iNeuron.ai

Insurance Premium Prediction

Low Level Design (LLD) Documentation

Himanshu Banodha

6-26-2024

Document Version Control

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

Table of Contents

1. Introduction

1.1. 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.

1.2. 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.

1.3. Risk

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

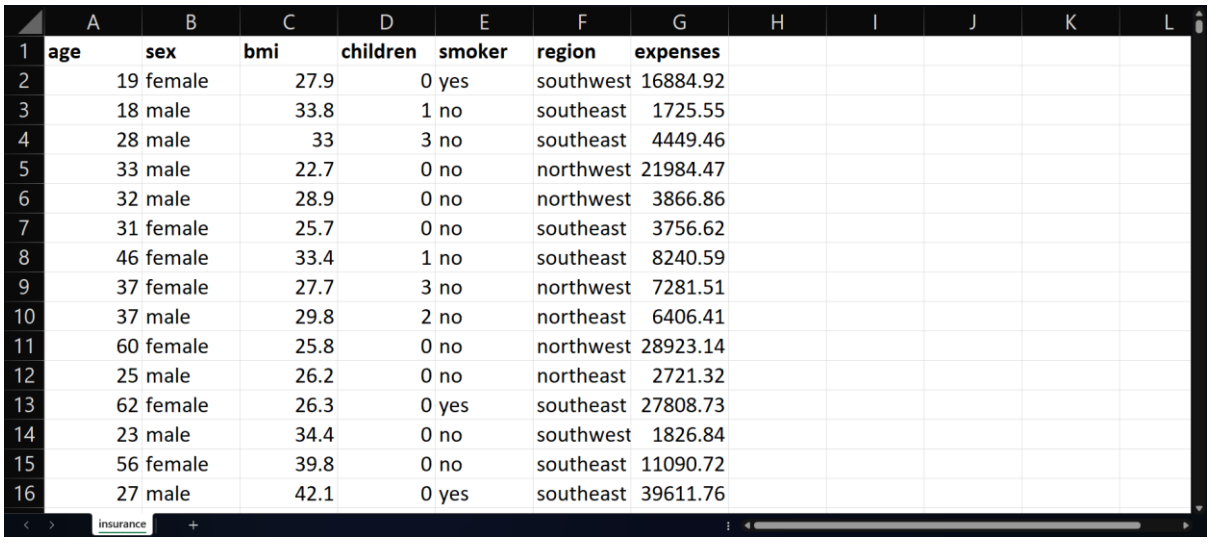
1.4. Out of Scope

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

2. Technical Specifications

2.1. 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.



	A	B	C	D	E	F	G	H	I	J	K	L
1	age	sex	bmi	children	smoker	region	expenses					
2	19	female	27.9	0	yes	southwest	16884.92					
3	18	male	33.8	1	no	southeast	1725.55					
4	28	male	33	3	no	southeast	4449.46					
5	33	male	22.7	0	no	northwest	21984.47					
6	32	male	28.9	0	no	northwest	3866.86					
7	31	female	25.7	0	no	southeast	3756.62					
8	46	female	33.4	1	no	southeast	8240.59					
9	37	female	27.7	3	no	northwest	7281.51					
10	37	male	29.8	2	no	northeast	6406.41					
11	60	female	25.8	0	no	northwest	28923.14					
12	25	male	26.2	0	no	northeast	2721.32					
13	62	female	26.3	0	yes	southeast	27808.73					
14	23	male	34.4	0	no	southwest	1826.84					
15	56	female	39.8	0	no	southeast	11090.72					
16	27	male	42.1	0	yes	southeast	39611.76					

insurance

3. Technology Stack

Front end	HTML/CSS
Back end	Flask

4. Architecture Description

4.1. 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).

4.2. 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.

4.3. 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.

4.4. 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).

4.5. 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.

4.6. 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.

4.7. Prediction:

Gradient Boosting Regressor got the highest accuracy score 88.11

4.8. Saving the Model:

Here we saved the model using pickle library, which

4.9. Deploy In Localhost:

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