|  |
| --- |
| Himanshu Banodha  June 26, 2024 |

|  |  |
| --- | --- |
| iNeuron.ai | |
| Insurance Premium Prediction |
| **Low Level Design (LLD) Documentation** | |

# Document Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Date Issued | Version | Description | Author |
| 26/06/2024 | 1 | Initial LLD – V1.0 | Himanshu Banodha |
|  |  |  |  |

Table of Contents

[Document Version Control 1](#_Toc168061100)

[1. Introduction 3](#_Toc168061101)

[1.1. Why this Low-Level Design Document? 3](#_Toc168061102)

[1.2. Scope 3](#_Toc168061103)

[1.3. Risk 3](#_Toc168061104)

[1.4. Out of Scope 3](#_Toc168061105)

[2. Technical Specifications 3](#_Toc168061106)

[2.1. Dataset Information 3](#_Toc168061107)

[3. Technology Stack 3](#_Toc168061108)

[4. Architecture Description 4](#_Toc168061109)

[4.1. Data description: 4](#_Toc168061110)

[4.2. Data Preprocessing: 4](#_Toc168061111)

[4.3. Exploratory Data Analysis: 4](#_Toc168061112)

[4.4. Data Ingestion: 4](#_Toc168061113)

[4.5. Data Transformation: 4](#_Toc168061114)

[4.6. Model Trainer: 4](#_Toc168061115)

[4.7. Prediction: 4](#_Toc168061116)

[4.8. Saving the Model: 4](#_Toc168061117)

[4.9. Deploy In Localhost: 4](#_Toc168061118)

# Introduction

## Why this Low-Level Design Document?

The purpose of this document is to present a detailed description of the credit card default system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, 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.

## Scope

This software system will be a web application, and this system will be designed to predict whether the credit card holder will default the payment in the upcoming month or not.

## Risk

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

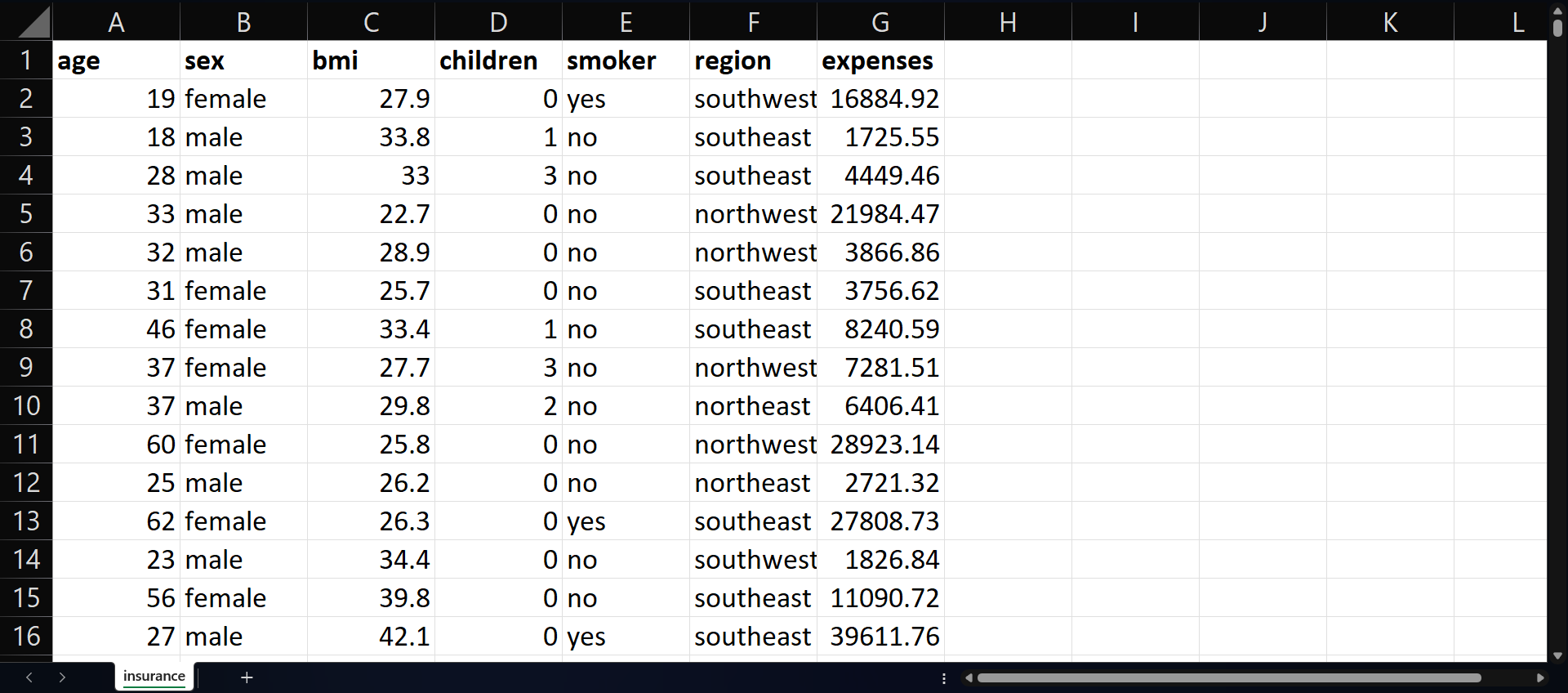
## Out of Scope

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

# Technical Specifications

## Dataset Information

The insurance.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 value designated for each level.



# Technology Stack

|  |  |
| --- | --- |
| Front end | HTML/CSS |
| Back end | Flask |

# Architecture Description

## Data description:

The Dataset was taken from Kaggle (<https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction/data>),

This dataset contains information about age, sex, bmi, children, smoker, region, expenses. It has 1338 observations (rows) & 7 features (columns).

## Data Preprocessing:

In this step we will import the necessary Python libraries such as Pandas, NumPy, Matplotlib, Seaborn, Scikit-learn etc.

And importing the dataset as pandas DataFrame.

## Exploratory Data Analysis:

In this step we handled null values, changed the columns names, plotted multiple graphs & charts in Seaborn and Matplotlib to understand the data properly and also the distribution of the data.

As there were no missing values in the data so we proceed with the visualization and analysis. For each specific feature, by analysing the data we got to know about some key points which can impact the final predictions.

## Data Ingestion:

In this step, we divided the data into 3 CSV files, raw.csv, train.csv & test.csv. with the help of Train Test Split, we divided the data into train and test set, in the ratio of 80-20%, where 80% data got for training the model(train.csv) and 20% is for testing the model(test.csv).

## Data Transformation:

In this step, we performed feature scaling using scikit-learn.

First, we divided the both train & test dataset into 2 categories, categorical data & numerical data. Then we apply the scaling by using the fit-transformed method. Also, we have read the train and test data and changed them into arrays. Then saved this as preprocessor.pkl file for further steps.

## Model Trainer:

In this step, we train the model using multiple algorithms and find the best algorithm with highest accuracy. We used Linear Regression, Decision Tree, Random Forest, Gradient Boosting etc. algorithms to train the model.

## Prediction:

Gradient Boosting Regressor got the highest accuracy score 88.11

## Saving the Model:

Here we saved the model using pickle library, which

## Deploy In Localhost:

We have created an HTML template and deployed the model using Flask