ARCHITECTURE DOCUMENT

P r o j e c t : Thyroid Disease Detection

Technologies: Machine Learning Technology

Domain: Healthcare

Project Difficulty Level: Intermediate
Version No.: 1.0

Last Date of Revision:
30 Aug, 2023

DOCUMENT VERSION CONTROL

Date	**Version**	**Description**	**Author**
30 Aug, 2024	1.0	Introduction and Architecture defined.	Raksha Ray

CONTENTS

1. ABSTRACT

The Thyroid Disease Detection project's architecture is made to smoothly combine web development, machine learning, and data processing in order to accurately forecast thyroid illnesses. The entire procedure is described in this guide, from gathering data from the UCI Machine Learning Repository to preparing it for analysis and importing it into a MongoDB database.

The design goes on to describe the cleaning, preprocessing, and visualisation of the data for preliminary insights during the exploratory data analysis (EDA) phase. To create a strong prediction model, machine learning operations such as handling class imbalance, clustering, and hyperparameter tweaking are then carried out. The top- performing model is then chosen, saved, and included into a Flask-developed, user- friendly online application. With the help of an interactive interface that allows users to enter data and obtain predictions in real time, the program is deployed in a cloud environment.

This documentation ensures transparency and repeatability in the development process by providing a detailed guide that covers every aspect of the project's architecture.

2. INTRODUCTION

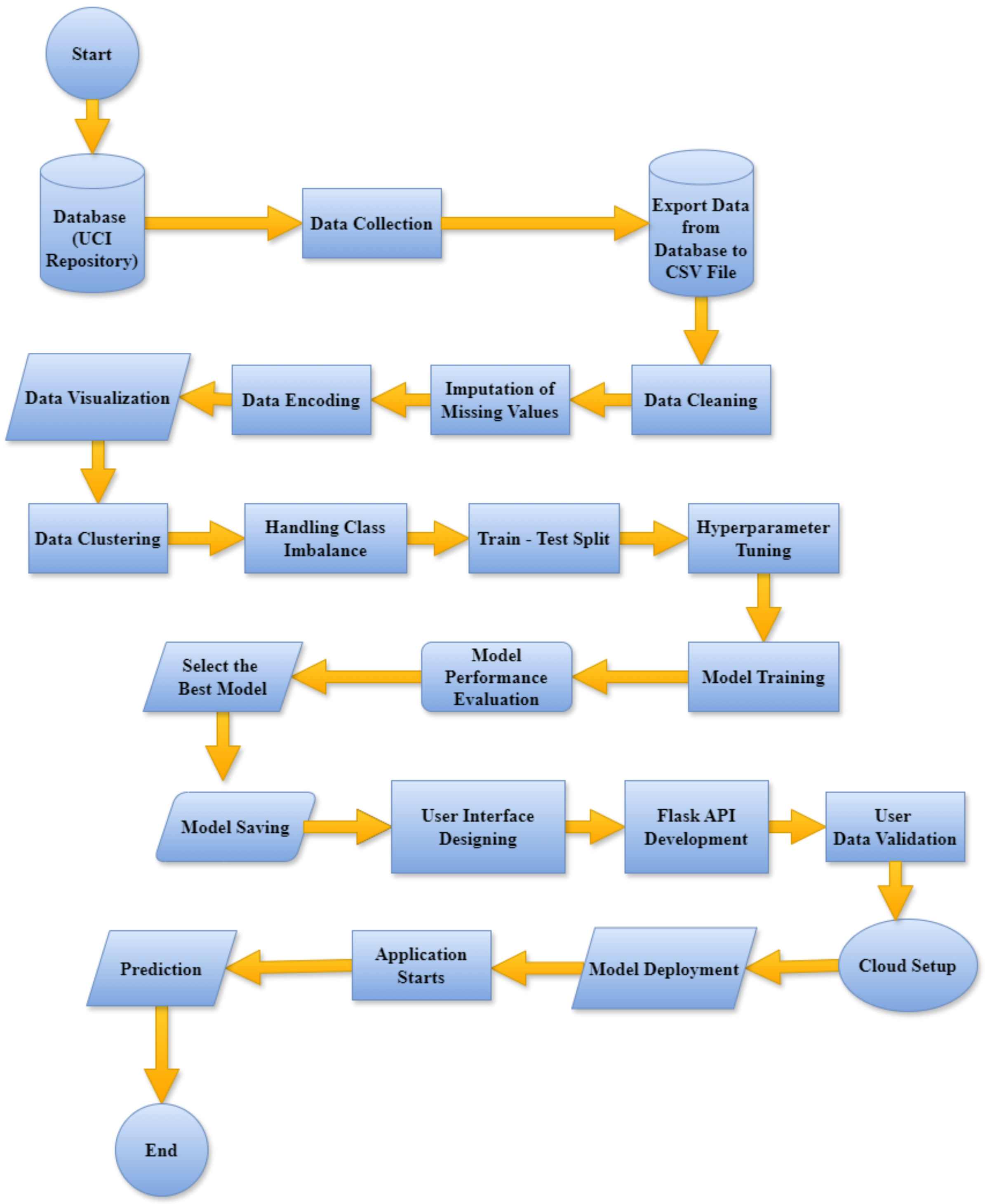
This architecture documentation aims to give a thorough and organised summary of the Thyroid Disease Detection project's conception and execution. It functions as a guide for comprehending the many elements and procedures involved in creating and implementing a web application based machine learning model for the prediction of thyroid problems.

Everything from data collection and preprocessing to model training, assessment, and deployment is covered in the documentation. This documentation covers every important facet of the project, including web development, machine learning procedures, exploratory data analysis (EDA), and data administration. This document attempts to aid in a clear knowledge of the architecture, empowering stakeholders, data scientists, and future developers to efficiently duplicate, maintain, or expand the system. It does this by thoroughly documenting each stage of the project. It also guarantees that all procedures are open and compliant with industry standards for software and machine learning development, which enhances the project's overall dependability and expandability.

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) and read 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 initial insights about the dataset.
  - The duplicate values are dropped.
  - The Missing Values are imputed.
  - The Categorical Columns have been encoded, so that they can be used 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 by plotting 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 on the 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 able to 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 edit all 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

A thorough and in-depth manual for the design, development, and implementation procedures necessary to produce a successful predictive model can be found in the Architecture Documentation for the Thyroid Disease Detection project. This documentation makes sure that every step—from data collection and preprocessing to machine learning operations and web application development—is well-integrated and in line with the project's goals by detailing each one. A deeper comprehension of the technical workflow is made possible by the comprehensive explanation of the architecture, which also acts as a guide for future project scalability and maintenance.

The documentation highlights the significance of every stage, emphasising the careful data handling, thorough model evaluation, and user-centred web application design that all play a part in the project's accomplishment. This paper is going to be a key reference as the project develops, helping to make sure the system stays reliable, scalable, and able to provide accurate forecasts. In the end, the Architecture Documentation strengthens the project's structure and guarantees its long-term viability and efficiency in the medical diagnostics industry.