Low Level Design (LLD)

Insurance Premium Predictions



Revision Number: 1.1

Last date of revision: 16/08/2022

Kanchana chopra

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 16th Aug 2022 | 1.1 | Initial HLD-V1.0 | Kanchana chopra |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Contents**

[Document Version Control 2](#_Toc131702188)

[1 Introduction 5](#_Toc131702189)

[1.1 Why this Low-Level Design Document? 5](#_Toc131702190)

[1.2 Scope 6](#_Toc131702191)

[1.3 Constraints 6](#_Toc131702192)

[1.4 Risks 7](#_Toc131702193)

[1.5 Out of Scope 7](#_Toc131702194)

[2 Architecture 7](#_Toc131702195)

[2.1 architecture description 7](#_Toc131702196)

[2.2 Data requirement 7](#_Toc131702197)

[2.3 Data collection 8](#_Toc131702198)

[2.4. Data description 8](#_Toc131702199)

[2.5. Data transformation 9](#_Toc131702200)

[2.6. Exploratory data analysis 9](#_Toc131702201)

[2.7. Data preprocessing](#_Toc131702202)

[2.8. Machine learning algorithms Proposed Solution 11](#_Toc131702203)

[2.9. Evaluation metrics 12](#_Toc131702204)

2.10. Model building [12](#_Toc131702205)

[2.11. data from user 13](#_Toc131702206)

2.12. Data validation

## 2.13. User data feeding into Machine Learning model

## 2.14. Insurance premium prediction

2.15. Deployment

Model training/validation workflow

User I/O workflow

Test cases

**Abstract**

Predictive analytics has indeed become an essential tool for health insurance companies to accurately assess and predict the insurance premiums for their insured customers. By leveraging advanced data analytics technologies, insurers can gain deeper understanding of customer behaviour and habits, improve pricing and underwriting decisions and enhance the overall customer experience.

Some of the key factors that insurers typically consider when determining the policy premium include age, gender, BMI, smoking habits, medical history and number of dependents. By analysing these factors and identifying patterns and correlations, insurers can gain a more accurate understanding of each customer’s risk profile and adjust their policy premiums accordingly.

# Introduction

## Why this Low-Level Design Document?

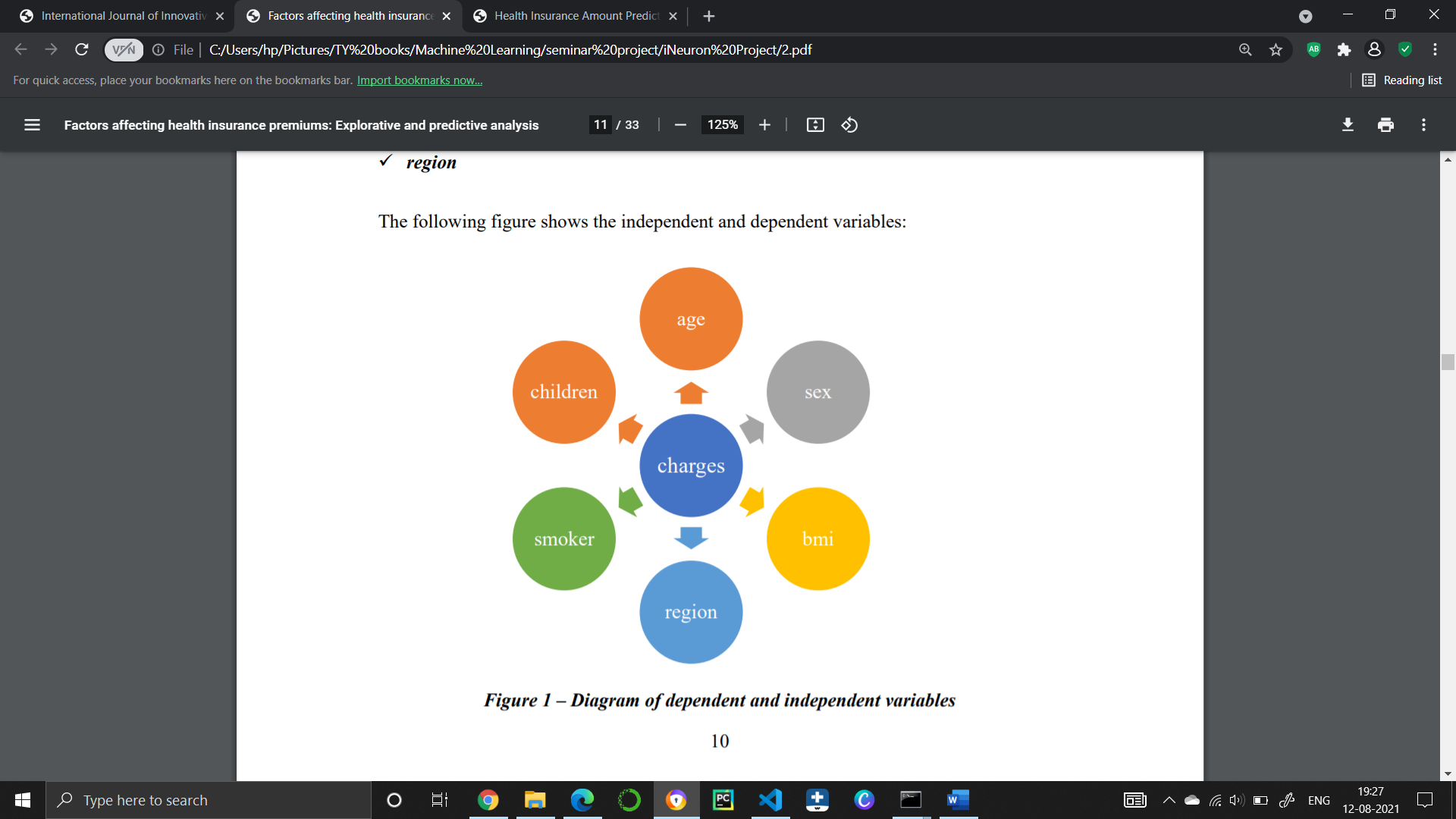
A low-level Design (LLD) document is a detailed document that describes the detailed design of a software or system. It includes all the technical specifications, algorithms, and data structures required for implementing the system.

The purpose of this document is to present a detailed description of the Insurance Premium Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the higher management for its approval.

The goal of this project is to allow a person to get an idea about the necessary amount required according to their own health status. Later they can comply with any health insurance company and their schemes & benefits keeping in mind the predicted amount from our project. This can help a person in focusing more on the health aspect of an insurance rather than the futile part.

An Insurance Premium Prediction contains the information, such as:

* age
* sex
* children
* bmi
* smoker
* region



This project shall be delivered in two phases:

Phase 1: All the functionalities with Scikit-learn packages.

Phase2: Integration of UI to all the functionalities.

## Scope

This software system will be a Web application. This system will be designed to predict health insurance premium. Premium amount prediction focuses on person’s own health rather than other company’s insurance terms and conditions. The models can be applied to the data collected in coming years to predict the premium. This can help not only people but also insurance companies to work in tandem for better and more health centric insurance amount. This system is designed to predict the health insurance premium based on some information like age, sex, BMI, region, smoker, children etc.

## Constraints

The Insurance Premium Prediction must be user friendly, as automated as possible and users should not be required to know any of the workings.

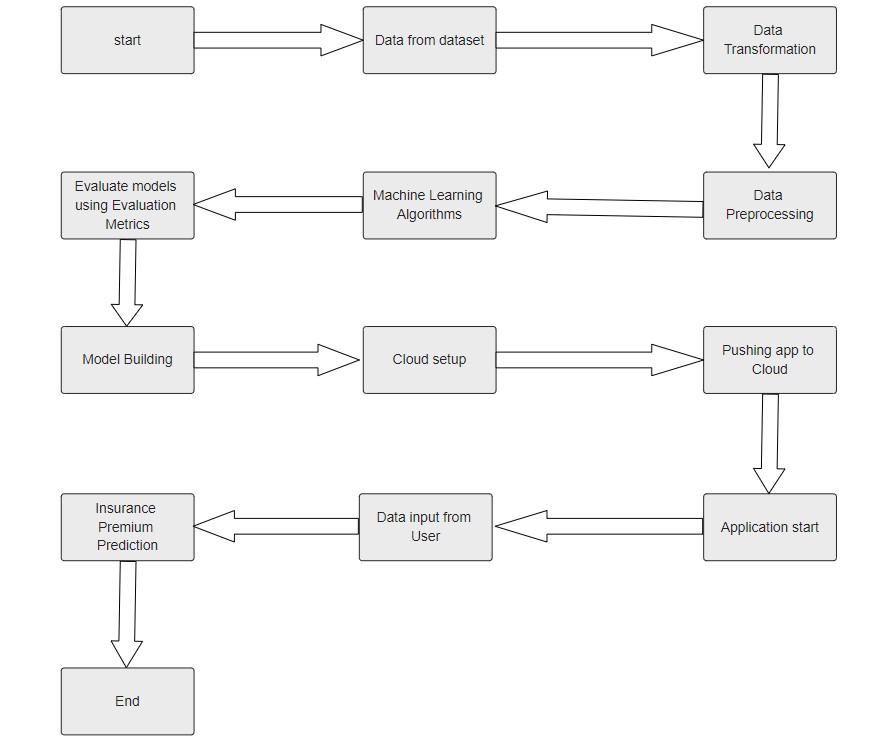
## Risks

Document specific risks that have been identified or that should be considered.

## Out of Scope

Delineate specific activities, capabilities, and items that are out of scope for the project.

**2. Architecture**



# 2.1 Architecture Description

This project is completely based on the life cycle of machine learning, where we will be predicting the insurance premium. The tools used in this project are Python, Pandas, NumPy, Matplotlib, Seaborn, Scikit learn etc. For the Version Control system Git was used and for deployment Heroku was used.

**2.2 Data Requirement**

Whenever we are working on any project the data is completely dependent on the requirement of the problem statement. For this project the problem statement was to create a Hyper tuned Regression machine learning model which can predict the insurance premium

**2.3 Data Collection**

The data which is used in this project was taken from Kaggle.

Dataset link: [*https://www.kaggle.com/datasets/noordeen/insurance-premium- prediction*](https://www.kaggle.com/datasets/noordeen/insurance-premium-%20prediction)

## Data Description

## The primary source of data for this project from Kaggle. The above dataset is comprised of 1338 records with 7 attributes (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 value designated for each level. The data is in structured format and stored in a CSV file.

The dataset separated into two-part the first part called training data, and the second called test data; training data makes up about 80 percent of the total data used, and the rest 20 percent for test data The training data set is applied to build a model as a predictor of medical insurance cost year and the test set will use to evaluate the regression model. The following table shows the Description of the Dataset.

|  |  |
| --- | --- |
| Name | Description |
| Age | Age of the client |
| BMI | Body mass index |
| children | Number of children the client have |
| sex | Male / Female |
| Smoker | Weather the client is smoker or not |
| Region | Where the client live southwest, southeast, northwest, northeast. |
| expenses | The total premium paid by the client |

## 2.5 Data Transformation

The Transformation Process involves converting the data into a format that is suitable for analysis. Here, we will convert the categorical features into numerical features.

## 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.

## Data Pre-processing

Data preprocessing is a crucial process in data analysis that involves transforming raw data into a format that is suitable for analysis. The goal of data preprocessing is to ensure that data is clean, consistent and accurate so that it can be easily analyzed and interpreted. This could include dealing with null values, missing values, correcting typos or other errors in the data, and dealing with outliers.

## 2.8 Machine Learning Algorithms

We predict the values using Machine Learning algorithms including Linear Regression, Polynomial Regression, Ridge Regression, Decision Tree Regression, Random Forest Regressor.

## 2.9 Evaluation Metrics

We check the performance of each Machine Learning model using the standard evaluation Metrics: R2 score, k-fold Cross Validation and Root Mean Square Error value.

## 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 t he best score. The model is then serialized (pickled) .

## 2.11 Data from User

Here we will collect data from user by prompting for numerical features and giving options for categorical features.

## 2.12 Data Validation

Here Data Validation will be done on the test set.

## 2.13 User data feeding into Machine Learning model

The data collected from the User is input into the Machine Learning model.

## 2.14 Insurance Premium Prediction

The machine learning model operates on the input data and provides a prediction for that data.

## 2.15 Deployment

We will be deploying the model to Heroku platform.



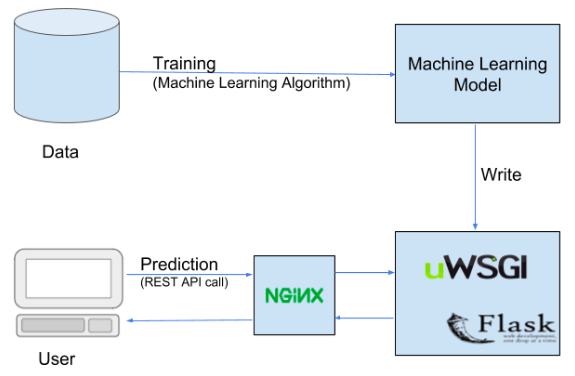
## Database

## System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User gives required information.

2. The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL. Here we use MySQL.

**4. Model training/validation workflow**



# User I/O workflow

Start

Prediction

# 6. 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   1. 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 |
|  |  |  |