**Thyroid Disease Classification**

A Multiclass Classification Approach

Low Level Design (LLD)

Revision Number: 1.0

Last date of revision: 03/05/2023

Mohammad Junaid

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 03rd May 2023 | 1.1 | First Draft | M. Junaid |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_gjdgxs)

[**Abstract** 4](#_30j0zll)

[1](#_1fob9te) Introduction 5

[1.1](#_3znysh7) Why this Low-Level Design Document? 5

[1.2](#_2et92p0) Scope 6

[1.3](#_tyjcwt) Constraints 6

[1.4](#_3dy6vkm) 1.3 Risks 6

[1.5](#_1t3h5sf) 1.4 Out of Scope 7

[2](#_4d34og8) Technical specifications 7

[2.1 Predicting Disease 7](#_2s8eyo1)

[2.2 Logging 7](#_17dp8vu)

[2.3 Database 7](#_3rdcrjn)

[3](#_26in1rg) Technology stack 8

[4](#_lnxbz9) Proposed Solution 9

[5](#_35nkun2) Model training/validation workflow 10

[6](#_1ksv4uv) User I/O workflow 12

[7](#_44sinio) Exceptional scenarios 13

**Abstract**

Thyroid diseases, such as hyperthyroidism and hypothyroidism, affect a significant portion of the population and can lead to serious health complications. Accurate diagnosis and early intervention are crucial for effective treatment. In this project, we aim to develop a machine learning-based approach for the early detection and classification of thyroid disease. Our approach involves the use of multiclass classification models to predict the likelihood of a patient having a diseased state of the thyroid. We utilize various machine-learning techniques and feature engineering methods to achieve accurate and reliable predictions. Our system also includes data preprocessing, model training and evaluation, and automated reporting of data drift and model performance to ensure the model remains robust and effective. Overall, our project aims to provide a reliable and efficient tool for healthcare professionals in the diagnosis and treatment of thyroid disease.

# Introduction

## Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the Thyroid Disease Classifier model. It will explain the purpose and features of the model, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The main objective of the project is to predict if a person have thyroid disease if yes the which disease based on the thyroid diagnosis report.

An Thyroid diagnosis report contains patient health information, such as:

* Age
* Sex
* Sick or not
* Pregnant or not
* Thyroid Surgery or not
* Goitre or not
* Tumor or not
* Hypopituitary or not
* Hypothyroid or not
* Hyperthyroid or not
* Psych
* Thyroxine therapy or not
* Antithyroid Medication or not
* I131 treatment or not
* TSH Level
* T3 level
* T4 level
* T4U level
* TT4 level
* FTI level

This project shall be delivered in two phases:

Phase 1: All the functionalities with PyPi packages.

Phase2: Integration of UI to all the functionalities.

## Scope

This software system will be a Web application This system will be designed to detect the diseases at earliest for better disease management, improved interventions, and more efficient health-care resource allocation using previous diagnosis records available. More specifically, Early detection of any preventable diseases is important for better disease management. This system is designed to predict the diseases from patient information such as demographics, lab results, procedures and medications.

## Constraints

We group the 15 different conditions of thyroid into 8 different medical conditions.

# Technical specifications

## 2.1 Dataset

|  |  |  |
| --- | --- | --- |
| **Grouped** | **Disease** | **Source** |
| Negative | Negative | https://archive.ics.uci.edu/ml/datasets/thyroid+disease |
| Binding Protein | Increased Binding Protein Decreased Binding Protein |
| Hypothyroidism | Compensated Hypothyroid Primary Hypothyroid Secondary Hypothyroid |
| Hyperthyroidism | Hyperthyroid T Toxic Secondary Toxic |
| Goitre | Goitre |
| Replacement | Replacement Therapy Underreplacement Overreplacement |
| Sick | Sick |
| Discordant | Discordant |

## Diabetes dataset overview

Dataset consist of 5 datasets:

* Allbp ( All Binding Protein)
* Allhyper (All Hyperthyroid)
* Allrep (All Replacement Therapy)
* Allhypo (All Hypothyroid)
* Sick (All Sick)
* Dis (All Discordant)

## 2.1.2 Input schema

|  |  |
| --- | --- |
| **Feature name** | **Datatype** |
| Age | float |
| TSH | float |
| T3 | float |
| T4U | float |
| TT4 | float |
| FTI | float |
| Sex | category |
| On\_thyroxine | category |
| on\_antithyroid\_medication | category |
| query\_on\_thyroxine | category |
| Sick | category |
| Pregnant | category |
| Thyroid surgery | category |
| I131\_treatment | category |
| query\_hypothyroid | category |
| query\_hyperthyroid | category |
| lithium | category |
| goitre | category |
| tumor | category |
| hypopituitary | category |
| Psych | category |
| referral\_source\_SVHC | category |
| referral\_source\_SVHD | category |
| referral\_source\_SVI | category |
| referral\_source\_other | category |
| major\_class\_encoded | category |

## 2.2 Predicting Disease

* User needs to answer yes/no type of questions related to thyroid therapies, the harmonal tests and diagnosis
* The user gives required information.
* The system should be able to make predictions regarding the one of the medical condition of thyroid.

## 2.3 Logging

We should be able to log every activity done by the user.

* The System identifies at what step logging required
* The System should be able to log each and every system flow.
* Developers can choose logging methods. You can choose database logging/ File logging as well.
* System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

## 2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User chooses the disease.
2. The User gives required information

The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

**2.5 Deployment**

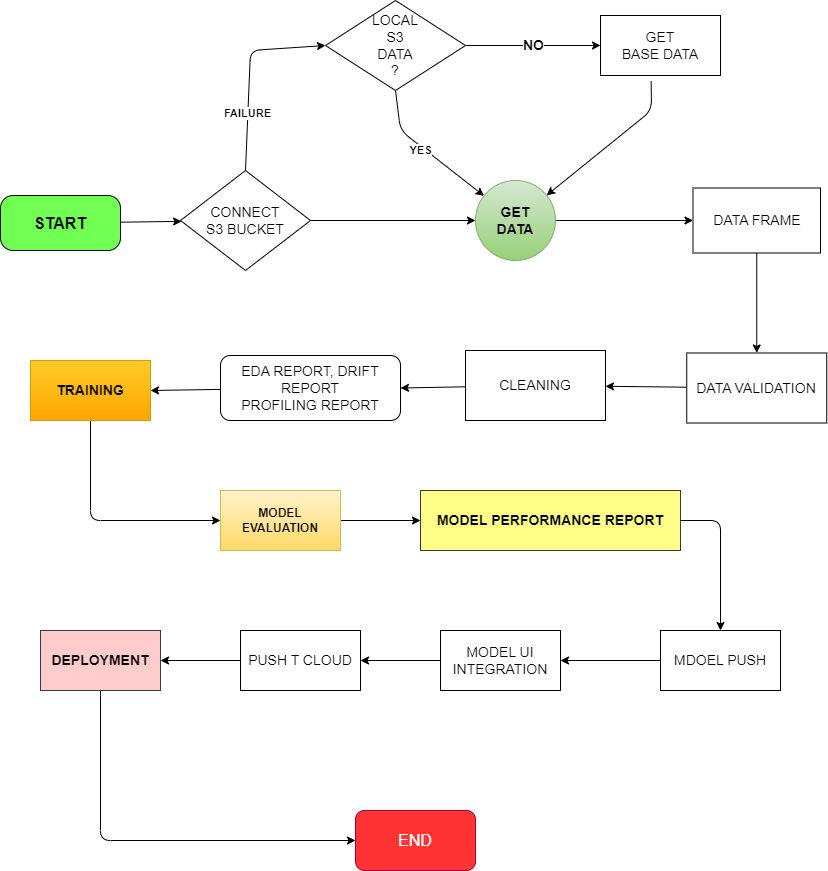
1. AWS
2. Azure
3. Render

# Technology stack

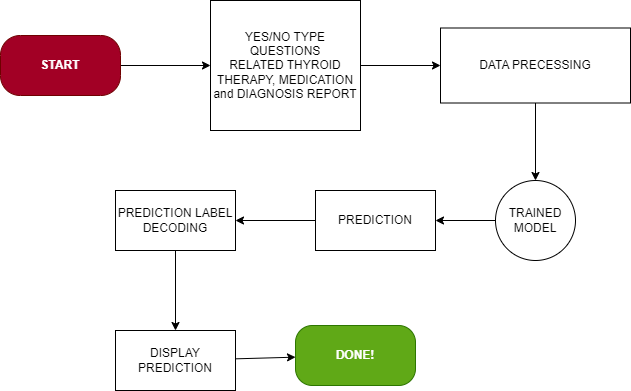
|  |  |
| --- | --- |
| **Front End** | HTML/CSS/JS |
| **Backend** | Python Flask |
| **Database** | MongoDB/MySql |
| **Deployment** | AWS, AZURE, RENDER |

# Proposed Solution

# Model training/validation workflow



# User I/O workflow

****

# Exceptional scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| Step | Exception | Mitigation | Module |
| 3rd May 2023 | 1.0 | First Draft | M Junaid |
|  |  |  |  |

# Test cases

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | Steps to perform test case | Module | Pass/Fail |
|  |  |  |  |

# Key performance indicators (KPI)

* F1 weighted score
* ROC AUC ovr weighted score
* Balanced Accuracy
* Log loss