Low Level Design

Insurance Premium Prediction

Written By	Roshan Kshirsagar
Document Version	0.3
Last Revised Date	05 Feb, 2024

Document Control

Change Record:

Version	Date	Author	Comments
0.1	02 – FEB-2024	Roshan Kshirsagar	Introduction & Architecture defined.
0.2	03 – FEB-2024	Roshan Kshirsagar	Architecture & Architecture Description appended and updated.
0.3	05 – FEB-2024	Roshan kshirsagar	Unit Test Cases defined and appended.

Contents

1. Introduction
   1.1. What is Low-Level design document?
   1.2. Scope.
2. Architecture
3. Architecture Description
   3.1. Data Description
   3.2. Data Ingestion
   3.3. Data Validation
   3.4. Data Transformation
   3.5. Model Trainer
   3.6. Model Evaluation
4. Unit Test Cases

1. Introduction

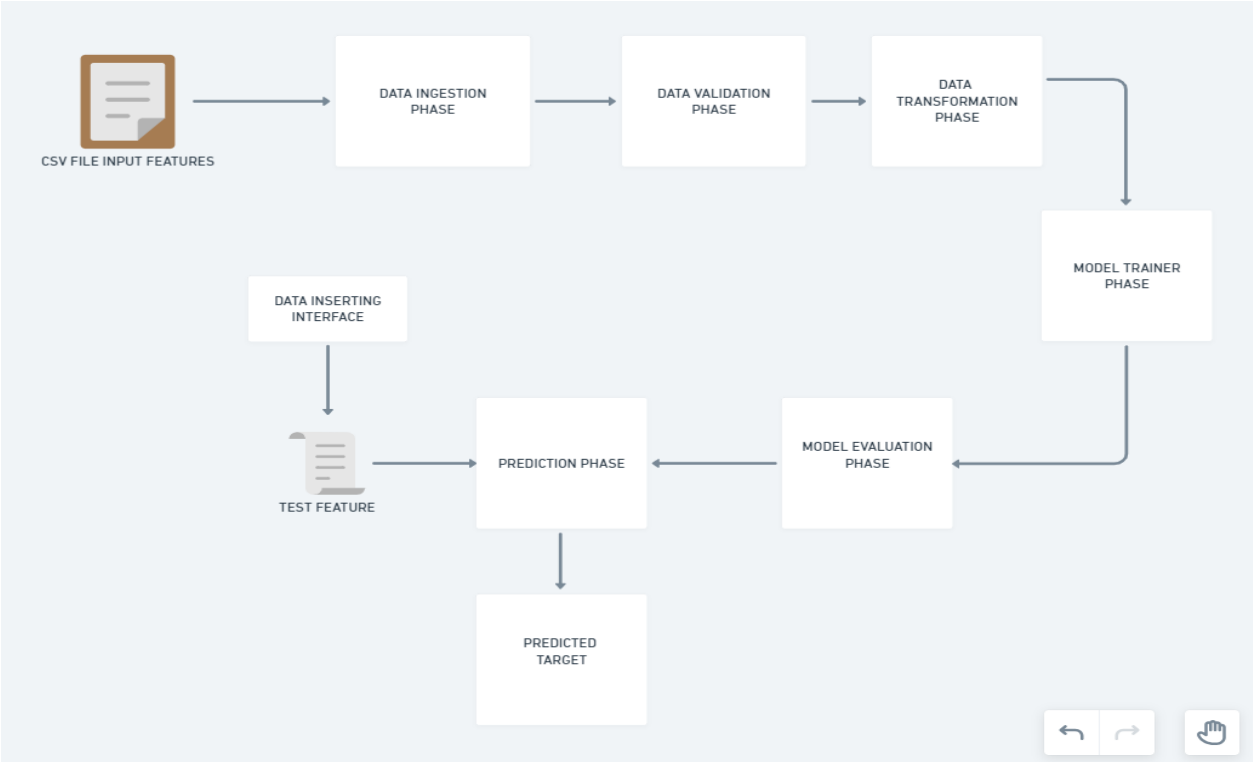
1.1 What is Low-Level design document?

The goal of LLD or a low-level design document (LLDD) is to give the internal logical design of the actual program code for Insurance Premium Prediction 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-by step refinement process. This process can be used for designing the 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

a. Data Description

This dataset contains records 1339 records along with 6 columns like age, sex, gender, region, bmi etc.

b. Data Ingestion

In the ingestion process, we will upload our dataset into the MongoDB database and retrieve this database and collection into the project through mongo client.

c. Data Validation

In data validation process, we check if our data is similar to our base data or not. This basically create the information for the retraining of the model on the another dataset.

d. Data Transformation

In this step, we will transform our data. We are using robust scaler for scaling our data and label encoder to encode our categorical features.

e. Model Training

This step include the training of our well performed model on our dataset. We are storing our best performed model into the model directory.

f. Model Evaluation

This step we check if our new trained model has better accuracy than previously trained one. If it

has better accuracy, then we will productionize the newest one.

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	Application URL is accessible.	The Application should load completely for the user when the URL is accessed
Verify whether user is able to edit all input fields	User should be able to access the url.	User should be able to edit all input fields.
Verify whether user gets Submit button to submit the inputs	User should be able to access the url.	User should get submit button to submit the inputs.
Verify whether user is presented with predicted results on clicking submit	User should be able to access the url.	User should be presented with predicted results on clicking submit