

# Low Level Design

## Thyroid Disease Detection

Written By	Vrushabh Vaishnav
Document Version	1.0
Last Revised Date	23 – sept-2023

## Document Control

### Change Record:

Version	Date	Author	Comments
0.1	19 – Sept - 2023	Vrushabh vaishnav	Introduction & Architecture defined
0.2	20 – sept - 2023	Vrushabh vaishhnav	Architecture & Architecture Description appended and updated
0.3	21- sept - 2023	Vrushabh vaishhnav	Unit Test Cases defined and appended

### Reviews:

Version	Date	Reviewer	Comments

**Approval Status:**

Version	Review Date	Reviewed By	Approved By	Comments

## Contents

### DOCUMENT VERSION CONTROL

<b>1. Introduction .....</b>	<b>1</b>
<b>1.1. What is Low-Level design document? .....</b>	<b>1</b>
<b>1.2. Scope .....</b>	<b>1</b>
<b>2. Architecture .....</b>	<b>2</b>
<b>3. Architecture Description .....</b>	<b>3</b>
<b>3.1. Data Description .....</b>	<b>3</b>
<b>3.2. Web Scrapping.....</b>	<b>3</b>
<b>3.3. Data SPLITTING.....</b>	<b>3</b>
<b>3.4. Data PREPROCESSING.....</b>	<b>3</b>
<b>3.5. MODEL TRAINING.....</b>	<b>4</b>
<b>3.6. MODEL EVALUATION.....</b>	<b>4</b>
<b>3.7. MODEL SAVING.....</b>	<b>4</b>
<b>3.9. CLOUD SETUP.....</b>	<b>4</b>
<b>3.10. PUSH APP TO CLOUD.....</b>	<b>4</b>
<b>3.11.DATA FROM CLIENT SIDE FOR PREDICTION .....</b>	<b>5</b>

<b>3.13. EXPORT PREDICTION TO CSV .....</b>	<b>5</b>
<b>4. Unit Test Cases.....</b>	<b>6</b>



## 1. Introduction

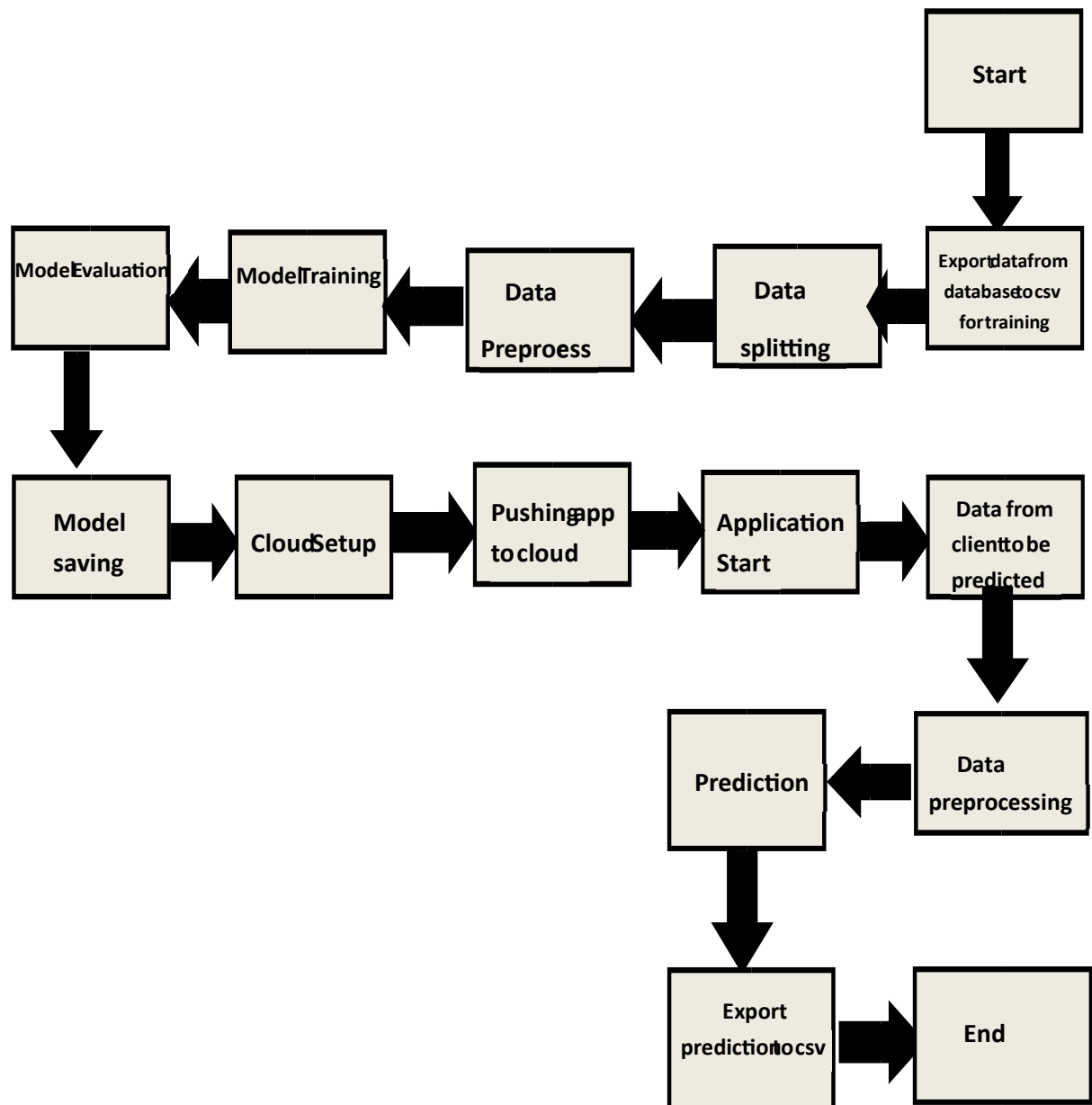
### 1.1. What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Food Recommendation System. LLD describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

### 1.2. Scope

Low-level design (LLD) is a component-level design process that follows a step-bystep refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

## 2. Architecture





### 3. Architecture Description

#### 3.1. Data Description

We will utilize Thyroid Sickness Informational collection present in UCI AI Store. This Informational collection is fulfilling our information necessity. Absolute 3772 examples present in various clumps of information.

#### 3.2. Web Scrapping

Here we will trade all bunches of information from data set into one csv document for preparing.

#### 3.3. Data Splitting

We channel the sections for dividing the information for train and test for additional purposes

#### 3.4. Data preprocessing

We will investigate our informational collection here and do EDA whenever required and perform information preprocessing relying upon the informational collection. We initially investigate our informational collection in Jupyter Scratch pad and choose what

pre-handling and Approval we have to do like attribution of invalid qualities, and so on and afterward we need to compose separate modules as per our examination, so we can carry out that for preparing as well as expectation information.

### 3.5. Data Training

We prepared a Random Forest Classifier model in our journal and was great on it. We prepared with our handled information.

### 3.6. MODEL EVALUATION

Model assessment done by grouping and report was saved to .pkl document .

### 3.7. MODEL SAVING

we will save our models so we can involve them for expectation reason

### 3.8. CLOUD SETUP

Here We will really do cloud arrangement for model organization. Here we additionally make our carafe application and UI and coordinate our model with jar application and UI.

### 3.9. PUSH APP TO CLOUD

In the wake of doing cloud arrangement and checking application locally, we will push our application to cloud to begin the application.

### 3.10. Data FROM CLIENT SIDE FOR PREDICTION PURPOSE

Presently our application on cloud is prepared for doing expectation. The forecast information which we get from client side .

### 3.11. DATA PREPROCESSING AND PREDICTION

Client information will likewise come similar cycle Information pre-handling and as indicated by that we will foresee those information.

### 3.12. EXPORT PREDICTION TO CSV

At long last when we get all the expectation for client information, then our last undertaking is to send out forecast to csv document and hand over it to client.

#### 4. Unit Test Cases

Test Case Description	Pre-Requisite	Expected Result
Verify whether the Application URL is accessible to the user	1. 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	1. 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 app.	User should be able to successfully login to the application
Verify whether user is able to see input fields logging in	1. Application is accessible 2. User is signed up to the application	User should be able to see input fields on logging in

	3. User is logged in to the application	
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
Verify whether user get prediction/output back after submitting the inputs.	1. Application is accessible 2. User is signed up to the application 3.	User should be presented with recommended results on clicking submit

	User is logged in to the application	
Verify whether the output which user get is accordance to inputs user made.	1. Application is accessible 2. User is signed up to the application 3. User is logged in to the application	The output should be in accordance to the inputs user made
Verify whether user has options to download their results or not	3. Application is accessible 1. User is signed up to the application 3. User is logged in to the application	User should have options to download results as well .