EasyChair Preprint

№ 10684

Thyroid Disease Detection Using Machine Learning Approach

V Viswanatha, A.C Ramachandra, Rahul R Sankar and Chinmay R Sanjay Mahulika

August 7, 2023

Thyroid Disease Detection Using Machine Learning Approach

Viswanatha V[1]*
Asst.Professor, Electronics and Communication Engineering Department,
Nitte Meenakshi Institute of Technology, Bangalore.

Ramachandra A.C[2]
Professor and Head, Electronics and Communication Engineering Department
Nitte Meenakshi Institute of Technology, Bangalore.

Rahul R Sankar[3]
BE Student, Electronics and Communication Engineering Department
Nitte Meenakshi Institute of Technology, Bangalore.

Chinmay Sanjay Mahulikar[4]
BE Student, Electronics and Communication Engineering Department
Nitte Meenakshi Institute of Technology, Bangalore.

ABSTRACT— **Thyroid disorders are prevalent worldwide and can significantly impact an individual's health and well-being. The accurate detection and diagnosis of thyroid diseases are crucial for effective management and treatment. This abstract presents a machine learning-based thyroid detection program that utilizes advanced algorithms and techniques to improve the efficiency and accuracy of thyroid disease diagnosis. The most common thyroid disorder is hypothyroidism. Hypo- means deficient or under (active), so hypothyroidism is a condition in which the thyroid gland is underperforming or producing too little thyroid hormone. Recognizing the symptoms of hypothyroidism is extremely important. The proposed program leverages a diverse dataset comprising various thyroid-related parameters, including patient demographics, medical history and laboratory test results. By harnessing the power of machine learning algorithms, the program learns intricate patterns and predicts accuracy accordingly.The program employs several machine learning techniques to build a robust and reliable thyroid disease detection model, including feature extraction, feature selection, and classification algorithms. We take the assistance of RandomForestClassifier and StandardScaler Through an iterative training process, the program optimizes the model's performance by minimizing false positives and false negatives, ensuring accurate predictions and reducing the likelihood of misdiagnosis. The program's performance is compared against existing diagnostic methods, including clinical guidelines and expert interpretations of medical professionals, to validate its efficacy and potential for clinical adoption.The results of the evaluation demonstrate that the machine learning-based thyroid detection program achieves superior performance in terms of accuracy and efficiency compared to traditional diagnostic approaches. The program exhibits the potential to assist healthcare professionals in making more accurate and timely diagnoses, leading to improved patient outcomes and optimized resource allocation in healthcare settings.**

Keywords— Thyroid, Hypothyroidism, extraction, selection, classification, RandomForestClassifier, StandardScaler.

I. .INTRODUCTION

SThe thyroid is a small, butterfly-shaped gland located in the front of the neck as shown in **fig.1.** It produces hormones that play a vital role in regulating various bodily functions, including metabolism, growth, and development[1]. The two primary hormones produced by the thyroid gland are thyroxine (T4) and triiodothyronine (T3).Machine learning models can assist in detecting thyroid-related conditions by analyzing various types of data, such as medical records, lab test results, imaging data, and patient symptoms[2]. They can be trained on large datasets containing patient information, including demographic data, medical history, and lab results. By analyzing this data, the models can identify patterns and risk factors associated with thyroid disorders. This helps in stratifying patients into different risk categories, allowing healthcare providers to focus on high-risk individuals for further evaluation[3].

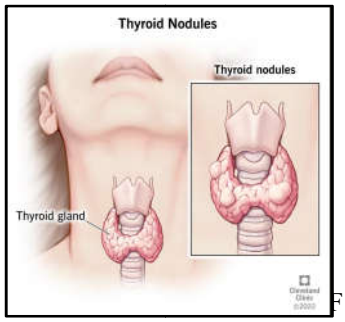

Fig. 1. A representation of a Thyroid Gland

Supervised Learning and Unsupervised Learning are two main categories of machine learning algorithms, differing in their approaches to training data and the types of tasks they can perform. Supervised learning algorithms are trained on labeled data, where the input features and their corresponding output labels or target values are provided. The goal is to learn a mapping function that can predict the correct output for new, unseen input data. Supervised learning algorithms are used for tasks such as classification (assigning input data to predefined classes) and regression (predicting continuous output values). Examples of supervised learning algorithms include the following.

- Decision trees
- Random forests
- Support Vector Machines (SVM)
- Naive Bayes classifiers
- Neural networks (e.g., deep learning)

Unsupervised learning algorithms are trained on unlabeled data, where only the input features are provided without any corresponding output labels[4]. The goal is to discover patterns, structures, or relationships within the data without prior knowledge of the expected outcomes as shown in **fig.2.** Unsupervised learning algorithms are used for tasks such as clustering (grouping similar data points) and dimensionality reduction (representing high-dimensional data in lower dimensions). Examples of unsupervised learning algorithms include:

- K-means clustering
- Hierarchical clustering
- Principal Component Analysis (PCA)
- Association rule mining (e.g., Apriori algorithm)
- Generative adversarial networks (GANs)

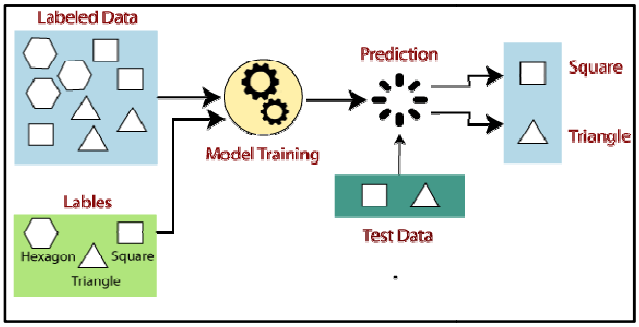

Fig. 2 – Workflow of Supervised ML Algorithms

When it comes to machine learning models for thyroid detection or any other application, it's important to understand the concepts of overfitting and underfitting. Overfitting occurs when a model learns the training data too well, capturing both the underlying patterns and the noise or random fluctuations in the data. In the context of a thyroid detection model, an overfitted model would fit the training data so closely that it would also capture the idiosyncrasies or outliers present in the training set, which may not necessarily represent the true underlying patterns of thyroid-related characteristics. The overfitted model may have very high accuracy on the training data but may not generalize well to new, unseen data. Cross-validation can be used to assess the model's performance on multiple folds of the data and identify potential overfitting. Underfitting occurs when a model fails to capture the underlying patterns in the training data and has poor performance on both the training data and new, unseen data. In the context of a thyroid detection model, an underfitted model would not be able to capture the complexities and patterns present in the data, resulting in low accuracy and limited ability to accurately classify thyroid-related conditions. The model may suffer from high bias, meaning it has systematic errors and fails to generalize well. Feature engineering or adding additional relevant features can enhance the model's ability to capture complex relationships.In thyroid tests, the following abbreviations are commonly used to refer to different parameters:1. TSH (Thyroid-Stimulating Hormone): TSH is produced by the pituitary gland and is responsible for regulating the production of thyroid hormones. It stimulates the thyroid gland to produce thyroxine (T4) and triiodothyronine 2. T3 (Triiodothyronine): T3 is one of the two main thyroid hormones produced by the thyroid gland as shown in **fig.3.**3. TT4 (Total Thyroxine): TT4 refers to the total amount of thyroxine (T4) in the bloodstream. T4 is the primary hormone produced by the thyroid gland and is converted into the active form, T3, in the body. 4. T4U (Thyroxine Uptake): T4U measures the amount of thyroxine (T4) that is bound to proteins in the bloodstream.

5. FTI (Free Thyroxine Index): FTI is a calculated value that estimates the level of free thyroxine (T4) in the blood. It is derived by multiplying the T4U by the total T4 level and dividing it by a specific conversion factor. 6. TGB (Thyroglobulin): Thyroglobulin is a protein produced by the thyroid gland. It is involved in the production and storage of thyroid hormones.

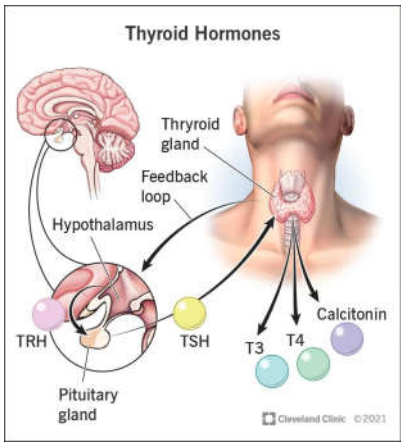

Fig. 3 –Thyroid gland and its hormones

II. LITERATURE SURVEY

1. Title: "Thyroid Disease Detection Using Machine Learning Techniques: A Review".Authors: Gupta, A., & Gupta, A. Published: 2019: This review paper provides an overview of different machine learning algorithms applied to thyroid disease detection. It discusses the use of decision trees, support vector machines, artificial neural networks, and other techniques. The authors also highlight the challenges and future directions in this area[5]. 2. Title: "Thyroid Disease Classification Using Support Vector Machines and Artificial Neural Networks"Authors: Patil, V. C., & Patil, H. Published: 2017: The authors propose a thyroid disease classification system using support vector machines (SVMs) and artificial neural networks (ANNs). They compare the performance of these algorithms on a dataset of thyroid patients and evaluate the accuracy of the classification models[6]. 3. Title: "Thyroid Disease Classification Using Naïve Bayes and K-Nearest Neighbors Algorithms" Authors: Patel, R. B., & Dholakiya, A. Published: 2018: This paper presents a thyroid disease classification system using naïve Bayes and k-nearest neighbors (KNN) algorithms. The authors compare the performance of these algorithms on a thyroid dataset and evaluate their accuracy, precision, and recall[7].4. Title: "Thyroid Disease Diagnosis Using Machine Learning Techniques: A Comparative Study" Authors: El-Dahshan, E. S. A., et al.Published: 2014:The authors compare the performance of different machine learning techniques, including artificial neural networks, support vector machines, and decision trees, for thyroid disease diagnosis. They evaluate the classification accuracy and discuss the advantages and limitations of each approach[8].5. Title: "Thyroid Disease Diagnosis Using Genetic Algorithm-Optimized Neural Networks" Authors: Ahmed, H. A., et al.Published: 2015: This study proposes a genetic algorithm-optimized neural network for thyroid disease diagnosis. The authors use a dataset of thyroid patients and healthy individuals to train and test their model. They demonstrate that the optimized neural network outperforms traditional neural networks in terms of accuracy[9].6. Title: "Thyroid Disease Classification Using Deep Learning Techniques" Authors: Rajesh, P. S., et al. Published: 2020: The authors explore the application of deep learning techniques, such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for thyroid disease classification. They compare the performance of these models and discuss their potential in improving thyroid disease detection[10]-[11].

III. METHODOLOGY

Dataset: Following are the details of Dataset used.

- 2800 training (data) instances and 972 test instances
- Plenty of missing data
- 29 or so attributes, either Boolean or continuously-valued
- two additional databases, also from Ross Quinlan, are also here
- Hypothyroid.data and sick-euthyroid.data
- Quinlan believes that these databases have been corrupted
- Their format is highly similar to the other databases

1 more database of 9172 instances that cover 20 classes, and a related domain theory

Another thyroid database from Stefan Aeberhard3 classes, 215 instances, 5 attributes, the ML methods used are as follows:

1. Data Preprocessing:Data Cleaning: Various data cleaning techniques are used such as replacing missing values, converting categorical variables to numeric, and removing unnecessary columns.Data Imputation: Simple Imputer from scikit-learn is used to impute missing values in columns 'TSH', 'T3', 'TT4', 'T4U', and 'FTI' using the mean strategy.Data Scaling: StandardScaler from scikit-learn is used to scale the feature variables 'x_train' and 'x_test'. The StandardScaler is a preprocessing technique used in machine learning to standardize or normalize the feature variables. It is commonly applied to continuous numerical features to ensure that they have zero mean and unit variance[12]-[13].

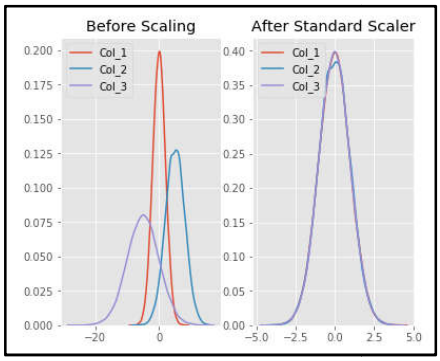

Fig. 4 – Scaling and its application

The StandardScaler works by calculating the mean and standard deviation of each feature in the training data. Then, it subtracts the mean and divides by the standard deviation for each feature in both the training and test data. This process scales the features, making them have a mean of zero and a standard deviation of one.By applying StandardScaler, the features are transformed to have a similar scale, which can be beneficial for many machine learning algorithms. It helps in cases where the features have different scales or units, as it brings them forstandardization as shown in fig.4 .2. Exploratory Data Analysis (EDA):Data Visualization: Various plots and charts are generated using the seaborn library to visualize the distribution and relationships between different variables[14]-[15].3. Machine Learning Modeling: Random Forest Classifier: RandomForestClassifier from scikit-learn is used as the machine learning model. It is initialized with parameters such as the number of estimators (10) and maximum depth (5). The model is trained using the training data ('x_train' and 'y_train') and then used to predict the target variable ('y_pred') for the test data ('x_test'). The model's accuracy score is calculated using the score() method.4. Model Evaluation:Confusion Matrix: The confusion matrix is computed using the confusion_matrix() function from scikit-learn. It provides information about the true positive, true negative, false positive, and false negative predictions made by the model[16]-[19].5.Confusion Matrix Visualization: Overall, the code performs data preprocessing, exploratory data analysis, builds a Random Forest Classifier model, and evaluates its performance using a confusion matrix.The StandardScaler is a preprocessing technique used in machine learning to standardize or normalize the feature variables. It is commonly applied to continuous numerical features to ensure that they have zero mean and unit variance. The StandardScaler works by calculating the mean and standard deviation of each feature in the training data. Then, it subtracts the mean and divides by the standard deviation for each feature in both the training and test data. This process scales the features, making them have a mean of zero and a standard deviation of one. By applying StandardScaler, the features are transformed to have a similar scale, which can be beneficial for many machine learning algorithms. It helps in cases where the features have different scales or units, as it brings them to a standardized range. This preprocessing step can improve the performance and convergence of certain algorithms, such as those based on distance calculations or gradient descent optimization.The Random Forest Classifier is a machine learning algorithm that belongs to the ensemble learning family. It is based on the concept of decision trees and combines multiple decision trees to create a more robust and accurate model.

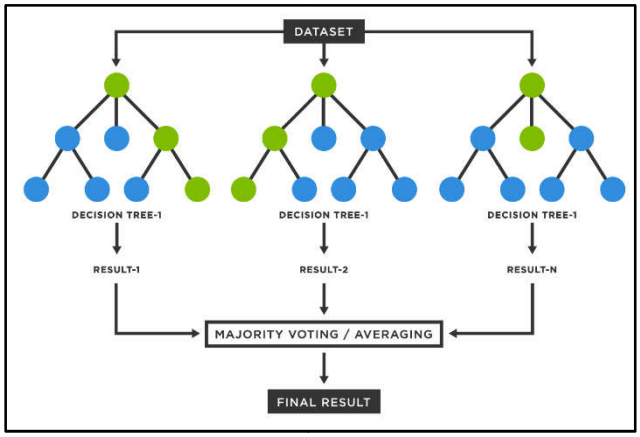

Fig. 5 – RandomForest Workflow

1. Ensemble of Decision Trees:The RandomForestClassifier builds an ensemble of decision trees during the training phase. Each decision tree in the ensemble is built using a random subset of the training data and a random subset of the input features. The randomness helps in reducing overfitting and increasing the model's ability to generalize to unseen data. 2. Random Feature Selection: At each node of a decision tree, a random subset of input features is considered for splitting.This random feature selection process helps to reduce the correlation between individual trees and improves the diversity of the ensemble as shown in fig.5. 3. Decision Tree Construction:Each decision tree is constructed by recursively splitting the data based on selected features. The splits are determined by optimizing a splitting criterion (typically Gini impurity or information gain) to maximize the separation of classes. The process continues until a stopping criterion is met, such as reaching a maximum tree depth or a minimum number of samples required to split a node. 4. Aggregation of Predictions: During prediction, each decision tree in the ensemble independently predicts the class label for a given input sample. For classification tasks, the class with the majority of votes among the trees is selected as the final predicted class label. For regression tasks, the average of the predicted values from individual

trees is taken as the final prediction.Random Forests tend to be robust against overfitting and can handle high-

dimensional datasets. They can capture complex relationships between features and target variables. Random Forests provide estimates of feature importance, indicating which features have a stronger influence on the predictions. They can handle both numerical and categorical features without requiring extensive data preprocessing.RandomForestClassifier has built-in mechanisms for handling missing values and outliers. Overall, the RandomForestClassifier algorithm is known for its versatility, accuracy, and ability to handle a wide range of classification tasks.

IV. Results and Discussion

1. `import pandas as pd`: This imports the pandas library and assigns it the alias 'pd'. Pandas is a powerful library for data manipulation and analysis. It provides data structures and functions to work with structured data, such as data frames, which are commonly used to represent tabular data.2. import matplotlib.pyplot as plt`: This imports the pyplot module from the matplotlib library and assigns it the alias 'plt'. Matplotlib is a popular plotting library in Python, and pyplot provides a simple interface for creating various types of plots and visualizations. 3. `from sklearn.impute import SimpleImputer`: This imports the SimpleImputer class from the impute module in the scikit-learn library. SimpleImputer is used for handling missing values in a dataset by providing strategies for imputing or filling in the missing values with a specified value or statistic. 4. `import statsmodels.api as sm`: This imports the statsmodels library and assigns it the alias 'sm'. Statsmodels is a library for statistical modeling and provides a range of statistical models and tools for data analysis, including linear regression, time series analysis, and more.5. `Import seaborn as sns`: This imports the seaborn library and assigns it the alias 'sns'. Seaborn is a data visualization library that builds on top of matplotlib. It provides a high-level interface for creating aesthetically pleasing statistical graphics and supports various types of plots and statistical visualizations. 6. `from sklearn.model_selection import train_test_split`: This imports the train_test_split function from the model_selection module in scikit-learn. It is used to split a dataset into training and testing subsets. This function randomly divides the data into two portions based on the specified test size or train size, which is useful for evaluating the performance of machine learning models.7. `from sklearn.preprocessing import StandardScaler`: This imports the StandardScaler class from the preprocessing module in scikit-learn. StandardScaler is used for standardizing or normalizing numerical features in a dataset. It scales the features to have zero mean and unit variance, which can help improve the performance of certain machine learning algorithms. 8.`from sklearn.ensemble import RandomForestClassifier`: This imports the RandomForestClassifier class from the ensemble module in scikit-learn. RandomForestClassifier is an ensemble learning algorithm based on decision trees. It constructs multiple decision trees and combines their predictions to make a final prediction. It is commonly used for classification tasks and has parameters to control the number of trees and their depth. 9. `from sklearn.metrics import confusion_matrix, metrics`: This imports the confusion_matrix function and the metrics module from scikit-learn. The confusion_matrix function is used to compute the confusion matrix, which provides information about the true positive, true negative, false positive, and false negative predictions made by a classification model. The metrics module provides various performance metrics for evaluating the performance of machine learning models, such as accuracy, precision, recall, etc.1. `df['TSH'] = imputer.fit_transform(df[['TSH']])`: The 'TSH' column is selected from the DataFrame and passed to the imputer's `fit_transform()` method. The fit_transform() method fits the imputer on the non-missing values of 'TSH' and replaces the missing values in 'TSH' with the computed mean.

2. `df['T3'] = imputer.fit_transform(df[['T3']])`: The same process is applied to the 'T3' column. The imputer is fitted on the non-missing values of 'T3', and the missing values are replaced with the mean of 'T3'. 3. `df['TT4'] = imputer.fit_transform(df[['TT4']])`: Similarly, the 'TT4' column is processed. The imputer is fitted on the non-missing values of 'TT4', and the missing values are replaced with the mean of 'TT4'. 4. `df['T4U'] = imputer.fit_transform(df[['T4U']])`: The 'T4U' column is imputed in the same way as the previous columns.

5. `df['FTI'] = imputer.fit_transform(df[['FTI']])`: Finally, the 'FTI' column is imputed using the imputer, replacing the missing values with the mean of 'FTI'.1. `sns.displot(df['T3'])`: This line creates a histogram-like plot to visualize the distribution of values in the 'T3' column of the DataFrame. The `displot` function automatically determines the appropriate number of bins for the histogram based on the data and displays the distribution of values along the x-axis.2. `sns.displot(df['TT4'])`: Similarly, this line generates a distribution plot for the 'TT4' column, showcasing the distribution of values in that specific variable. 3. `sns.displot(df['T4U'])`: This line produces a distribution plot for the 'T4U' column, displaying the distribution of values in the 'T4U' variable 4. `sns.displot(df['FTI'])`: Lastly, this line creates a distribution plot for the 'FTI' column, illustrating the distribution of values in that particular variable as shown in **fig.6.** and **fig.7**

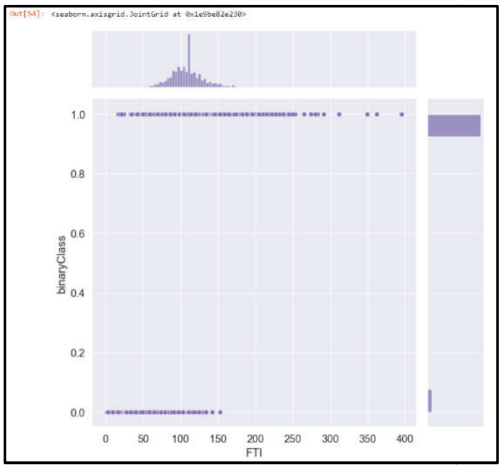

Fig. 6 – BinaryClass v/s FTI

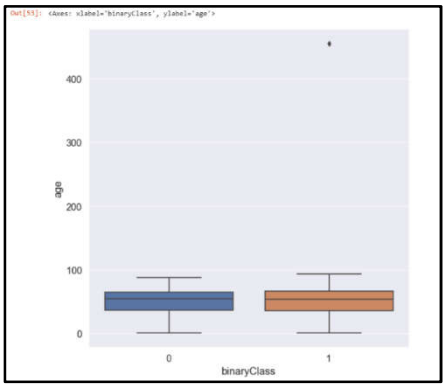

Fig. 7 – Binary Class v/s Age

Heat Map- A heatmap is a graphical representation of data where the individual values within a matrix are represented as colors. It is commonly used to visualize the correlation between variables in a dataset. In the given code snippet, a heatmap is created using the seaborn library to display the correlation matrix of the DataFrame. The resulting heatmap as shown in **fig.8** provides a visual representation of the correlation between variables in the DataFrame 'df'. Darker colors indicate stronger positive or negative correlations, while lighter colors indicate weaker or no correlation. The annotation within each cell provides the actual correlation coefficient values. The heatmap as shown in fig.10 helps to identify relationships and patterns between variables, which can be useful for feature selection or identifying multicollinearity in the dataset.

Accuracy of a ML Model: The accuracy of a machine learning model is a metric that measures how well the model predicts the correct outcomes or labels for a given dataset. It is calculated by dividing the number of correctly predicted instances by the total

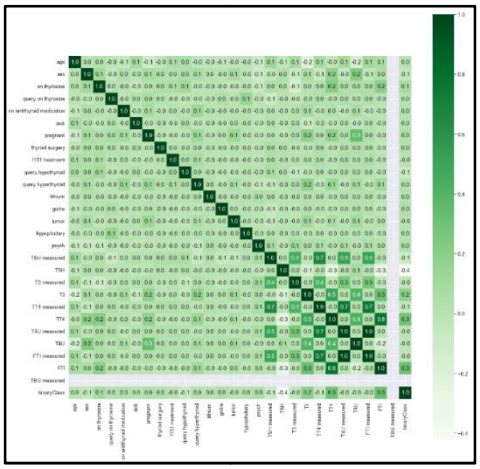

Fig. 8 – Confusion Matrix

number of instances in the dataset as shown in fig.9.

```
In [63]:  1 #checking for precision
          2 score=model.score(x_test,y_test)
          3 score

Out[63]: 0.95864262990456
```

Fig. 9– Accuracy of the model

Here's the formula for accuracy:

$$Accuracy = \frac{No.of\ Correct\ Predictions}{Toatl\ No.of\ Predictions}$$

The accuracy score is typically expressed as a percentage, ranging from 0% to 100%. A higher accuracy indicates that the model is making more correct predictions.However, accuracy alone may not always provide a complete picture of a model's performance, especially in scenarios where the dataset is imbalanced or there are different costs associated with different types of errors. In such cases, other evaluation metrics like precision, recall, F1-score, or area under the ROC curve (AUC-ROC) may be more appropriate. Conclusions: The machine learning model, specifically the Random Forest Classifier, was employed to detect thyroid-related conditions based on the given dataset. The dataset was preprocessed, including handling missing values, converting categorical variables, and scaling the features. The model was trained and evaluated using a test set, and its performance was assessed using a confusion matrix The accuracy score obtained from the model evaluation provides an indication of how well the model

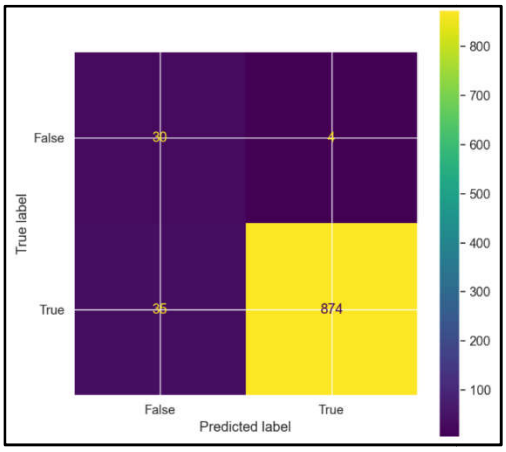

Fig. 10 – Prediction Graph

V. Conclusion and Future Scope

predicts the binary class for thyroid detection.*Future Scope:*Improve Data Quality: Collecting more diverse and comprehensive data, ensuring better representation of various subgroups, can improve the model's performance and generalization.Feature Engineering: Exploring additional features or transforming existing ones may enhance the model's ability to capture relevant patterns and improve accuracy.Algorithm Selection: Experimenting with different machine learning algorithms and ensemble techniques may provide better results or alternative insights into thyroid detection.Hyperparameter Tuning: Fine-tuning the parameters of the selected model or exploring different parameter combinations can optimize the model's performance.Model Interpretability: Investigating the importance of features in the model can provide insights into the factors influencing thyroid detection and enhance the interpretability of the model's predictions.Integration into Clinical Practice: Validating the model's performance on independent datasets and conducting real-world trials can pave the way for practical applications in clinical settings, aiding in thyroid diagnosis and decision-making.

REFERENCES

[1]. Zhang, Xinyu, et al. "Deep convolutional neural networks in thyroid disease detection: a multi-classification comparison by ultrasonography and computed tomography." Computer Methods and Programs in Biomedicine 220 (2022): 106823.

[2]. Chaganti, Rajasekhar, et al. "Thyroid disease prediction using selective features and machine learning techniques." Cancers 14.16 (2022): 3914.

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

[4]. Alyas, Tahir, et al. "Empirical method for thyroid disease classification using a machine learning approach." BioMed Research International 2022 (2022).

[5]. Abbad Ur Rehman, Hafiz, et al. "Performance analysis of machine learning algorithms for thyroid disease." Arabian Journal for Science and Engineering (2021): 1-13.

[6]. Asif, Md Asfi-Ar-Raihan, et al. "Computer aided diagnosis of thyroid disease using machine learning algorithms." 2020 11th International Conference on Electrical and Computer Engineering (ICECE). IEEE, 2020.

[7]. Isa, I. S., et al. "Suitable MLP network activation functions for breast cancer and thyroid disease detection." 2010 second international conference on computational intelligence, modelling and simulation. IEEE, 2010.

[8]. Sonuç, Emrullah. "Thyroid disease classification using machine learning algorithms." Journal of Physics: Conference Series. Vol. 1963. No. 1. IOP Publishing, 2021.

[9]. Kulkarni, Satish N., and A. R. Karwankar. "Thyroid disease detection using modified fuzzy hyperline segment clustering neural network." International Journal of Computers & Technology 3.3 (2012): 466-9.

[10]. Razia, Shaik, and MR Narasinga Rao. "Machine learning techniques for thyroid disease diagnosis-a review." Indian J Sci Technol 9.28 (2016): 1-9.

[11]. A C, R. ., V. . V, K. . K, S. . H, and P. S. . E. “In-Cabin Radar Monitoring System: Detection and Localization of People Inside Vehicle Using Vital Sign Sensing Algorithm”. International Journal on Recent and Innovation Trends in Computing and Communication, vol. 10, no. 8, Aug. 2022, pp. 104-9, doi:10.17762/ijritcc.v10i8.5682.

[12]. V. V, R. A. C, S. B. M, A. Kumari P, V. S. Reddy R and S. Murthy R, "Custom Hardware and Software Integration: Bluetooth Based Wireless Thermal Printer for Restaurant and Hospital Management," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-5, doi: 10.1109/MysuruCon55714.2022.9972714.

[13]. V. V, R. A. C, V. S. R. R, A. K. P, S. M. R and S. B. M, "Implementation of IoT in Agriculture: A Scientific Approach for Smart Irrigation," 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon), Mysuru, India, 2022, pp. 1-6, doi: 10.1109/MysuruCon55714.2022.9972734.

[14]. Viswanatha, V., and R. Venkata Siva Reddy. "Digital control of buck converter using arduino microcontroller for low power applications." 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon). IEEE, 2017.

[15]. Viswanatha, V., Venkata Siva Reddy, R., Rajeswari (2020). Research on State Space Modeling, Stability Analysis and PID/PIDN Control of DC–DC Converter for Digital Implementation. In: Sengodan, T., Murugappan, M., Misra, S. (eds) Advances in Electrical and Computer Technologies. Lecture Notes in Electrical Engineering, vol 672. Springer, Singapore. https://doi.org/10.1007/978-981-15-5558-9_106.

[16]. Dhanamithra, R. G., and R. Satheesh Kumar. "A Comparison Study on Machine Learning Approaches for Thyroid Disease Prediction." 2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS). Vol. 1. IEEE, 2022.

[17]. Sinha, Bam Bahadur, Mohammad Ahsan, and R. Dhanalakshmi. "LightGBM empowered by whale optimization for thyroid disease detection." International Journal of Information Technology (2023): 1-10.

[18]. Abbad Ur Rehman, Hafiz, Chyi-Yeu Lin, and Zohaib Mushtaq. "Effective K-nearest neighbor algorithms performance analysis of thyroid disease." Journal of the Chinese Institute of Engineers 44.1 (2021): 77-87.

[19]. Guleria, Kalpna, et al. "Early prediction of hypothyroidism and multiclass classification using predictive machine learning and deep learning." Measurement: Sensors 24 (2022): 100482.