LOW-LEVEL DESIGN

Insurance Premium Prediction

Tavishi Jaglan

kalluri Vasanthsai

Document Version Control

DateIssued	Version	Description	Author
13.10.2021	V1.0	Initial LLD- V1.0	**Tavishi, Vasanthsai**

Tavishi Jaglan

kalluri Vasanthsai

Insurance Premium Prediction

Contents

1.0 Introduction

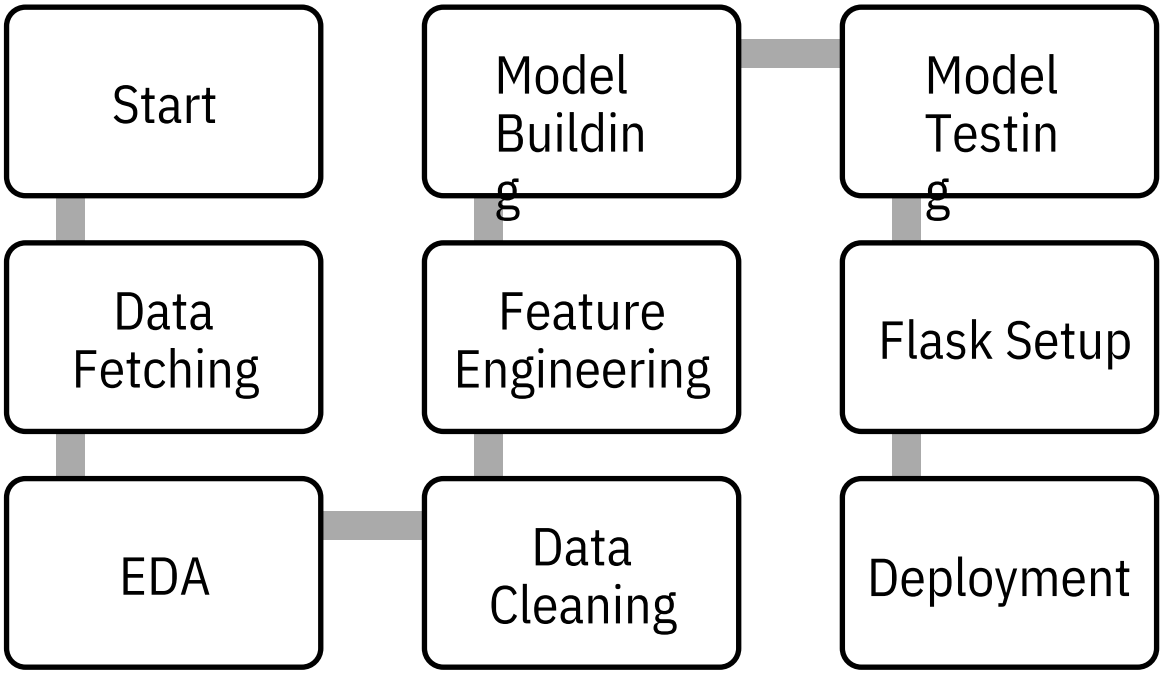
1.1 What is Low-Level Design Document?

The goal of LLD or Low-Level design document (LLDD) is to give the internal logical design of the actual program code. Low-Level design is created based on the High-Level design. 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 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 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.

2.0 Architecture



3.0 Architecture Description

3.1 Data Description

The primary source of data for this project from Kaggle. The dataset is comprised of 1338 records with 6 attributes. The data is in structured format and stored in a CSV file.

3.2 Exploratory Data Analysis

Exploring the data by visualizing the distribution of values in some columns of the dataset, and the relationships between expenses and other columns. Visualizing the distribution of age, BMI (body mass index). Also checking the region wise have any differences in the expenses.

3.3 Data Pre-processing

If data is not suited to take place directly for the regression. Then, cleaning of dataset becomes important for using the data under various regression algorithms.

3.4 Model Building

After data pre-processing is done, we will split the dataset into training set and validation set. Then we will use training set for building the best model. The model will be trained on several algorithms. We will

calculate RMSE and r2 score for each model and select the model with the best score.

3.5 Data Validation

Here Data Validation will be done on the test set.

3.6 Deployment

We will be deploying the model to Heroku platform.

4.0 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 URL is deployed	Application URL should load completely for the user when URL is accessed
Verify whether user can see input field after opening URL	1. Application is accessible	User should be able to see input fields after opening URL
Verify whether user can edit all the input fields	1. Application is accessible	User should be able to edit all the input fields
Verify whether user has options to filter the inputs fields	1. Application is accessible	User should filter the options of input fields
Verify whether user gets submit button to submit the inputs	1. Application is accessible	User should get submit button to submit the inputs
Verify whether user can see the output after submitting the inputs	1. Application is accessible	User should get outputs after submitting the inputs