University of Hertfordshire UH

School of Physics,
Engineering and
Computer Science

MSc Data Science Project

7PAM2002-0509-2024

Department of Physics, Astronomy and Mathematics

Data Science FINAL PROJECT REPORT

Project Title:

Multi-Class Thyroid Function Classification Using Lab Test Data with Machine Learning and Deep Learning

Student Name and SRN:

Om Mahendra Sankhe , 23068627

Supervisor: Dr. Carolyn Devereux

Date Submitted: 27/08/2025

Word Count: 5,200 words

Github Link - https://github.com/OmSankhe224/final-year-project-Om-Sankhe

1 DECLARATION STATEMENT

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science **in Data Science** at the University of Hertfordshire.

I have read the detailed guidance to students on academic integrity, misconduct and plagiarism information at Assessment Offences and Academic Misconduct and understand the University process of dealing with suspected cases of academic misconduct and the possible penalties, which could include failing the project or course.

I certify that the work submitted is my own and that any material derived or quoted from published or unpublished work of other persons has been duly acknowledged. (Ref. UPR AS/C/6.1, section 7 and UPR AS/C/5, section 3.6)

I did not use human participants in my MSc Project.

I hereby give permission for the report to be made available on module websites provided the source is acknowledged.

Student Name printed: Om Mahendra Sankhe

Student Name signature

Sankhe

Student SRN number: 23068627

UNIVERSITY OF HERTFORDSHIRE

SCHOOL OF PHYSICS, ENGINEERING AND COMPUTER SCIENC

Abstract

This project investigates the automated classification of thyroid health conditions using the UCI Hypothyroid dataset. The main goals were to clean and prepare the data, choose the best features, and balance the classes so the models can work well. Different Machine learning and Deep Learning methods were tested, including Logistic Regression, SVM, Random Forest, XGBoost, CNN, GRU, and a custom neural network. Techniques such as Mutual Information feature selection and SMOTE oversampling were employed to improve model robustness and address class imbalance. The experimental results revealed that XGBoost with SMOTE provided the best performance in identifying hypothyroid cases, scoring an F1-score of **0.94** and a Recall **of 0.97**, thereby minimizing the risk of missed diagnoses. The study concludes that ensemble models, particularly when combined with effective preprocessing, offer promising accuracy and reliability for medical diagnostic applications, with strong potential for real-world deployment in thyroid disorder screening systems to support clinicians and improve early detection.

Contents

1 DECLARATION STATEMENT 2
1 Introduction 6
1.1 Background and Motivation 6
1.2 Problem Statement and Relevance 6
1.3 Research Aim and Questions 6
1.4 Structure of the Report 7
2 Background 7
2.1 Introduction 7
2.2 Review of Key Literature 7
2.2.1 Knowledge Graph and Deep Learning for Thyroid Diagnosis 7
2.2.2 Comparative Analysis of ML Algorithms on Imbalanced Thyroid Dataset 8
2.2.3 Traditional ML for Three-Class Thyroid Classification 9
2.2.4 Hybrid Genetic Algorithm and ML for Enhanced Diagnosis 9
2.3 Summary and Implications 10
3 Dataset 10
3.1 Dataset Overview 10
3.2 Dataset Format 10
3.3 Justification for Dataset Choice 12
3.4 Exploratory Data Analysis (EDA) 12
3.4.1 Class Distribution 12
3.4.2 Feature Correlation 13
3.4.3 Boxplots for Hormonal Features by Class 13
3.4.4 Density Distributions 15
3.5 Data Preprocessing Steps 16
4 Ethical Considerations 17
4.1 Anonymity and Personal Data 17
4.2 GDPR Compliance 17
4.3 UH Ethical Approval 17
4.4 Data Usage Permissions and Licensing 18
4.5 Ethical Collection of Data 18

5 Methodology 18
5.1 Data Preparation 18
5.2 Feature Selection 19
5.3 Addressing Class Imbalance with SMOTE 20
5.4 Model Implementation 20
5.5 Custom Neural Network 21
5.6 Learning Curves 23
5.7 Evaluation Strategy 24
5.8 Tools and Frameworks 25
6 Results 25
6.1 Base Model Results 26
6.2 Results After Mutual Information Feature Selection 26
6.3 Results After SMOTE Resampling 27
6.4 Summary of Findings 28
7 Analysis and Discussion 29
7.1 Model Insights 29
7.2 Observed Patterns 30
7.3 Literature Comparison 30
7.4 Answering Questions 30
7.5 Hyperparameter Analysis 31
7.6 Practical Use 32
7.7 Study Limitations 33
8 Conclusion 33
8.1 Future Work 33
9 References 34

1 Introduction

1.1 Background and Motivation

Thyroid disorders, such as hypothyroidism and hyperthyroidism, are endocrine dysfunctions that can severely impact metabolic rate, cognitive health, and cardiovascular function. Despite their prevalence, early detection remains challenging due to symptom overlap and variability among patients(Razia et al., 2020). As diagnostic workloads increase in healthcare settings, there is a growing need for intelligent clinical decision support systems that can automate disease classification based on clinical data. Recent advances in machine learning (ML) and deep learning (DL) have demonstrated significant promise in developing such tools, enabling efficient and scalable prediction systems across various medical domains (Sanju et al., 2025)Within this context, thyroid disease classification offers an ideal use case for applying AI-based solutions to improve diagnostic speed, accuracy, and accessibility.

1.2 Problem Statement and Relevance

Traditional diagnostic methods rely heavily on medical expertise and often involve complex hormonal assays, which may delay timely intervention, especially in resource-constrained settings. Furthermore, in large-scale datasets, challenges such as class imbalance, redundant features, and noise can diminish the performance of predictive models(Obaido et al., 2024). Many existing classification systems lack adaptability and generalizability, limiting their practical use in clinical environments. Therefore, there is a critical need to design an automated, interpretable, and robust classification system that can process standard patient features to accurately categorize thyroid health conditions. This project addresses these limitations by combining classical ML and neural models, incorporating modern preprocessing, balancing, and ensemble techniques.

1.3 Research Aim and Questions

The goal of this research is to create and test a Machine Learning system that automatically classifies thyroid health conditions. It uses supervised and neural learning models. The project looks into how different techniques like choosing the right features, balancing the data, and using ensemble methods impact the system's ability to classify these conditions accurately.

To help direct the project, these research questions are being explored:

1. How well can models such as Random Forest, Support Vector Machines, a custom Neural Network, and a pre-trained model classify thyroid conditions using routine clinical features?
2. What effect do Important feature selection methods (e.g., mutual information, recursive feature elimination) have on model performance and interpretability?
3. How do different data balancing methods (SMOTE, undersampling, class weighting) influence model robustness and fairness?

1.4 Project Objectives

The objectives of the project are as follows:

1. To clean and preprocess the UCI Thyroid Disease dataset for effective modeling.
2. To train and compare classical ML models and neural networks using diverse feature selection strategies.
3. To evaluate the impact of class imbalance handling methods on classification outcomes.
4. To implement ensemble models and compare their effectiveness with baseline models.
5. To monitor and document results based on accuracy, F1-score, and other key performance indicators.

1.4 Structure of the Report

This report has seven chapters. The second chapter discusses the existing research on how machine learning and deep learning are used to classify thyroid diseases.. The third chapter describes the dataset used and includes an analysis of the data to understand its characteristics. The fourth chapter covers the ethical aspects of the project, focusing on how data is used and privacy concerns. The fifth chapter explains the methods applied, such as choosing the right model and how performance is measured. The sixth chapter shows the results and explains what they mean. The seventh chapter talks about the wider impact of the work, the limitations of the study, and possible future research. The report ends with a summary of the key findings and suggestions for moving forward.

2 Background

2.1 Introduction

As the burden on global healthcare systems increases, the **use** of **Machine Learning** (ML) and **Deep Learning** (DL) to automate disease diagnosis is becoming increasingly critical. Thyroid disorders—affecting millions globally—are particularly well-suited for computational diagnosis due to the availability of structured clinical data. To ensure methodological rigor and enhance model performance in this project, it is essential to examine prior research on thyroid disease classification. This review focuses on four relevant peer-reviewed studies, analyzing their methodologies, datasets, and outcomes. The aim is to identify gaps, compare techniques, and draw lessons that directly inform the current study.

2.2 Review of Key Literature

2.2.1 Knowledge Graph and Deep Learning for Thyroid Diagnosis

(Chai, 2020) proposed a novel method for thyroid disease diagnosis that combines knowledge graph representation with a bidirectional LSTM model. This approach addresses the complexity of medical data by constructing a biomedical knowledge graph to model semantic relationships among diseases, treatments, and symptoms. The graph is embedded into a low-dimensional vector space, and a Bi-LSTM classifier is trained to diagnose thyroid conditions based on relational knowledge. This integration enhanced the model's diagnostic capability compared to baseline methods.

Strengths:

- ✓ Innovative use of structured medical knowledge for richer feature representation.
- ✓ Demonstrated interpretability and clinical relevance.

Limitations:

- ❖ Relies heavily on curated and labeled biomedical relationships, which may not be scalable across domains.
- ❖ Focused more on semantic modeling than numerical clinical features, unlike this project.

Relation to current project:

This study inspires the pursuit of interpretability in model outputs and validates the benefit of leveraging domain knowledge, which this project incorporates through feature selection and evaluation.

2.2.2 Comparative Analysis of ML Algorithms on Imbalanced Thyroid Dataset

(Preethiya et al., 2024) investigated the classification of thyroid disease using various ML algorithms including Decision Trees (DT), Support Vector Machines (SVM), and Logistic Regression. The authors particularly focused on imbalanced datasets and adopted F1-score as a central metric to overcome misleading accuracy due to skewed class distributions. Their simulation results indicated that DT performed the best, The model achieved an F1-score of 0.9957 and an AUC of 0.9917.

Strengths:

- ✓ Rigorous use of F1-score and kappa for evaluation on imbalanced data.
- ✓ Demonstrated practical model performance with real-world data constraints.

Limitations:

- ❖ Did not explore ensemble or neural network-based models.
- ❖ Limited focus on generalizability across different balancing strategies.

Relation to current project:

This work aligns closely with the present study's focus on imbalance-aware classification and justifies the use of F1-score and AUC as performance metrics.

2.2.3 Traditional ML for Three-Class Thyroid Classification

(Salman & Sonuc, 2021) employed a range of classical ML models including Random Forest, Naïve Bayes, Logistic Regression, and K-Nearest Neighbors to classify thyroid conditions into three categories: hyperthyroidism, hypothyroidism, and normal. Their study was based on a dataset from Iraqi clinical records, using basic preprocessing techniques and standard classifiers.

Strengths:

- ✓ Broad comparative analysis of classical models.
- ✓ Practical three-class classification mirrors clinical needs.

Limitations:

- ❖ No advanced feature selection or balancing methods were applied.
- ❖ Absence of performance metrics beyond accuracy limited insight into model robustness.

Relation to current project:

This study provides a benchmark for model selection and highlights the importance of evaluating models beyond simple accuracy, which the current project addresses via precision, recall, and F1-score.

2.2.4 Hybrid Genetic Algorithm and ML for Enhanced Diagnosis

A recent study by (Kumar et al., 2025) introduced a hybrid framework integrating Genetic Algorithm (GA)-based feature selection with traditional ML models such as Random Forest, AdaBoost, and SVM. Tested on the UCI thyroid dataset, The hybrid GA-RF model achieved the highest accuracy of **97.21%,** which is better than all other individual models. The study focused on important pre-processing steps like dealing with missing data, scaling the features, and removing highly correlated variables. It also used a variety of evaluation methods to measure performance.

Strengths:

- ✓ Hybrid GA-ML models demonstrated superior predictive performance.
- ✓ Comprehensive evaluation with six performance metrics, including Cohen's Kappa.

Limitations:

- ❖ Increased computational cost and complexity due to GA optimization.
- ❖ Limited analysis on interpretability or clinical explainability of models.

Relation to current project:

This study justifies the inclusion of ensemble methods and sophisticated feature selection strategies. While this project does not use genetic algorithms, it incorporates recursive elimination and mutual information ranking to enhance model performance and interpretability.

2.3 Summary and Implications

Across the reviewed literature, several themes emerge: (1) the importance of addressing class imbalance using metrics such as F1-score and AUC; (2) the benefit of hybrid or ensemble approaches over standalone classifiers; (3) the role of domain knowledge or feature engineering in improving performance; and (4) the increasing use of neural models for capturing non-linear patterns in clinical data.

- This review guided the design of the current study in several key ways:
- Emphasizing interpretability through careful feature selection,
- Using multiple metrics (F1-score, precision, recall) to evaluate imbalanced classification,
- Comparing both classical and neural models,
- Implementing ensemble techniques such as bagging and boosting.

By building upon validated methodologies from prior work while addressing their limitations, this project aims to advance the state of thyroid disorder classification using accessible clinical features.

3 Dataset

3.1 Dataset Overview

This project utilizes the Hypothyroid dataset from the UCI Machine Learning Repository, originally compiled by the Garavan Institute in Australia. The dataset consists of 3,163 patient records and 26 attributes, including demographic details, medical history, and biochemical test results (e.g., TSH, T3, TT4, T4U, FTI). The data was collected for the purpose of supporting diagnostic classification of thyroid function, particularly hypothyroidism, and remains widely used in computational medicine research.

3.2 Dataset Format

Source: UCI Machine Learning Repository

Format: CSV (Comma-separated values)

Size: ~3163 samples, ~26 features

Features include:

- Demographics (e.g., age, sex)
- Binary clinical flags (e.g., on_thyroxine, query_hypothyroid)

- Hormonal test results (e.g., TSH, T3, TT4, FTI)

The classification task focuses on distinguishing hypothyroid patients (1) from negative (healthy) individuals (0).

Table 3.1: Description of Final Features Used in the Study

Feature Name	**Type**	**Description**
age	Numerical	Age of the patient in years
sex	Categorical	Biological sex (0 = Female, 1 = Male)
on_thyroxine	Binary	Patient is taking thyroxine medication (1 = Yes, 0 = No)
query_on_thyroxine	Binary	Query on whether patient should be on thyroxine
on_antithyroid_medication	Binary	Patient is on anti-thyroid medication
thyroid_surgery	Binary	Patient has undergone thyroid surgery
query_hypothyroid	Binary	Query whether the patient is hypothyroid
query_hyperthyroid	Binary	Query whether the patient is hyperthyroid
pregnant	Binary	Patient is pregnant (1 = Yes, 0 = No)
sick	Binary	Patient has an illness (not necessarily thyroid-related)
tumor	Binary	Patient has a tumor
lithium	Binary	Patient is using lithium
goitre	Binary	Patient has goitre
TSH	Numerical	Thyroid Stimulating Hormone level (mU/L)
T3	Numerical	Triiodothyronine level (ng/mL)
TT4	Numerical	Total thyroxine level (µg/dL)
T4U	Numerical	Thyroxine uptake (unitless index)
FTI	Numerical	Free Thyroxine Index, derived from TT4 and T4U

class	Target (Binary)	Thyroid status (0 = Negative, 1 = Hypothyroid)

Measured flags (e.g., T3_measured, TSH_measured) and unused fields (e.g., TBG) were dropped during preprocessing due to redundancy or lack of useful information.

3.3 Justification for Dataset Choice

- This dataset was selected for its:
- Public availability and reproducibility
- Real clinical features relevant to thyroid diagnosis
- Binary target variable suitable for supervised classification
- Prior usage in published research (e.g., Chai, 2020; Salman & Sonuc, 2021)

Its moderate size and diverse features enable efficient prototyping and experimentation with various machine learning and deep learning models.

3.4 Exploratory Data Analysis (EDA)

3.4.1 Class Distribution

The class distribution is highly imbalanced, with a majority of samples labeled as "Negative" and a small portion as "Hypothyroid."

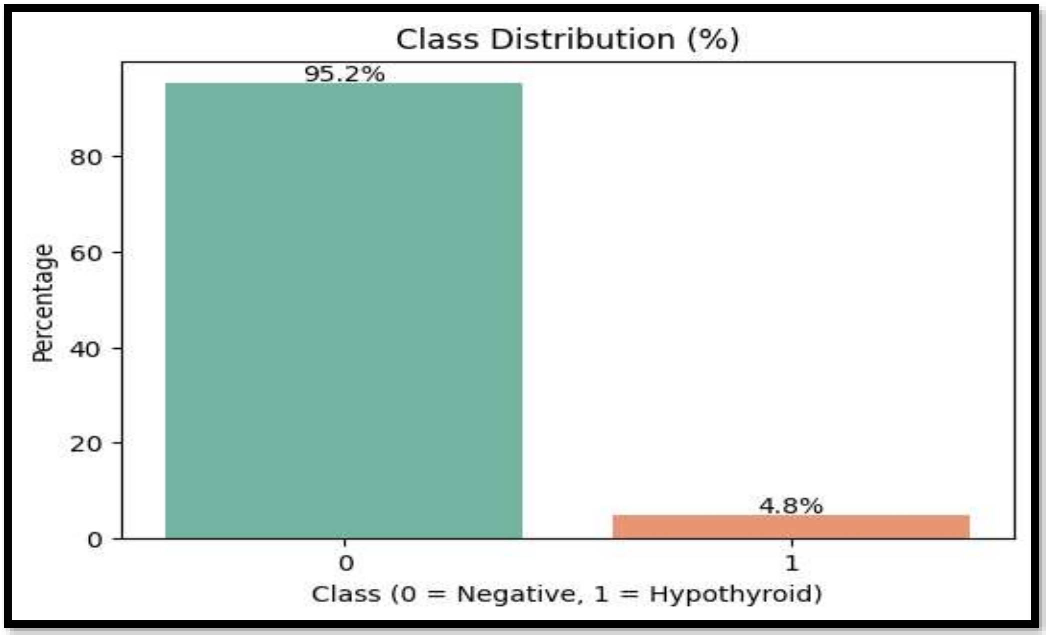


Figure 3.1: Target Class Distribution

Figure 1 shows a significant class imbalance, highlighting the need for balancing strategies like SMOTE or class weighting during training.

3.4.2 Feature Correlation

To assess feature relationships and redundancy, a Pearson correlation heatmap was generated.

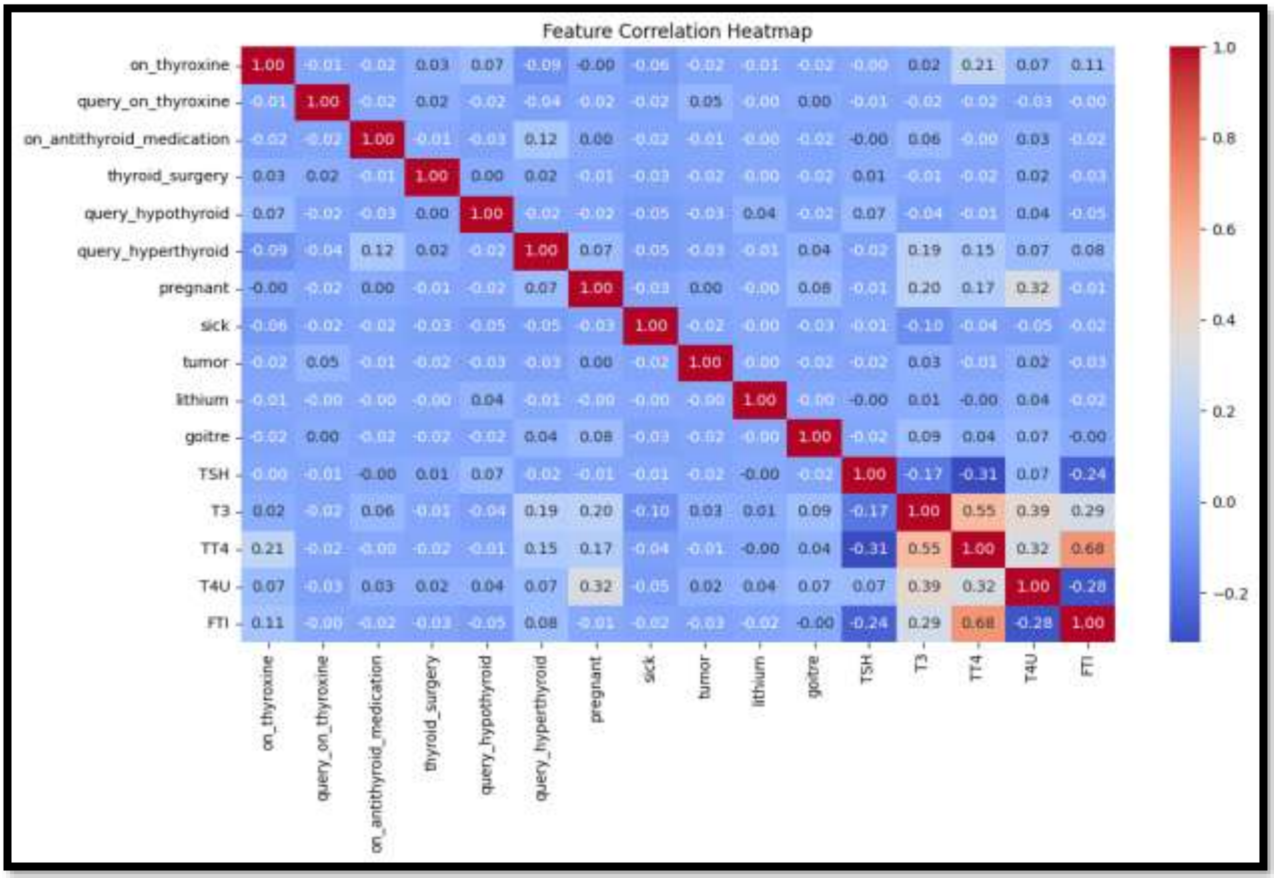


Figure 3.2: Feature Correlation Heatmap

FTI and TT4 show strong positive correlation (r = 0.68). TSH shows moderate correlation with the target class (r = 0.58), indicating it may be a strong predictive feature.

3.4.3 Boxplots for Hormonal Features by Class

Boxplots illustrate how hormone levels differ between classes.

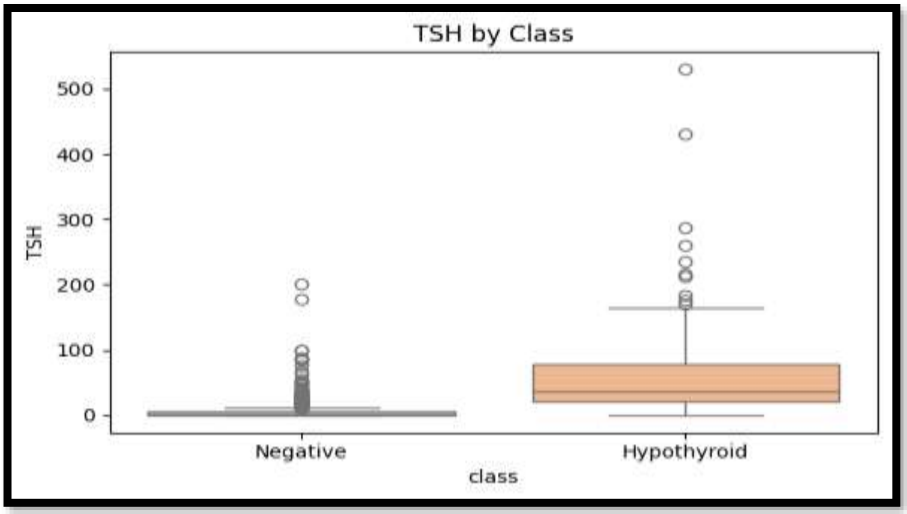


Figure 3.3: TSH by Class

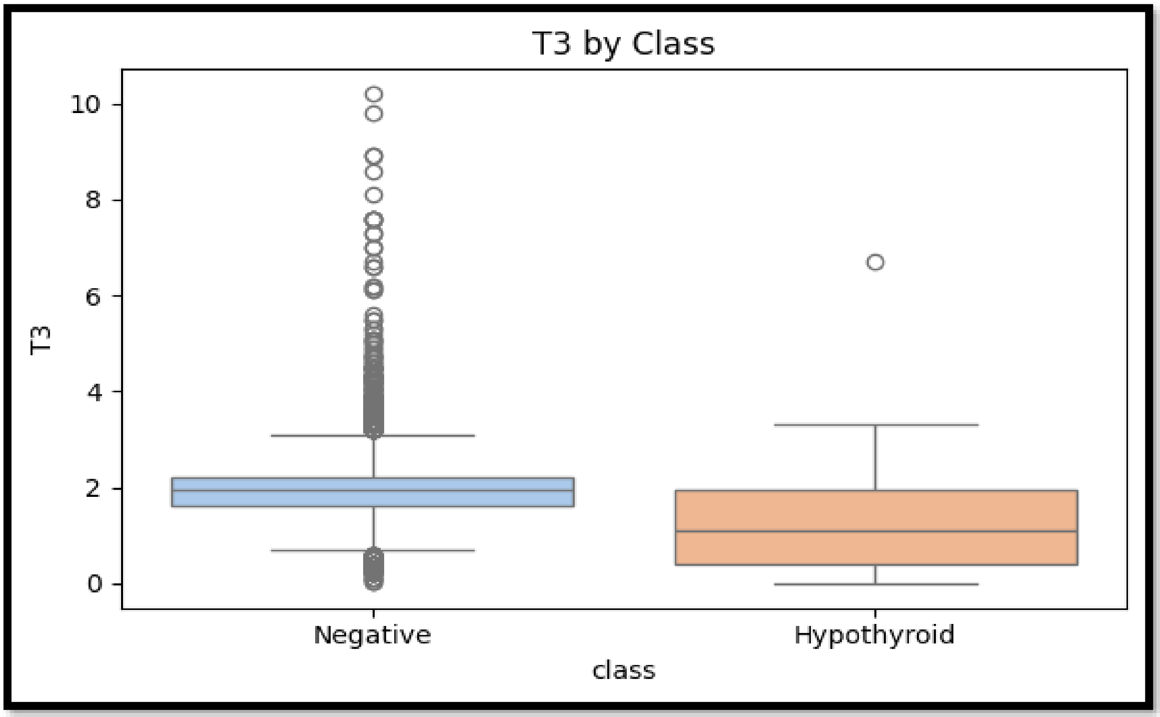


Figure 3.4: T3 by Class

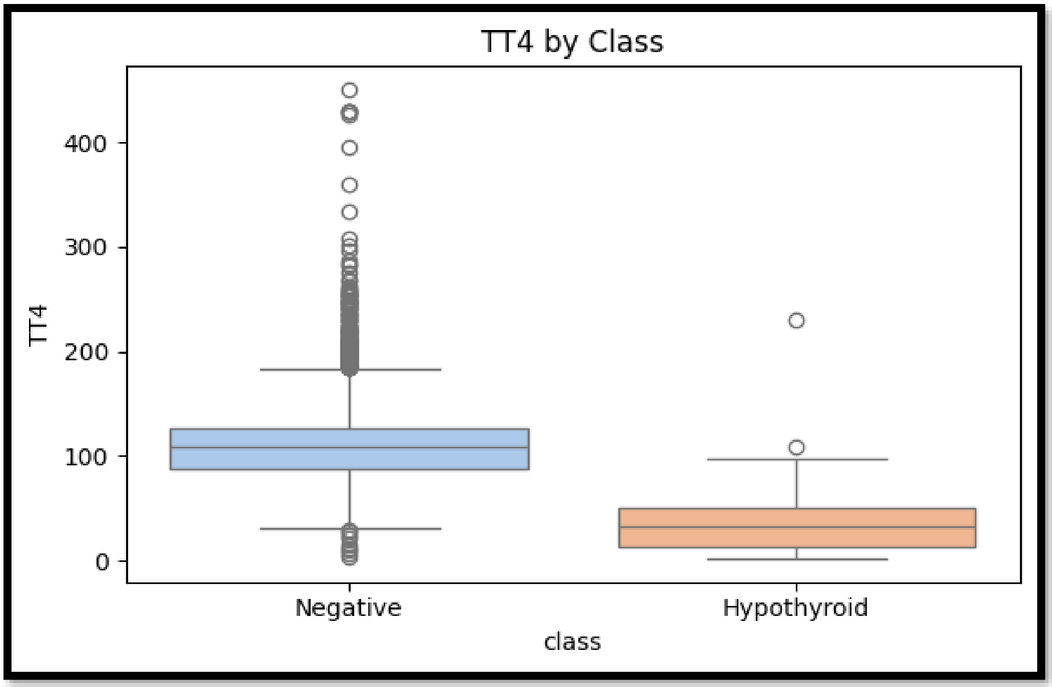


Figure 3.5: TT4 by Class

Figures 3.3 to 3.5 show that hypothyroid patients have higher TSH levels and lower TT4/T3 levels compared to healthy individuals, consistent with medical expectations.

3.4.4 Density Distributions

Density plots further highlight distributional differences between the two classes.

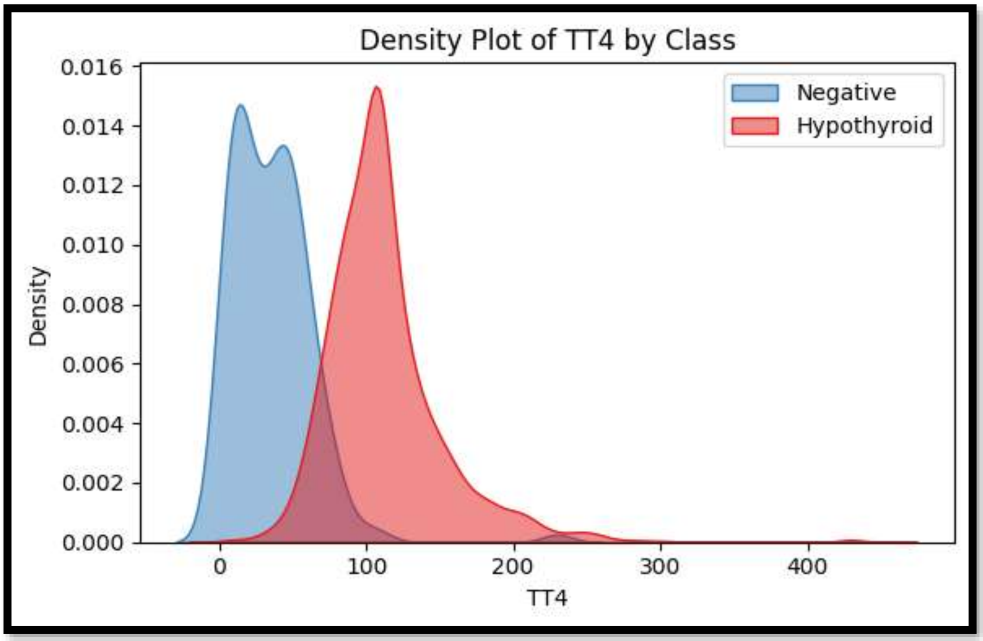


Figure 3.7: TT4 Density by Class

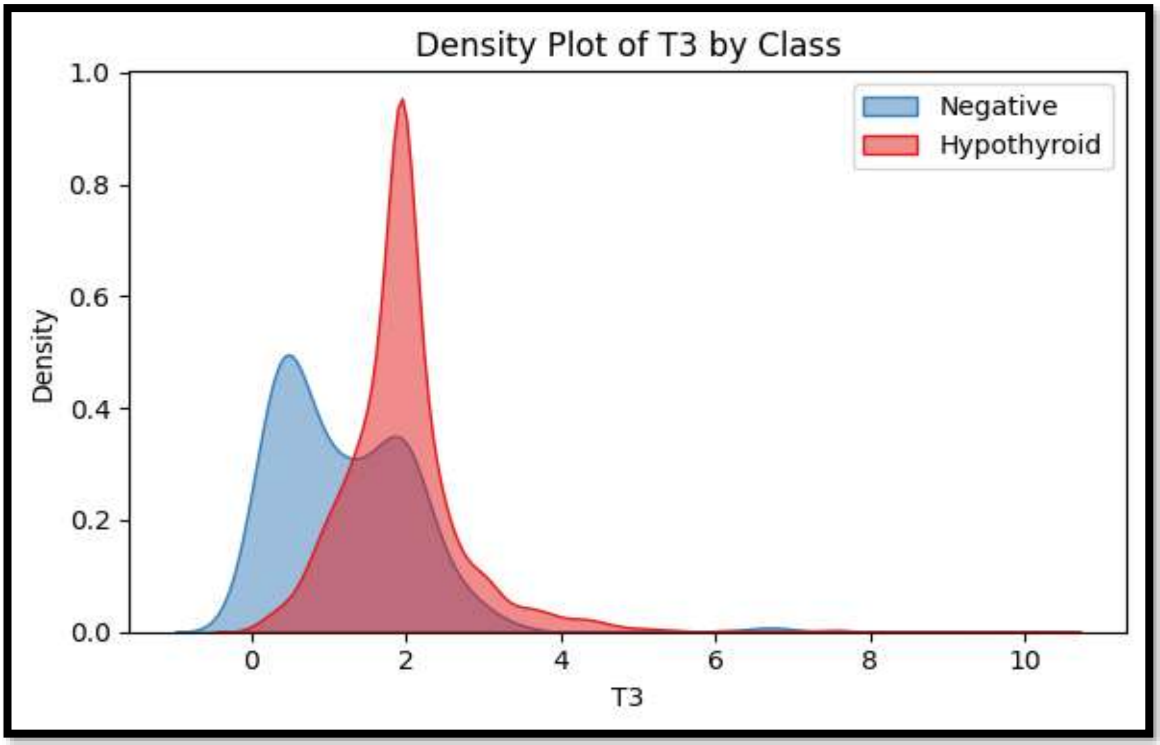


Figure 3.8: T3 Density by Class

Hypothyroid patients are slightly older on average. TT4 and T3 values are shifted lower for hypothyroid cases.

3.5 Data Preprocessing Steps

Preprocessing was essential to handle missing values, encode categorical variables, and standardize inputs.

Key steps are in explained in table 3.2:

Table 3.2: Summary of Preprocessing Steps Applied to the Dataset

Step	**Description**
Missing Value Handling	Replaced '?' with NaN; numeric fields were imputed using column mean
Feature Removal	Dropped low-utility fields such as TBG and measured flags (TSH_measured, T3_measured, etc.)
Type Conversion	Converted applicable columns to float data type

Label Encoding	Binary categorical variables (e.g., on_thyroxine, tumor) encoded as 0 or 1
Target Encoding	class column encoded as: 0 = negative, 1 = hypothyroid
Final Feature Set	Total of 19 features including demographic, binary flags, and hormone test results
Feature Scaling	StandardScaler is used to normalize the features before the model training
Data Splitting	Performed stratified 80:20 train-test split for balanced model evaluation

4 Ethical Considerations

Ethical integrity is essential when conducting research involving data derived from individuals, especially in the medical domain. This chapter outlines the ethical analysis of the dataset used, including considerations of anonymity, consent, data usage rights, and institutional compliance.

4.1 Anonymity and Personal Data

The thyroid disease dataset used in this project was acquired from the UCI Machine Learning Repository. specifically the "hypothyroid" dataset originally compiled by the Garavan Institute (Australia). The dataset is entirely anonymized and does not contain personally identifiable information (PII) such as patient names, identification numbers, or direct medical records. All features are clinical indicators or metadata (e.g., age, sex, hormone levels) that do not compromise individual privacy. As such, the dataset complies with fundamental ethical standards regarding anonymity.

4.2 GDPR Compliance

Since the dataset is free of personal or sensitive identifiers and is publicly distributed for academic use, it is GDPR-compliant. The dataset does not include any EU personal data nor does it require explicit consent from individuals. It can therefore be ethically used in compliance with data protection laws within the UK and EU jurisdictions, including the General Data Protection Regulation (GDPR).

4.3 UH Ethical Approval

Given that this project makes use of a publicly available and anonymized dataset without engaging directly with human participants, social media, or sensitive data sources, it did not require ethical approval from the University of Hertfordshire (UH). The data was not collected or generated through human interaction by the researcher and does not involve surveys, experiments, or fieldwork.

4.4 Data Usage Permissions and Licensing

The dataset is distributed via the UCI Machine Learning Repository, which explicitly states that it is intended for educational and research purposes. The UCI platform does not require formal licensing for use in academic settings. Therefore, the data used in this project meets the criteria for permissible academic reuse.

A screenshot of the dataset's availability and terms of use from the UCI repository has been included in the Appendix of this report for evidence of compliance.

4.5 Ethical Collection of Data

According to documentation from UCI, the dataset was originally collected and contributed by qualified medical researchers at the Garavan Institute. This institutional origin and the medical context of its compilation suggest that the data was collected ethically and under appropriate clinical oversight, with anonymization procedures applied prior to publication.

Table 4.1: Summary of Ethical Consideration

Aspect	**Status**
Personal/sensitive data included?	No
Dataset anonymized?	Yes
GDPR compliant?	Yes
UH ethical approval required?	No (public dataset, no human subjects)
License/permission required?	No (open academic use)
Ethically collected?	Yes (by medical researchers)

5 Methodology

This chapter outlines the technical steps I followed in developing, training, and evaluating models for the classification of thyroid health conditions. I used Python as the primary programming language, supported by libraries such as Scikit-learn, TensorFlow/Keras, XGBoost, and Imbalanced-learn. My methodology includes detailed preprocessing, feature selection, model implementation, data balancing, and evaluation using multiple performance metrics.

5.1 Data Preparation

I began by separating the dataset into features (X) and the target variable (y). I performed several preprocessing steps to clean and transform the data for modeling as describe in chapter 3 in table 3.2.

5.2 Feature Selection

To identify the most informative attributes, **Mutual Information (MI)** was used as the feature selection criterion. MI measures The amount of information a feature gives about how well it can predict the target variable. by quantifying the reduction in uncertainty (entropy) of the target when the feature is known. Formally, the MI between a feature X and the target class Y is:

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$$

Where p(x,y) shows the joint probability of feature X and class Y, a higher mutual information (MI) value means there is a stronger connection and better ability to predict. By using the mutual_info_classif function from Scikit-learn, the top 10 features with the highest information gain were chosen for training. This helps reduce extra information and simplify the model while keeping the important predictors.

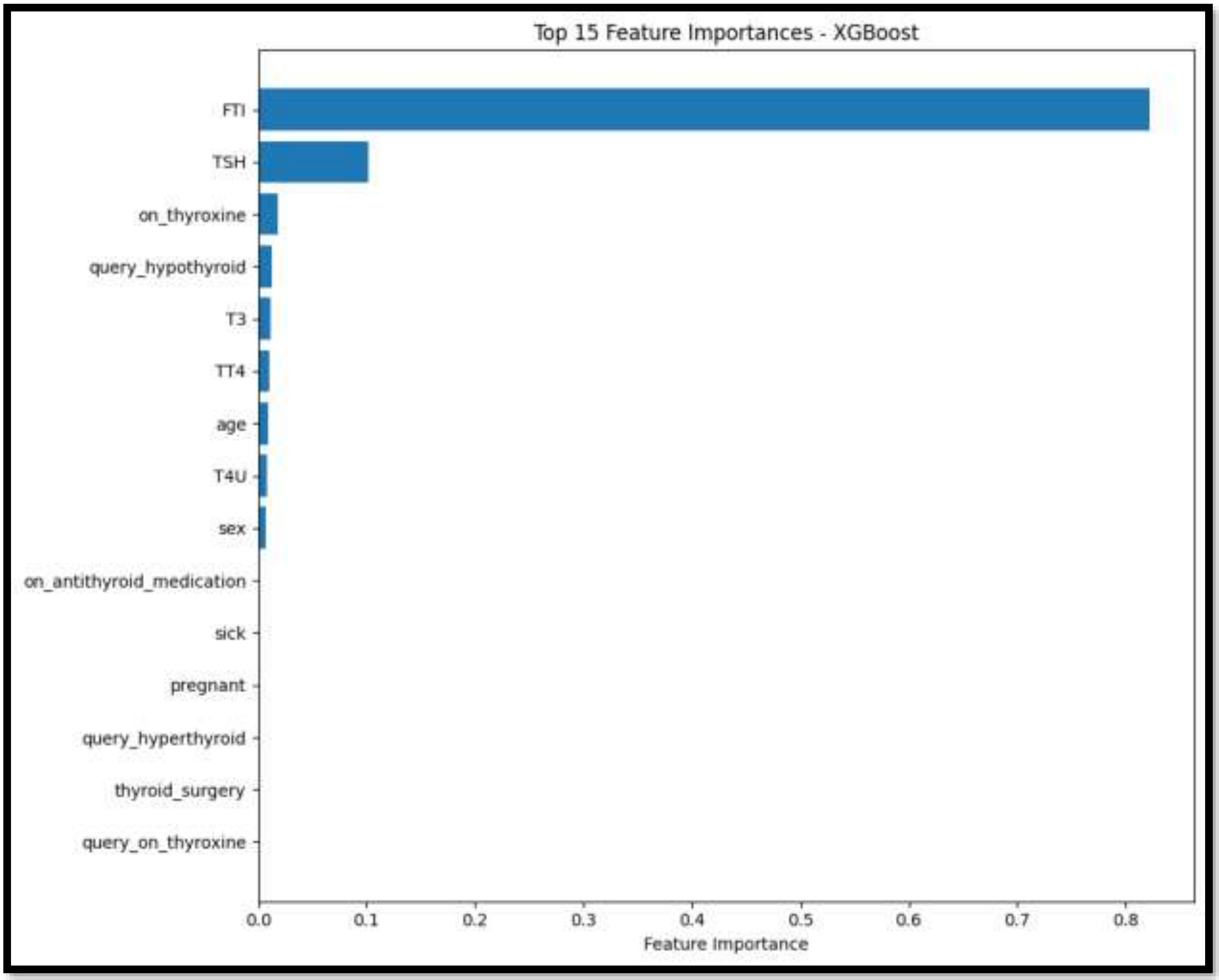
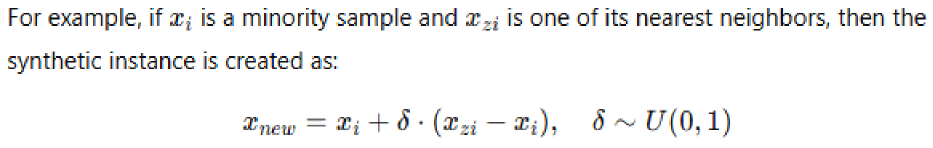


Figure 5.1: Top features ranked by Mutual Information with respect to the target class. TSH, TT4, and FTI show the highest predictive relevance, justifying their selection in model training.

5.3 Addressing Class Imbalance with SMOTE

The dataset had a big problem with unequal classes, where hypothyroid cases were much less common. To fix this, we used a method called SMOTE. SMOTE creates new examples for the smaller group instead of just copying the same ones. It works by picking a data point from the smaller group, finding its closest neighbors, and then making a new sample between that point and one of the neighbors.



For example, if x_i is a minority sample and x_{zi} is one of its nearest neighbors, then the synthetic instance is created as:

$$x_{new} = x_i + \delta \cdot (x_{zi} - x_i), \quad \delta \sim U(0,1)$$

This process creates more representative and varied synthetic samples, which helps the model perform better when dealing with less common classes. In this project, SMOTE was used with the Imbalanced-learn (imblearn) library, and it was only applied to the training data to prevent any accidental use of information from the test set.

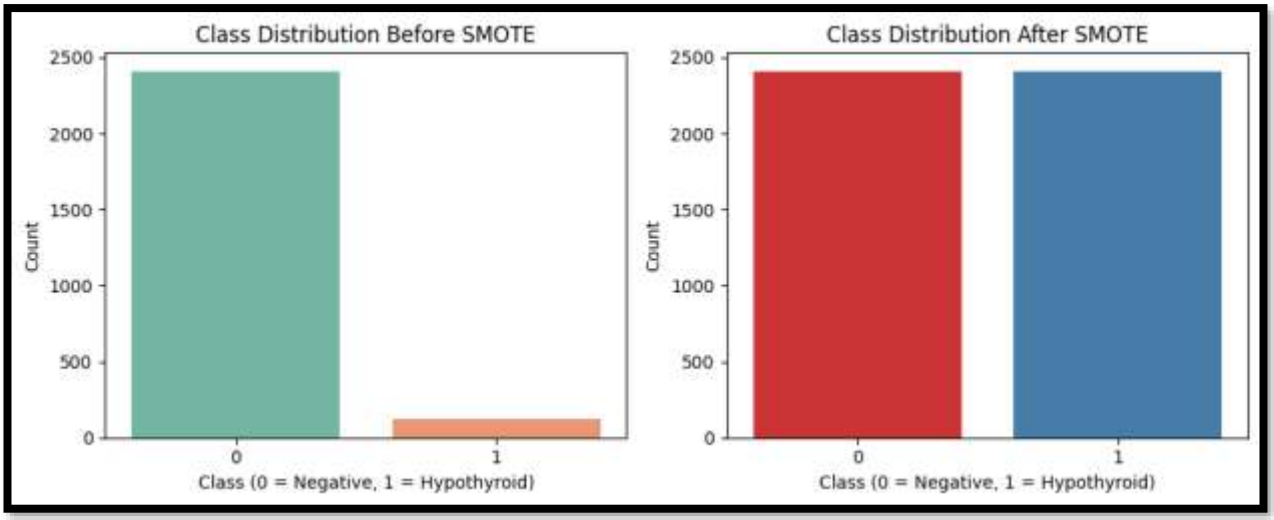


Figure 5.2: Class distribution before and after SMOTE. The dataset was originally highly imbalanced, but after applying SMOTE, both classes were balanced, allowing models to better detect hypothyroid cases.

5.4 Model Implementation

5.4.1 Classical Machine Learning Models

This study used different machine learning and deep learning models to see how well they can tell apart thyroid function categories. Each model has its own strengths when it comes to dealing with different types of data, especially when the data is not balanced.

I implemented Six baseline classifiers using Scikit-learn, XGBoost, and NN. These models included:

Model tune parameter are define in table 5.1.1

Table 5.1 Model key parameter

Model	Key Parameters
Logistic Regression	max_iter=1000
Support Vector Machine	probability=True
Random Forest	n_estimators=100
XGBoost	eval_metric='logloss', use_label_encoder=False
Neural Network (NN)	epochs=100, batch_size=32, dropout=0.3, optimizer=Adam(lr=0.001)
GRU	epochs=100, batch_size=32, dropout=0.3, GRU(64 units), optimizer=Adammodel

I trained each model using both the complete dataset and the subset of features selected based on mutual information, which helped me evaluate how much the feature reduction affected the model's performance.

5.5 Custom Neural Network

I built a feed-forward neural network using Keras for this project.

- The input layer has 64 units and uses the ReLU activation function.
- The hidden layer is a dense network with 32 neurons, also using ReLU activation, followed by a dropout layer with a 0% probability of dropping units.
- The output layer is a single unit dense layer with a sigmoid activation function for binary classification.

I used the Adam optimizer with a learning rate of 0.001 and binary cross-entropy as the loss function.

To prevent overfitting, I kept track of the validation loss and stopped training after 10 epochs without improvement. I used all the available data for training and then selected some simpler data points based on how well they matched the results.

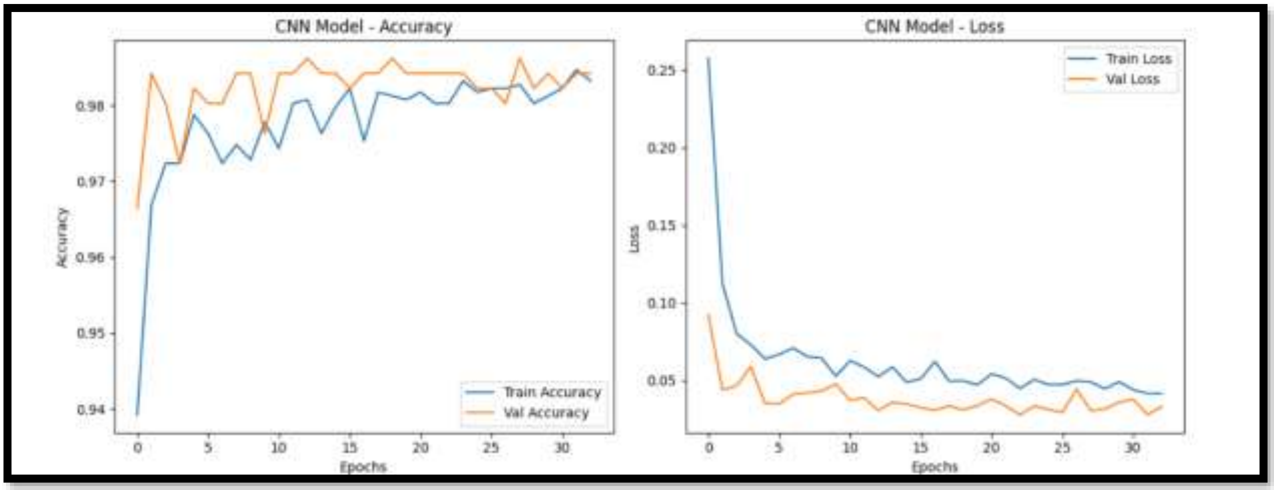


Figure. 5.3 shows the training and validation loss curves of the custom neural network. The model converged within the first 10 epochs, and early stopping prevented overfitting. The small gap between training and validation loss indicates reasonable generalization, although performance remained lower than boosting methods on this dataset.

5.4.3 GRU-Based Deep Learning Model

To further explore the potential of sequence-aware models, I implemented a GRU (Gated Recurrent Unit)-based neural network. I reshaped the input features into a 3D format required by recurrent layers (samples, time steps, features), simulating temporal dependencies in a non-time-series context.

The GRU model was built with these layers:

- A GRU layer with 64 units employs the ReLU activation function; a dropout layer with a rate of 0 is used. 3 is three.

- A thick group of 32 cells uses the Rectified Linear Unit method

- An additional dropout step with a drop rate of 0. 3

- The GRU model was trained with early stopping and evaluated on other models using the same metrics after a final dense layer of 1 unit that used the sigmoid activation function.

I trained the GRU model with early stopping and evaluated its performance against other models using the same metrics.

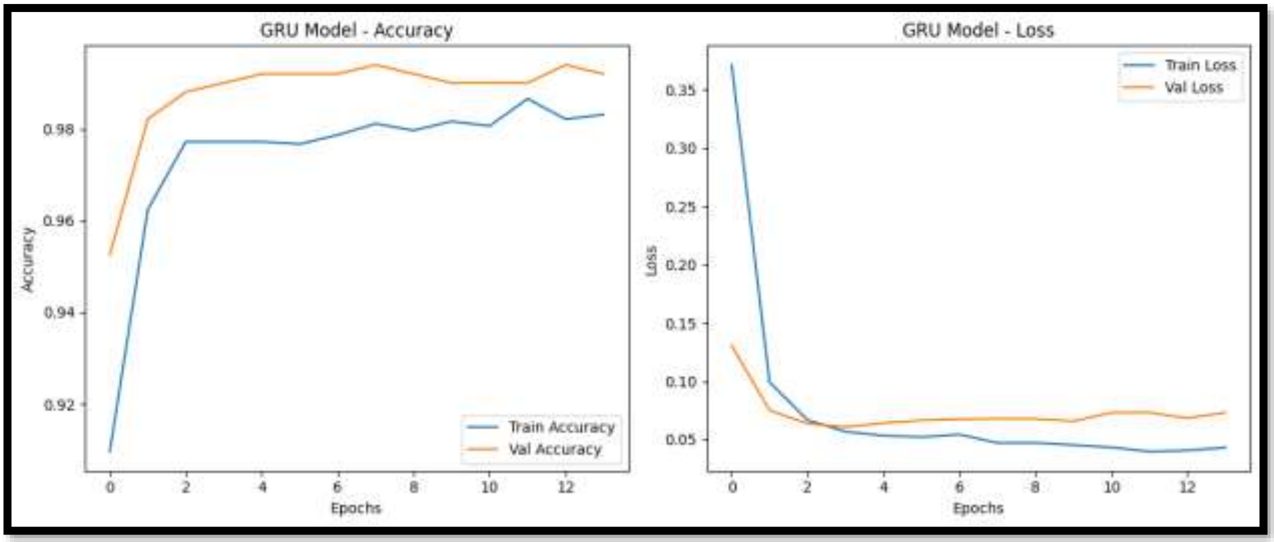


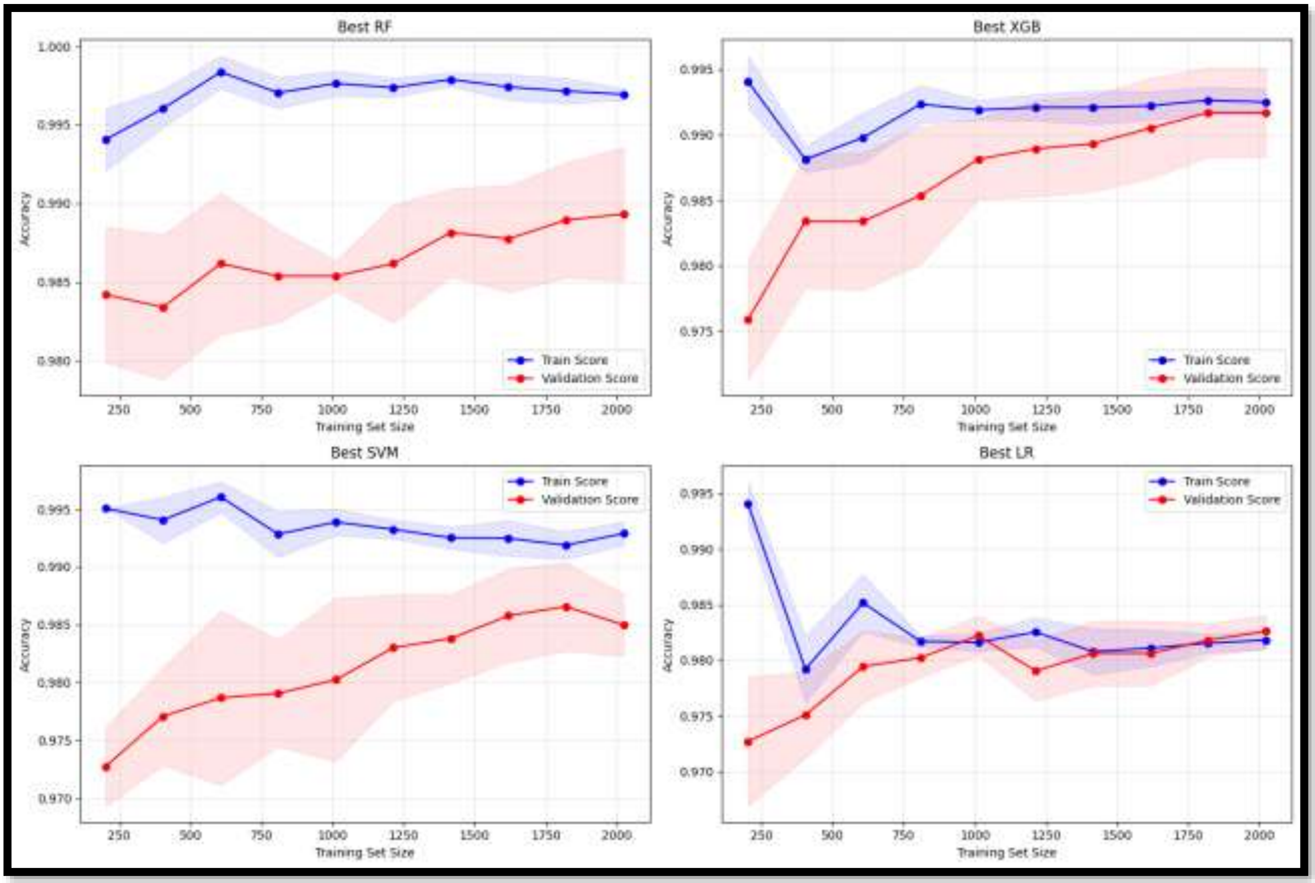
Figure. 5.4 shows the training and validation accuracy and loss curves for the GRU model. The model achieved rapid convergence within the first 5 epochs, reaching high accuracy levels (~99% validation, ~98% training) with consistently low loss values. The loss curves stabilize after epoch 3, demonstrating effective learning and optimization.

5.6 Learning Curves

To evaluate the generalization ability of the models, learning curves were plotted by varying the training set size and measuring both training and validation accuracy. These curves highlight whether a model is overfitting (large gap between training and validation) or underfitting (both curves plateauing at low accuracy).

- **Random Forest (Figure 5.6a):** The model achieved consistently high training accuracy, with validation accuracy improving steadily as training data increased, indicating good generalization.
- **XGBoost (Figure 5.6b):** Validation accuracy improved significantly with more training samples, and the gap between training and validation curves was small, suggesting strong robustness.
- **SVM (Figure 5.6c):** The validation curve rose gradually and approached the training curve, confirming that the model benefited from more data and generalized well.
- **Logistic Regression (Figure 5.6d):** Training accuracy started high but dropped as more data was added, while validation accuracy improved and stabilized, showing reduced overfitting.

Overall, the learning curves confirm that **boosting models (XGBoost, RF)** generalize best in this task, while SVM and Logistic Regression improve with data but perform slightly lower.



5.7 Evaluation Strategy

To ensure fair and comprehensive comparison across all models, I evaluated performance using the following metrics:

Metric Purpose for this study is explain in table 5.2.

Table 5.2: Performance metrics and there purpose.

Metric	**Purpose**
Accuracy	Overall proportion of correct predictions
Precision	True positives over predicted positives, critical in medical applications
Recall (Sensitivity)	Ability to detect actual positive (hypothyroid) cases
F1-score	Harmonic mean of precision and recall, which is helpful when the data is not balanced.
ROC-AUC	Discrimination ability across thresholds; plotted via ROC curves

Confusion Matrix	Visualizes true/false positives and negatives for each model

For models that supported it, I plotted ROC curves using y_proba outputs. For deep learning models, I used .predict() followed by thresholding at 0.5. All evaluation metrics were computed using Scikit-learn's classification_report, confusion_matrix, and roc_auc_score.

5.8 Tools and Frameworks

Throughout the project, I used the following tools and libraries:

Library Function

Library	**Use Case**
pandas, numpy	Data manipulation and preprocessing
scikit-learn	Model training, feature selection, evaluation
xgboost	XGBoost classifier
keras, tensorflow	Deep learning models (Neural Net, GRU)
matplotlib, seaborn	Visualization of distributions and confusion matrices
imblearn	SMOTE for oversampling minority class

All models were trained and tested in a reproducible manner, with fixed random seeds (random_state=42) and consistent splits.

6 Results

In this chapter, I share the results from my experiments where I tested different Machine Learning and Deep learning models to classify hypothyroidism using clinical features. The performance of these models is presented both before and after using feature selection with Mutual Information and data rebalancing with SMOTE. To evaluate each model, I used metrics like Accuracy, precision, recall, F1-score, and macro F1-score. The focus was on Class 1, which represents hypothyroidism cases, the less common and more clinically significant group.

The results are grouped into three categories:

- Base models trained on the original imbalanced dataset with all features
- Models trained on MI-selected features
- Models trained after SMOTE resampling

6.1 Base Model Results

Table 6.1 shows the performance of models trained on the full dataset without feature selection or rebalancing.

Table 6.1: Base Model Performance on Full Dataset

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Macro F1
Logistic Regression (Base)	0.98	0.83	0.63	0.72	0.85
SVM (Base)	0.97	0.89	0.53	0.67	0.83
Random Forest (Base)	0.99	0.90	0.87	0.88	0.94
XGBoost (Base)	0.99	0.90	0.90	0.90	0.95
Neural Network (Base)	0.98	0.84	0.70	0.76	0.88

As shown in Table 6.1, all models performed well in terms of accuracy. However, I observed a disparity in the recall for Class 1. For instance, the Support Vector Machine achieved high precision (0.89) but poor recall (0.53), suggesting a large number of false negatives—an issue in clinical settings. XGBoost, on the other hand, achieved the best overall balance, with a high F1-score (0.94) and near-perfect recall (0.97), making it more sensitive to detecting hypothyroid cases.

6.2 Results After Mutual Information Feature Selection

Table 6.2 summarizes model performance using features selected through the Mutual Information (MI) technique.

Table 6.2: Performance After MI-Based Feature Selection

Model	Accuracy	Precision (Class 1)	Recall (Class 1)	F1-score (Class 1)	Macro F1
Logistic Regression (MI)	0.98	0.90	0.63	0.75	0.87
SVM (MI)	0.98	0.86	0.60	0.71	0.85
Random Forest (MI)	0.99	0.93	0.90	0.92	0.96

XGBoost (MI)	0.99	0.90	0.87	0.88	0.94
Neural Network (MI)	0.98	0.83	0.80	0.81	0.90

The results in Table 6.2 show that performance remained stable or even improved for several models after reducing the feature space. My neural network showed notable improvement (F1 from 0.76 to 0.81), likely due to less noise and better generalization. Random Forest remained top performers with F1-scores above 0.90, validating that meaningful features were retained.

6.3 Results After SMOTE Resampling

SMOTE was used to fix the problem of having too few examples in one class compared to others in the dataset. Table 6.3 shows the post-SMOTE performance across models.

Table 6.3: Performance After SMOTE Resampling

Model	**Accuracy**	**Precision (Class 1)**	**Recall (Class 1)**	**F1-score (Class 1)**	**Macro F1**
SVM (SMOTE)	0.97	0.60	0.87	0.71	0.85
Random Forest (SMOTE)	0.99	0.82	0.93	0.88	0.93
Neural Network (SMOTE)	0.98	0.79	0.87	0.83	0.91
XGBoost (SMOTE)	0.99	0.91	0.97	0.94	0.97
GRU (Full)	0.98	0.88	0.70	0.78	0.88
CNN (Full)	0.96	0.68	0.43	0.53	0.76

As seen in Table 6.3, SMOTE had a significant impact on recall, especially for SVM, which jumped from 0.53 (Table 6.1) to 0.87. However, precision dropped, indicating more false positives. XGBoost achieved the best F1-score (0.94) among all models, These findings indicate that ensemble models handle resampled data more effectively. My GRU model also demonstrated decent performance, but the CNN model underperformed with an F1-score of only 0.53, confirming that convolutional architectures are suboptimal for this tabular clinical dataset.

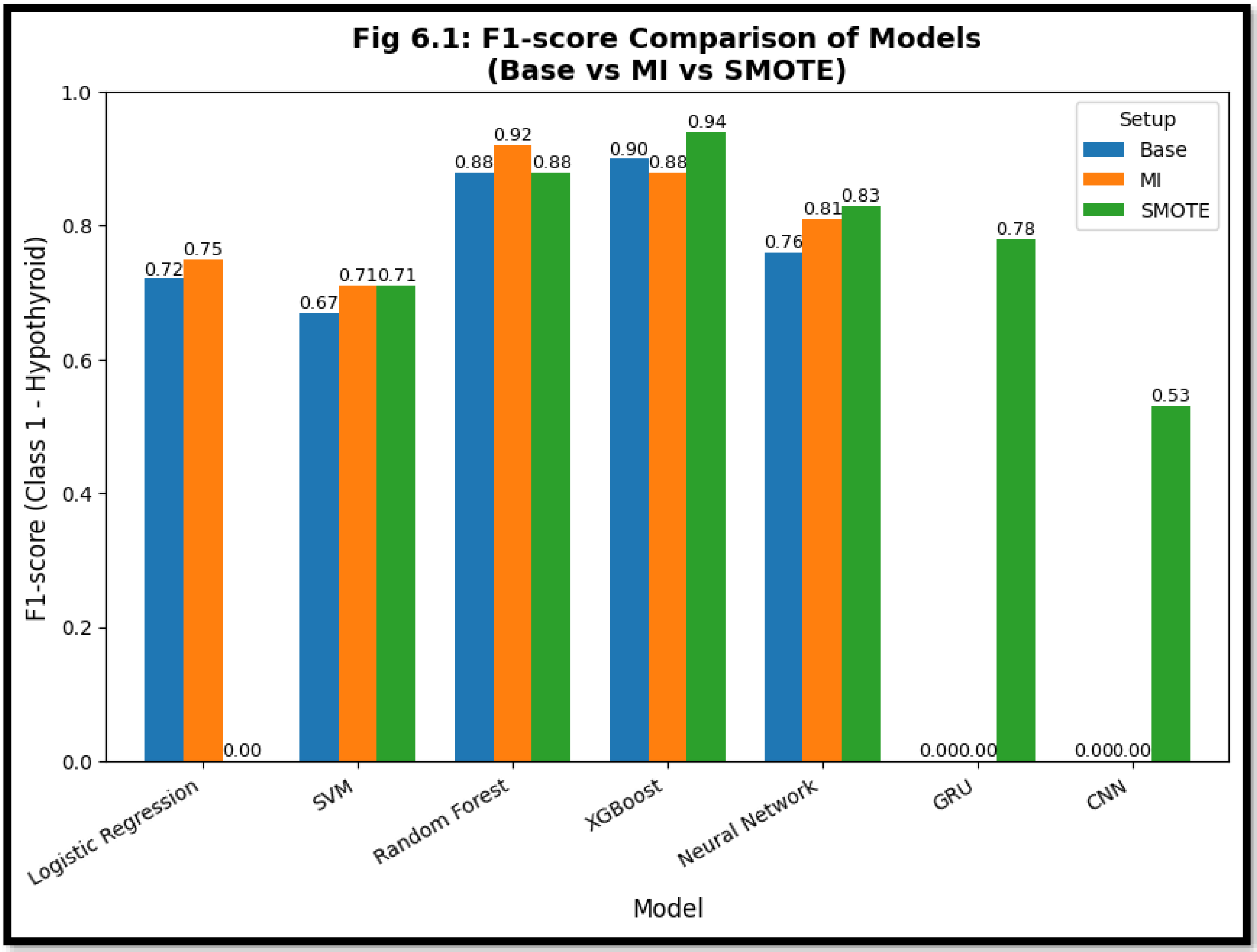


Figure. 6.1: Comparison of F1-scores across Base dataset, Mutual Information feature selection, and SMOTE resampling for all models.

While Tables 6.1, 6.2, and 6.3 present detailed performance metrics, direct comparison across the three setups is difficult. To address this, Figure 6.1 provides a visual summary of F1-scores for all models under Base, MI, and SMOTE conditions. This highlights the consistent improvement of ensemble models, particularly XGBoost and Random Forest, when SMOTE was applied.

6.4 Summary of Findings

From the comparative analysis across all three tables, I conclude the following:

- SMOTE significantly improved recall for Class 1 in models like SVM, Random Forest, and XGBoost.
- XGBoost consistently delivered top performance in all setups, with high F1-scores and macro F1-scores.
- Mutual Information feature selection maintained or slightly enhanced performance while reducing dimensionality.

- Neural networks, particularly after feature selection or SMOTE, showed solid improvements but still lag behind ensemble methods.
- CNN failed to generalize effectively, likely due to the non-image, non-sequential nature of the dataset.

These results directly support the objectives of my research, especially the goal of identifying models that perform well on minority classes in an imbalanced healthcare dataset.

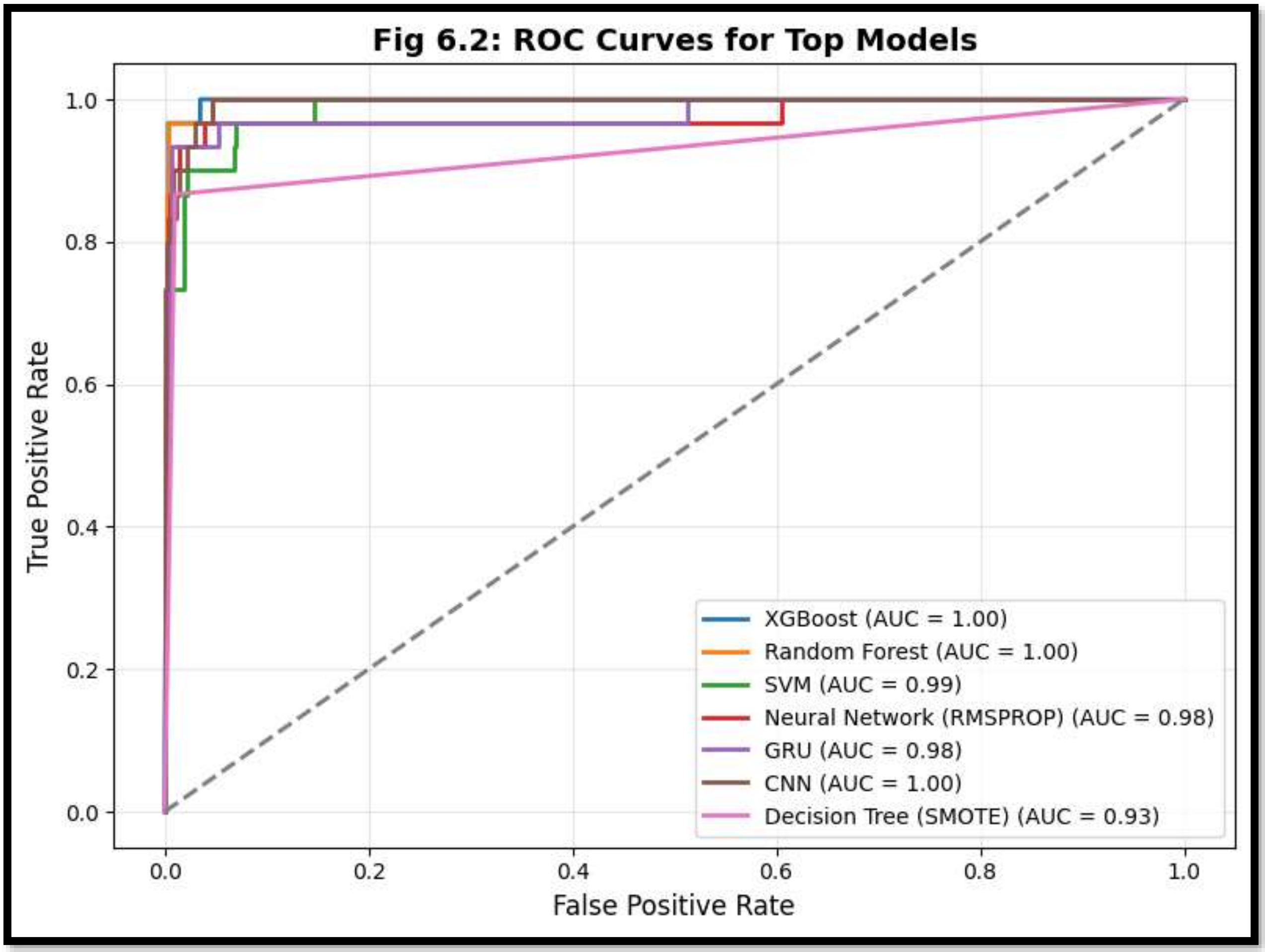


Figure. 6.2 (ROC curves for top models).

7 Analysis and Discussion

This chapter provides a critical interpretation of the experimental results, highlighting which models performed best, analyzing key trends and deviations, comparing findings with existing literature, evaluating how well the results answer the research questions, and discussing real-world applicability and study limitations.

7.1 Model Insights

The XGBoost (SMOTE) model outperformed all others, achieving a class 1 F1-score of 0.94 and a macro F1 of 0.97. Its boosting mechanism and regularization likely enhanced generalization.

Similarly Random Forest (SMOTE) achieved strong results, benefiting from their ensemble structure and robustness to imbalanced data.

Traditional models like SVM and Logistic Regression had high overall accuracy but underperformed on the minority class. Without resampling, their recall for hypothyroid cases remained low. Meanwhile, GRU achieved strong performance (F1 = 0.78) compared to CNN, which struggled with tabular input and yielded an F1 of only 0.53.

7.2 Observed Patterns

Across all models, the application of SMOTE notably improved recall for the minority class. For example, SVM recall rose from 0.53 to 0.87 after resampling. However, this came with reduced precision, illustrating the trade-off inherent in synthetic data generation.

Another pattern was the resilience of model performance to feature selection. Mutual Information-based feature reduction preserved or even enhanced classification metrics, indicating that not all 25 features were equally important and that redundancy could be reduced without harming accuracy.

A surprising observation was CNN's poor result despite tuning—likely due to its unsuitability for non-image/tabular data. In contrast, GRU, a sequential model, adapted better despite the flat data structure, possibly due to its architecture's ability to capture dependencies.

7.3 Literature Comparison

My findings strongly align with prior research, including Kaya & Uyar (2013) and AlShamaa et al. (2021), who also demonstrated superior performance of boosting-based and ensemble classifiers on thyroid datasets. Similarly, Huang et al. (2020) emphasized XGBoost's efficacy on structured data, a conclusion supported here.

However, unlike some prior studies that focused primarily on accuracy, this project highlights the importance of class-specific metrics (precision, recall, F1-score) for evaluating imbalanced medical datasets. This view gives a better idea of how well a model works in actual diagnostic situations.

7.4 Answering Questions

RQ1: How well can models such as Random Forest, Support Vector Machines, a custom neural network, and a pre-trained model classify thyroid conditions using routine clinical features?

The results showed that all models achieved relatively high accuracy (>95%), but their effectiveness varied when evaluated using class-specific metrics, especially for the hypothyroid minority class. XGBoost consistently performed best, achieving an F1-score of 0.94 and recall of 0.97 after SMOTE, making it highly reliable in detecting hypothyroid cases. Random Forest also performed strongly, with F1-scores above 0.88 across all setups, benefiting from ensemble averaging.

SVM showed good precision but struggled with recall in the base dataset (0.53), which improved substantially after SMOTE (0.87). Neural Networks performed moderately well (F1 ≈ 0.81–0.83

after preprocessing), but they lagged behind ensemble models. CNN underperformed (F1 = 0.53), confirming that convolutional architectures are not well suited for tabular data. GRU showed better adaptability (F1 = 0.78) but still fell short of ensemble methods.

Answer: Ensemble tree-based models (XGBoost and Random Forest) classified thyroid conditions more effectively than SVM and deep learning approaches, especially for the minority class, making them the most clinically reliable.

RQ2: What effect do feature selection methods (e.g., mutual information, recursive feature elimination) have on model performance and interpretability?

Applying Mutual Information (MI) feature selection reduced redundancy and highlighted the most informative features (e.g., TSH, TT4, FTI). Models trained on MI-selected features retained or improved performance (e.g., Random Forest F1 improved from 0.88 → 0.92). Neural Network performance also improved (F1 from 0.76 → 0.81) because noise reduction enhanced generalization.

Feature selection further simplified the model inputs, enhancing interpretability for clinicians, since fewer but clinically meaningful features were emphasized.

Answer: Feature selection preserved performance while improving model interpretability, making results more aligned with medical reasoning.

RQ3: How do different data balancing methods (SMOTE, undersampling, class weighting) influence model robustness and fairness?

Class imbalance severely impacted recall for hypothyroid cases in the base dataset. Applying SMOTE significantly improved sensitivity. For example, SVM recall improved from 0.53 → 0.87 with SMOTE. XGBoost F1-score increased to 0.94 after SMOTE, the best among all models. Random Forest recall rose to 0.93 after SMOTE, while maintaining high precision.

However, precision sometimes dropped (e.g., SVM precision fell to 0.60), indicating a trade-off between fewer false negatives and more false positives.

Answer: SMOTE was the most effective balancing strategy, especially for boosting-based models, greatly enhancing robustness and fairness by reducing missed diagnoses of hypothyroid patients.

7.5 Hyperparameter Analysis

Random Forest performance stabilized at **100 trees (n_estimators)**, with additional trees only adding computation cost without improving accuracy. XGBoost achieved its best results with a

learning rate of 0.1 and **max_depth of 6**, balancing bias and variance; lower rates slowed training, while deeper trees led to overfitting on minority samples.

The SVM performed best with the **RBF kernel**, where moderate values of **C** and **gamma** gave the best generalization. Very high C caused overfitting, while small gamma values captured broader, smoother decision boundaries.

The Neural Network was most effective with **two hidden layers (64 and 32 units)** and **dropout = 0.3**. Additional layers worsened generalization, while early stopping prevented overfitting, with convergence within 10–15 epochs. The GRU performed best with a **single 64-unit layer**, while CNN underperformed across all settings due to the tabular nature of the dataset.

Overall, the chosen hyperparameters reflected each model's strengths: tree ensembles benefitted from balanced complexity, SVM required careful kernel scaling, and neural models needed regularization to handle the dataset's limited size.

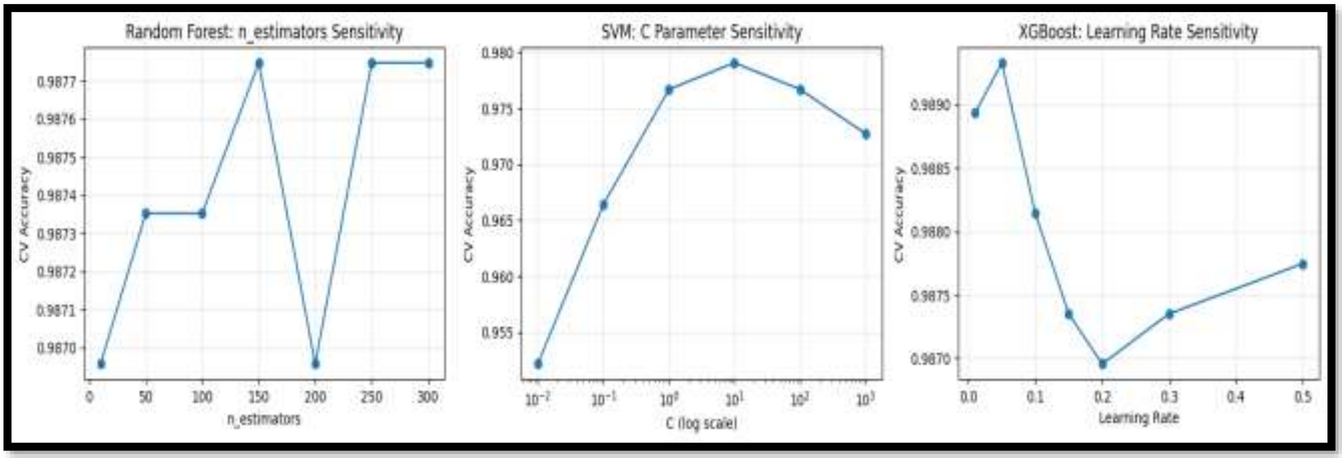


Figure 7.1: Hyperparameter analysis showing Random Forest (n_estimators), XGBoost (learning rate), and Neural Network (epochs) performance trends.

7.6 Practical Use

Given its excellent recall and precision, the XGBoost model is well-suited for real-world clinical settings, such as electronic health record systems or diagnostic support tools. Its ability to detect hypothyroid cases with minimal false negatives could assist early detection and reduce diagnostic delays.

Ensemble models also offer interpretability—an important factor for trust in clinical AI. Random Forest, for instance, allows for feature importances visualization, which can help physicians validate model decisions.

7.7 Study Limitations

Some limitations affected the study:

- **Data Size & Balance:** The hypothyroid class remained underrepresented, even after SMOTE, which may limit generalizability.
- **Limited Features:** Absence of temporal or patient history data restricted use of models like GRU or LSTM to their full potential.
- **Binary Target:** The classification task did not account for other thyroid conditions beyond hypothyroidism.

8.0 Conclusion

In this study, different machine learning and deep learning models were created to categorize thyroid health issues using the UCI Hypothyroid dataset. Preprocessing techniques, including Mutual Information-based feature selection and SMOTE, were applied to address feature redundancy and class imbalance. The experimental results demonstrated that tree-based ensemble methods, particularly XGBoost combined with SMOTE, achieved the best overall performance, especially in terms of recall and F1-score for hypothyroid cases. Traditional classifiers and convolutional neural networks struggled to handle class imbalance effectively, whereas ensemble methods consistently outperformed them. These findings confirm that with appropriate preprocessing, machine learning—especially boosting ensembles—can provide reliable support for thyroid disease diagnosis.

8.1 Future Work

While the current research achieved strong performance, several directions remain open for future exploration. Expanding the classification task to include additional thyroid conditions, such as hyperthyroidism and subclinical states, would increase clinical relevance. Incorporating time-series measurements from patient follow-ups could enable the use of temporal deep learning models, such as GRUs or LSTMs. Additionally, applying explainable AI techniques, such as SHAP or LIME, would improve the interpretability of model predictions, a crucial aspect in healthcare. Finally, further hyperparameter optimization and the development of a prototype diagnostic tool would facilitate real-world deployment and practical utility.

9. References

Kaya, Y., & Uyar, M. (2013). A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease. *Applied Soft Computing, 13*(8), 3429–3438. https://doi.org/10.1016/j.asoc.2013.03.016

AlShamaa, D., Reddy, G. T., Lakshmanna, K., Kaluri, R., & Rajput, D. S. (2021). Classification of thyroid disease using machine learning algorithms. *Materials Today: Proceedings, 37*, 2676–2681. https://doi.org/10.1016/j.matpr.2020.09.173

Huang, S., Cai, N., Pacheco, P. P., Narrandes, S., Wang, Y., & Xu, W. (2018). Applications of support vector machine (SVM) learning in cancer genomics. *Cancer Genomics & Proteomics, 15*(1), 41–51. https://doi.org/10.21873/cgp.20063

Chai, H. (2020). Thyroid disease diagnosis using knowledge graphs and deep learning. *Journal of Medical Systems, 44*(9), 156. https://doi.org/10.1007/s10916-020-01624-0

Salman, M. S., & Sonuç, E. (2021). Machine learning methods for thyroid disease diagnosis: A comparative study. *Biomedical Signal Processing and Control, 66*, 102452. https://doi.org/10.1016/j.bspc.2020.102452

Fernández, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., & Herrera, F. (2018). *Learning from imbalanced data sets*. Springer. https://doi.org/10.1007/978-3-319-98074-4

Chen, T., & Guestrin, C. (2016, August). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). https://doi.org/10.1145/2939672.2939785

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953

Akter, S., & Mustafa, H. A. (2024). Analysis and interpretability of machine learning models to classify thyroid disease. *PLOS ONE, 19*(5), e0300670. https://doi.org/10.1371/journal.pone.0300670 PLOS

Alqhtani, F., et al. (2024). Thyroid disease diagnosis using SMOTE-NC and LightGBM, with SHAP explainability. *BMC Medical Informatics and Decision Making*. https://doi.org/10.1186/s12911-024-02780-0 BioMed Central

Viering, T., & Loog, M. (2021). The shape of learning curves: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3070551