

LOW LEVEL DESIGN (LLD)

HIGH LEVEL DESIGN (HLD)

Insurance Premium PredictionHIGH LEVEL DESIGN (HLD)

HIGH LEVEL DESIGN (HLD)

# Insurance Premium Prediction

NIKITA MAKARAND PANDE

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| **Date Issued** | **Version** | **Description** | **Author** |
| **0.1.10.2021** | V1.0 | Initial LLD- V1.0 | NIKITA |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

1.Introduction........................................................................................................1

1.1. What is Low-Level design document? .......................................................1

1.2. Scope..........................................................................................................1

1.3. Project Introduction................................................................................. 1

1.4. Problem Statement.................................................................................. 1

1.5. Database Information.............................................................................. 1

2. Architecture.................................................................................................. 2

3. Architecture Description............................................................................... 3

3.0 Architecture Description............................................................................ 3

3.1 Data Description........................................................................................ 3

3.2 Exploratory Data Analysis.......................................................................... 3

3.3 Data Pre-processing.................................................................................. 3

3.4 Model Building .......................................................................................... 3

3.5 Data Validation .......................................................................................... 3

3.6 Deployment ............................................................................................... 3

4. Unit Test Cases ................................................................................................4

**1. INTRODUCTION**

**1.1 What is a Low-Level Design Document?**

The goal of an LLD or Low-Level design document is to provide a clear and logical internal structure for the actual code that will be write. Low-Level design is create based on High-Level design, which describes class diagrams with methods and relationships between classes. It also outlines modules so that the programmer can understand how the program functions from within the document.

**1.2 What is Scope?**

Low-level design is a component-oriented design process that typically proceeds in a structured manner, starting with an analysis of the system's requirements and proceeding to development work on data structures, software architecture, and finally performance algorithms. Throughout this process, deriving relevant details about the overall data layout can be accomplish gradually as needed.

**1.3 Project Introduction**

This project is design to provide an estimate of the amount of health coverage a person needs depending on their specific situation. After that, customers can continue to work with any health insurance carrier and its plans without worrying about exceeding their budgeted costs from our study. This will help people focus on the positive aspects of having insurance, such as receiving proper medical care.

**1.4 Problem Statement**

The goal of this project is to provide an estimate of how much health coverage a person will need based on their specific situation. After receiving this information, customers can work with any insurance carrier and its plans while keeping in mind the estimated cost from our study. This could help someone concentrate on the positive aspects of having insurance instead of worrying about the expenses associated with it.

**1.5 Dataset Information**

The insurance\_data.csv dataset contains 1338 observations (rows) and 7 features (columns). The dataset contains 4 numerical features (age, BMI, children, and expenses) and 3 nominal features (sex, smoker, and region) that were converted into factors with numerical values designated for each level.

The goal of this exercise is to investigate different factors related to an individual's age, physical health and family conditions in order for the data to be use in a regression model designed to predict future medical expenses. This information will then be utilize by insurance companies as part of their decision-making process regarding premiums.

**2. Architecture**

End

Model Building

Visualization

Start

Deployment

Model Testing

Data Cleaning

Export data from csv

Flask Setup

Prediction

Data Transformation

Data Preprocessing

**3. 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

By reviewing and visualizing the distribution of values in some columns of a dataset, as well as assessing relationships between expenses and other variables, may provide valuable insights for future analysis. Additionally, exploring data by stratifying it based on age or BMI (body mass index) can identify any differences across regions.

3.3 Data Pre-processing

In order to properly utilized data for regression, it must be suitably clean. This process of cleaning the dataset involves removing any incorrect or invalid information.

3.4 Model Building

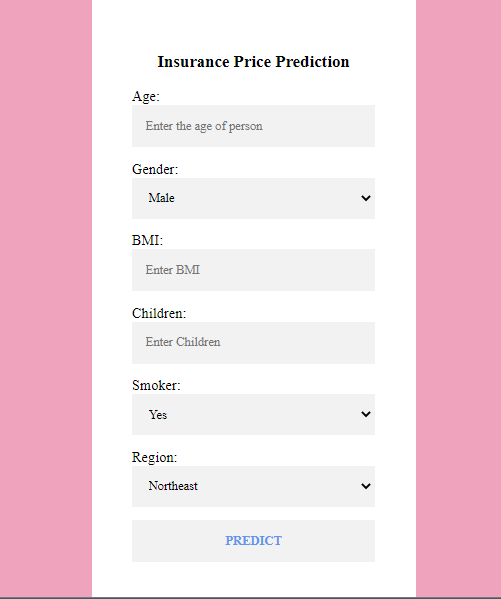
After data pre-processing is complete, the dataset will be divide into a training set and a validation set. The training set used to build the best model while the validation set will serve as an evaluation metric for this process. Several algorithms may be utilize during model building; RMSE and r2 score are among these metrics and after that choosing which algorithm produces the best result.

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 |
|  |  |  |  |