Low Level Document

Project Title: Thyroid Disease Detection

Technologies: Machine Learning Technology

Domain: Healthcare

Project Difficulty Level: Intermediate

Version No.: 1.0

Last Date of Revision: 30, Aug 2024

DOCUMENT VERSION CONTROL

Date	Version	Description	Author
30, Aug 2024	1.0	Introduction and Architecture defined, explaining the details of technical implementation, including data preprocessing, model selection, optimization and deployment.	Raksha Ray

CONTENTS

1. ABSTRACT

A thorough description of the implementation procedure is given in the Low Level Documentation for the Thyroid Disease Detection project, with an emphasis on the technical elements of developing a machine learning model for thyroid disease prediction. In order to guarantee a clean and reliable dataset, this paper describes the step-by-step processes involved in data preprocessing, such as addressing missing values, encoding categorical characteristics, and normalising numerical data. It also discusses feature engineering methods like feature modification and selection that areused to increase the dataset's predictive power.

The document explores the process of choosing a model, going over the several machine learning techniques that were assessed. These algorithms include XGBoost, logistic regression, decision trees, support vector machines, random forest classifier, and K-Nearest Neighbours (KNN) classifier. The model evaluation metrics, including recall, accuracy, precision, F1 score, confusion matrix, and classification report, are described in depth, along with the model optimisation approaches used to fine-tune performance. The deployment procedure is also described, including how to integrate the model into a Flask web application for real-time prediction and build up an ongoing monitoring system to guarantee the model's effectiveness over time.

With an emphasis on developing dependable and scalable solutions for thyroid illness diagnosis, this thorough documentation acts as a technical guide for developers and data scientists, providing insights into the real-world application of machine learning in healthcare.

2. INTRODUCTION

The Low Level Documentation offers a thorough overview of the Thyroid Disease Detection project's technical components, including information on the feature engineering, deployment, data preparation, and model selection procedures. This paper provides a resource for comprehending the procedures involved in implementing machine learning-based diagnosis, guaranteeing precision and efficacy.

2.1 Purpose

The purpose of a low-level design document, or LLD, is to provide the internal logical design of the Thyroid Disease Detection System's actual program code. Program specifications and class diagrams with their corresponding methods and relationships are described by LLD. It explains the modules such that the programmer may start writing code right out of the page.

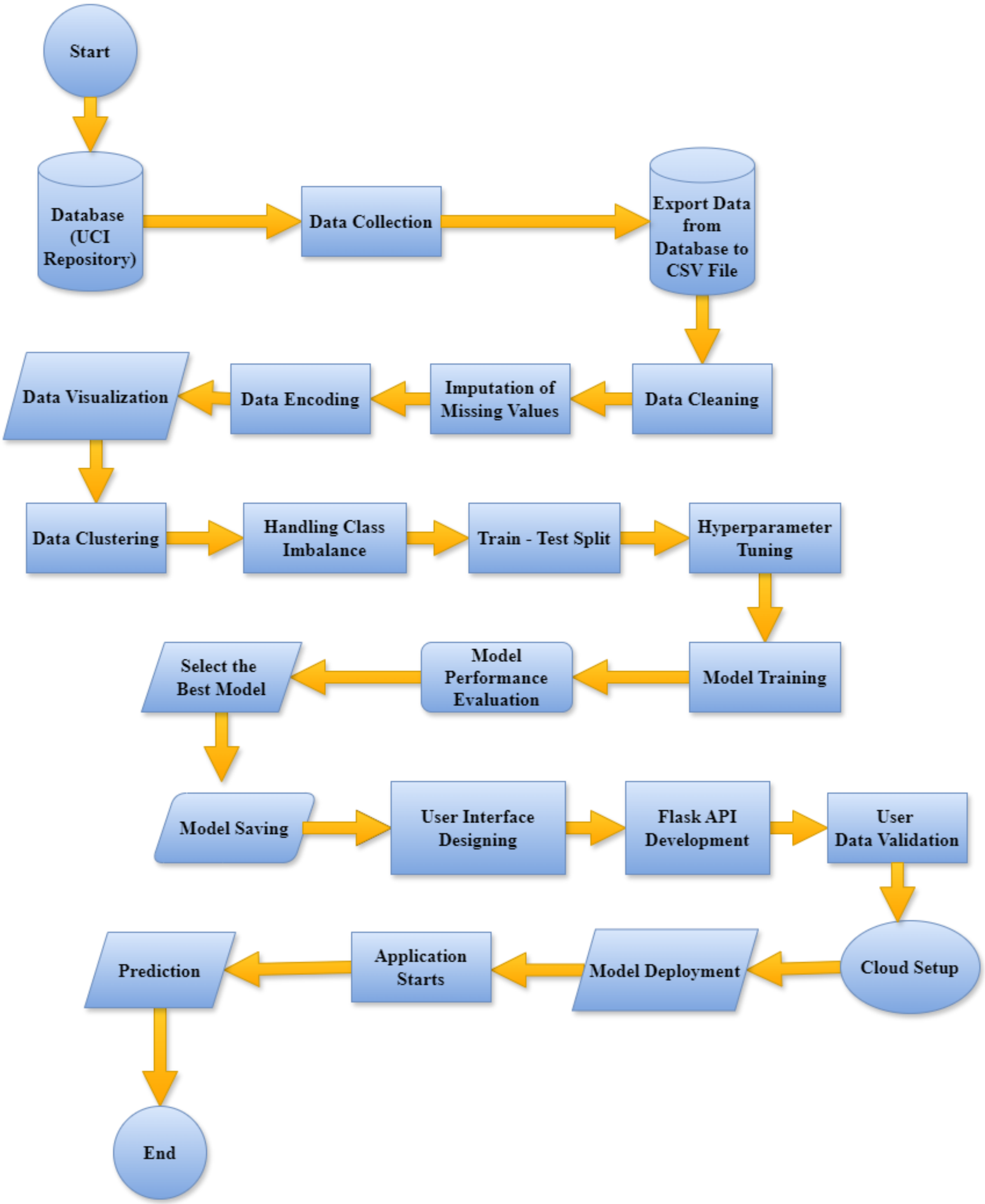
2.2 Scope

A methodical refining process is followed in the component-level design process known as low-level design (LLD). Data structures, necessary software architecture, source code, and, in the end, performance algorithms can all be designed using this method. In general, requirement analysis may outline the data organisation, which may subsequently be further developed during data design work.

3. ARCHITECTURE

3.1 Architecture Diagram

The Architecture of the Project has been shown below:



3.2 Architecture Description

The Architecture of the Project can be described as follows:

- ***Data Description:-***
  - The Data used in this project is the Thyroid Disease Dataset present in UCI Machine Learning Repository. Link to the repository - <Thyroid Disease - UCI Machine Learning Repository>
  - The Data from the above mentioned repository is exported to MongoDB Database and transformed into a CSV File.
  - The CSV File is loaded into Jupyter Notebook (used in VS Code) andread by using Pandas Library. This is done in order to proceed further in the project.
- ***Exploratory Data Analysis (EDA):-***
  - In this segment of the Project Work, firstly the dataset is explored in the Jupyter Notebook (used in VS Code), in order to get the initialinsights about the dataset.
  - The duplicate values are dropped.
  - The Missing Values are imputed.
  - The Categorical Columns have been encoded, so that they can beused in Machine Learning operations, in the subsequent steps.
  - Data Visualization is performed by using Matplotlib and Seaborn Libraries.
- ***Machine Learning Operations:-***
  - K – Means Algorithm has been used to create clusters in the pre-processed data and the optimum number of clusters is selected byplotting the elbow plot.
  - The class imbalance is handled.
  - The dataset is split into Training Dataset and Test Dataset.
  - Hyperparameter tuning is done, so that the performance of models become better.
  - Models are trained.
  - The performance of the models are evaluated.
  - The Best Performing Model is selected.
  - The Best Model is saved.
- ***Web – App Development:-***
  - The user interface is designed by using HTML, CSS and JavaScript.
  - The Flask app is developed.
  - Cloud Setup is done so that the model can be deployed.
  - The Application starts, once the model is deployed.
  - The client enters data as required and once, the Submit button is clicked, the prediction starts.
  - Once the prediction is done, the Predicted Result is displayed onthe screen.

4. UNIT TEST CASES

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	Application URL should be defined	Application URL should be accessible to the user
Verify whether the Application loads completely for the user when the URL is accessed	1. Application URL is accessible 2. Application is deployed	The Application should load completely for the user when the URL is accessed
Verify whether the User is able to sign up in the application	Application is accessible	The User should be able to sign up in the application
Verify whether user is able to successfully login to the application	1.Application is accessible 2.User is signed up to the application	User should be able to successfully login to the application
Verify whether user is able to see input fields on logging in	1.Application is accessible 2.User is signed up to the application 3.User is logged in to the application	User should be able to see input fields on logging in
Verify whether user is ableto edit all input fields	1.Application is accessible 2.User is signed up to the application 3.User is logged in to the application	User should be able to editall input fields
Verify whether user gets Submit button to submit the inputs	1.Application is accessible 2.User is signed up to the application 3.User is logged in to the application	User should get Submit button to submit the inputs

5. CONCLUSION

The technical details required to comprehend and duplicate the project's implementation are contained in the Low Level Documentation for the Thyroid Disease Detection project. A comprehensive process roadmap for the development of a machine learning model for thyroid disease prediction is provided in this paper, which follows an organised format with sections for the Abstract, Introduction, Scope, Purpose, and Architecture.

The sequence from data collection to preprocessing, model training, and deployment is outlined in the thorough architecture, which makes sure that each stage is distinct and makes sense. This documentation acts as a basis for the project's future revisions and additions in addition to being a source of reference for the existing implementation. Developers, data scientists, and other stakeholders may better understand and collaborate with one other by providing insights into the design and decision-making processes.

To sum up, this Low Level Document is an essential tool for all project participants, guaranteeing openness, uniformity, and the capacity to efficiently scale or maintain the system. It contributes to the continuous efforts to enhance patient outcomes and diagnostic accuracy in the field of thyroid disease detection by demonstrating the completeness and precision necessary in the development of a dependable machine learning application in healthcare.