68.0    M            f                  f  ...   83.0 0.89   93.0     No.
5   84.0    F            f                  f  ...  115.0 0.95  121.0     No.
6   71.0    F            f                  f  ...  171.0 1.13  151.0     No.
7   59.0    F            f                  f  ...   97.0 0.91  107.0     No.
8   28.0    M            f                  f  ...  109.0 0.91  119.0     No.
9   42.0  NaN            f                  f  ...   70.0 0.86   81.0     No.
10  63.0    F            f                  f  ...  117.0 0.96  121.0     No.
11  80.0    F            f                  f  ...   99.0 0.95  104.0     No.
12  28.0    M            f                  f  ...  121.0 0.94  130.0     No.
13  46.0    M            f                  f  ...  108.0 0.91  119.0     No.
14  81.0    M            f                  f  ...  102.0 0.96  106.0     No.
15  55.0    M            f                  f  ...  134.0 1.02  131.0     No.
16  63.0    F            f                  f  ...  199.0 1.05  190.0     No.
17  60.0    M            t                  f  ...   57.0 0.62   92.0     No.
18  73.0    F            f                  f  ...  113.0 1.06  106.0     No.
19  34.0    F            f                  f  ...  119.0 1.55   76.0     No.

[20 rows x 22 columns]
---------------------------------------------------------
```

Fig 6.1: Interface Implementation

7. RESULTS

The Thyroid disease Predictor is aimed to help users interpret their thyroid function test in order to clarify their diagnosis or track the progression of treatment. Thyroid stimulating hormone (TSH) regulates the amount of thyroid hormone produced by the thyroid gland. TSH levels increase (stimulating the thyroid gland) when thyroid hormone levels fall, and they decrease (reducing thyroid stimulation) when thyroid hormone levels increase. The TSH test is probably the most important thyroid test, often used to screen for thyroid disease and monitor the effectiveness of treatment in a person who has thyroid disease.

The user who wants to test thyroid disease has to fill the mentioned data in the form then after submitting based on the values or the entered details model depicts that whether patient or user is affected by thyroid or not. If the patient has not been affected by thyroid it displays as “User are normal!!” or if patient is in minor stage then displays as “User are in Minor stage...” and same for Major, Critical also.



Thyroid Prediction

age

sex: ○ Male ○ Female

on thyroxine: ○ True ○ False

query on thyroxine: ○ True ○ False

on antithyroid medication: ○ True ○ False

sick: ○ True ○ False

pregnant: ○ True ○ False

thyroid surgery: ○ True ○ False

I131 treatment: ○ True ○ False

query hypothyroid: ○ True ○ False

query hyperthyroid: ○ True ○ False

lithium: ○ True ○ False

goitre: ○ True ○ False

tumor: ○ True ○ False

hypopituitary: ○ True ○ False

psych: ○ True ○ False

TSH

T3

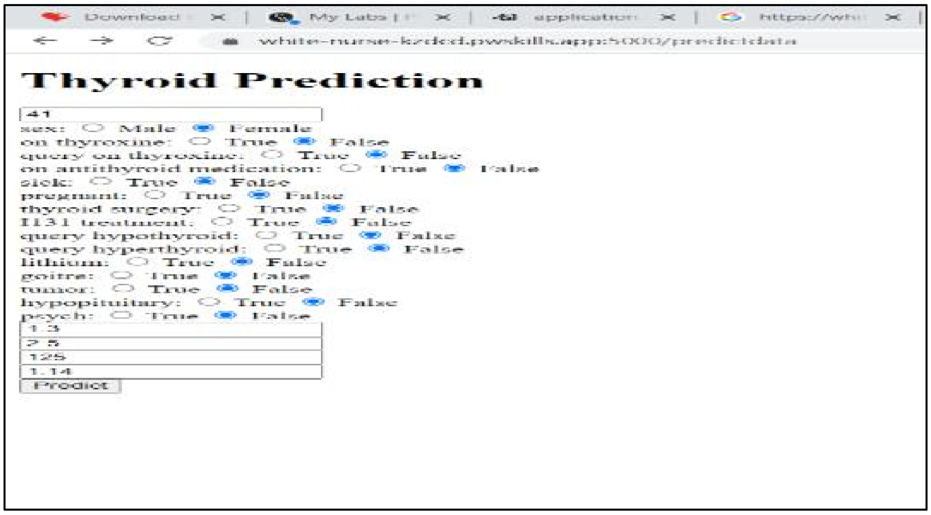
TT4

T4U

Predict

Figure 7.1:User Interface

The patient details includes name, age, gender, TSH, T3, T4U, pregnant, query hypothyroid, query hyperthyroid values after submitting result will be displayed.



Thyroid Prediction

41

sex: ○ Male ● Female

on thyroxine: ○ True ● False

query on thyroxine: ○ True ● False

on antithyroid medication: ○ True ● False

sick: ○ True ● False

pregnant: ○ True ● False

thyroid surgery: ○ True ● False

I131 treatment: ○ True ● False

query hypothyroid: ○ True ● False

query hyperthyroid: ○ True ● False

lithium: ○ True ● False

goitre: ○ True ● False

tumor: ○ True ● False

hypopituitary: ○ True ● False

psych: ○ True ● False

1.3

2.5

125

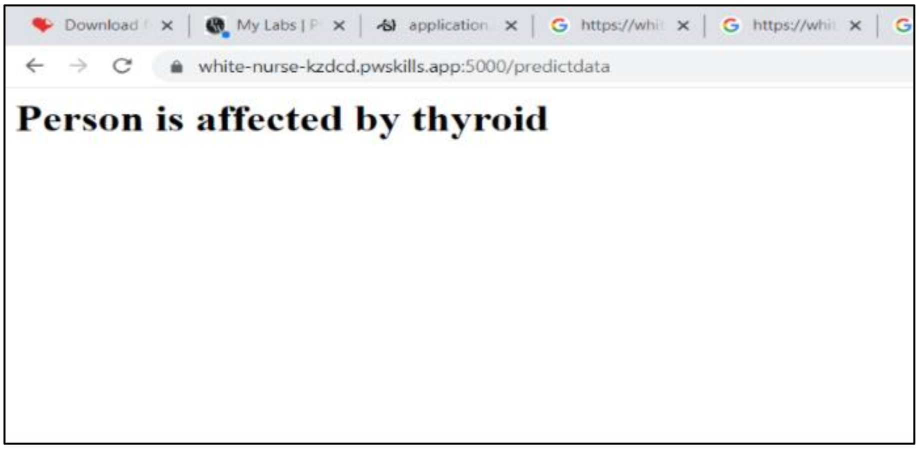
1.14

Predict

Figure 7.2: Entering patient details

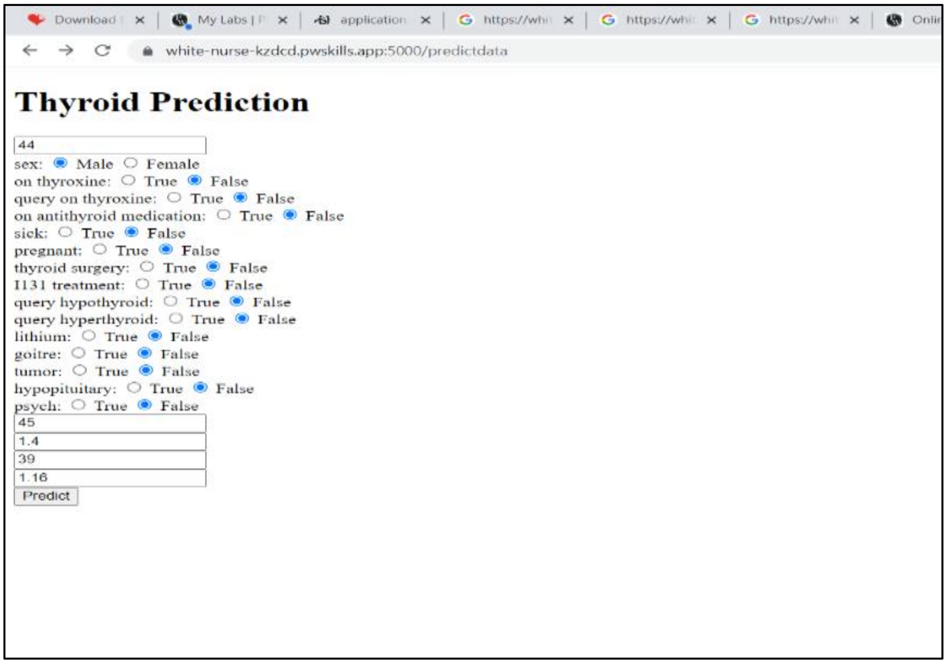
The normal range for TSH is 0.3 - 4 mu/dL and T3 is 1.2 – 2.8 nmol/L and T4U is 71 – 155 nmol/L.Hence, If the given values of TSH,T3,T4U are in between the specified range then the person is not affected by the thyroid. Hence the output will be Normal.

The Figure 7.3 shows that user has entered the details of the patient in the form after submitting, it is displaying in Figure 7.4 that patient is in minor stage of thyroid disease.



Person is affected by thyroid

Figure 7.3: Output for the person affected by the thyroid

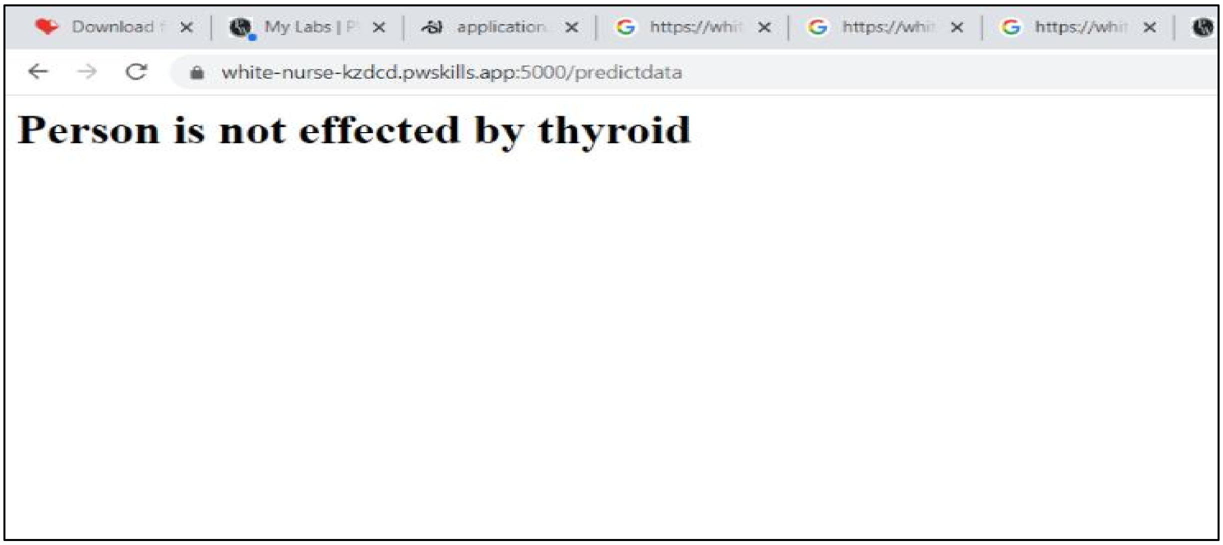


Thyroid Prediction

44
sex: ● Male ○ Female
on thyroxine: ○ True ● False
query on thyroxine: ○ True ● False
on antithyroid medication: ○ True ● False
sick: ○ True ● False
pregnant: ○ True ● False
thyroid surgery: ○ True ● False
I131 treatment: ○ True ● False
query hypothyroid: ○ True ● False
query hyperthyroid: ○ True ● False
lithium: ○ True ● False
goitre: ○ True ● False
tumor: ○ True ● False
hypopituitary: ○ True ● False
psych: ○ True ● False
45
1.4
39
1.16
Predict

Figure 7.4: Entering patient details

Triiodothyronine, or T3, is one of the two important thyroid hormones. Most of the T3 in the blood (99.7 percent) is bound to blood proteins, but only the unbound T3 is active. The free T3 test measures the tiny amount of circulating T3 that is unbound, active T3. In contrast to the measurement of total T3, the advantage to measuring free T3 is that this test is not affected by changes in the thyroid-binding blood proteins.



white-nurse-kzdcd.pwskills.app:5000/predictdata

Person is not effected by thyroid

Figure 7.5: Output for the patient not affected by thyroid

The Figure 7.5 shows that user has entered the details of the patient in the form after submitting, it is displaying in picture .T4, is one of the two important thyroid hormones. Almost all of the T4 in the blood (99.97 percent) is bound to blood proteins, but only the unbound T4 is active. The free T4 test measures the tiny amount of circulating T4 that is unbound, active T4.

Based on the TSH,T3,T4U values of the patient details entered by the user, severity in the thyroid person is classified and displayed on the screen like Minor, Major and Critical stages. The advantage to measuring free T4 is that this test is not affected by changes in the thyroid-binding blood proteins. In most people, the free T4 test can be combined with the TSH test to assess the overall status of the thyroid gland.

Users can analyze one test at a time. Many of these tests are related and the various thyroid hormones interact with each other and are affected by multiple factors. Predictor do not track which lab tests user analyze and do not store any lab values user enter. User are the only one who can see user analysis. Also, User will not be able to return to user results, so if user would like to save them it is best to print them .

7.1 Performance Measures

The Final Result will get generated based on the overall classification and prediction. The persformance of this proposed approach is evaluated using some measures like,

Accuracy

Accuracy of classifier refers to the ability of classifier. It predicts the class label correctly and the accuracy of the predictor refers to how Well a given predictor can guess the value of predicted attribute for a new data.

Precision

Precision is defined as the number of true positives divided by the number of true positives plus the number of false positives.

Recall

Recall is the number of correct results divided by the number of results that should have been returned. In binary classification, recall is called sensitivity. It can be viewed as the probability that a relevant document is retrieved by the query.

Train_test Split	Accuracy
(80,20)	98.2%
(70,30)	90%
(60,40)	87.3%

Table 7.1:Accuracy for different test size

8 .CONCLUSION AND FUTURE ENHANCEMENT

This Model will be very useful to identify the thyroid disease in a patient using classification-based machine learning techniques. The proposed work will be useful to identify the thyroid disease in a patient using classification-based machine learning techniques. These algorithms give various levels of precision and accuracy.

These methods also aid in decreasing the unwanted redundant data from the patient's database. The algorithms used in the proposed we are cost effective and also have good output performance and speed. These classification methods make the treatment of the thyroid patient simple by reducing further complex procedures with an affordable price.

There is no doubt that researchers worldwide have attained a lot of success to diagnose thyroid diseases, but it is suggested to decrease the number of parameters used by the patients for diagnosis of thyroid diseases. More attributes mean a patient has to undergo a greater number of clinical tests which is both cost effective as Well time consuming. Thus, there is a need to develop such type of algorithms and thyroid disease predictive models which require minimum number of parameters of a person to diagnose thyroid disease and saves both money and time of the patient.

So, in the future, There may be chance to work with a larger dataset and We hope that more people from our country will show interest to work on this disease that will help us to find a better solution and able to predict disease in the primary stage with better accuracy. Hope that will help the people of our country to maintain a healthy society.

9.REFERENCES

[1] Ammulu K. and Venugopal T. “Thyroid Data Prediction using Data Classification Algorithm” IJIRST –International Journal for Innovative Research in Science & Technology| Volume 4 | Issue 2 | July 2017.

[2] Ankith Tyagi, Ritika Mehra, Aditya Saxena “Interactive Thyroid Disease Prediction System Using Machine Learning Technique” 5th IEEE International Conference on Parallel, Distributed and Grid Computing (PDGC-2018), 20-22 Dec, 2018, Solan, India.

[3] Aswathi A K and Anil Antony “An Intelligent System for Thyroid Disease Classification and Diagnosis” Proceedings of the 2nd International Conference on Inventive Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978-1-5386-1974-2.

[4] Bibi Amina Begum and Dr.Parkavi “Prediction of thyroid Disease Using Data Mining Techniques” 5Th International Conference on Advanced Computing & Communication Systems (ICACCS), 2019- Vol. 8, Issue 3, March 2021.

[5] Dr. Srinivasan B, K.Pavya “Diagnosis of Thyroid Disease Using Data Mining Techniques: A Study” International Research Journal of Engineering and Technology Volume: 03 Issue: 11 | Nov – 2016.

[6] https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[7] https://www.kaggle.com/datasets/kumar012/hypothyroid

[8] https://www.thehealthsite.com/diseasesconditions/world.

[9] K. Pavya, and B. Srinivasan, “Feature Selection Algorithms To Improve ThyroidDisease Diagnosis”, IEEE International Conference on Innovations in Green Energy and Healthcare Technologies (ICIGEHT’17), pp. 1-5, 02 November, 2017.

[10] M Deepika and Dr. K. Kalaiselvi “A Empirical study on Disease Diagnosis using Data

Mining Techniques.” Proceedings of the 2nd International Conference on Inventive

Communication and Computational Technologies (ICICCT 2018) IEEE Xplore Compliant - Part Number: CFP18BAC-ART; ISBN:978-1-5386-1974-2.

[11] Md. DendiMaysanjaya, Hanung Adi Nugroho and Noor Akhmad Setiawan “A

Comparison of Classification Methods on Diagnosis of Thyroid Diseases” 2015 International Seminar on Intelligent Technology and Its Applications.

[12] Roshan Banu D and K.C.Sharmili “A Study of Data Mining Techniques to Detect

Thyroid Disease” International Journal of Innovative Research in Science, Engineering and Technology (Vol. 6, Special Issue 11, September 2017).

[13] Sumathi A, Nithya G and Meganathan S “Classification of Thyroid Disease using Data Mining Techniques” International Journal of Pure and Applied Mathematics, Volume 119 No.

018, 13881-13890.

[14] Sunila Godara and Sanjeev Kumar "Prediction of Thyroid Disease Using Machine learning Techniques" International Journal of Electronics Engineering (ISSN: sssssss09737383) Volume 10 ; Issue 2 pp. 787-793 June 2018.

[15] V. Vapnik, Estimation of Dependences Based on Empirical Data, Springer, New York, 2012- ISBN:

978-1-4419-2158-1