Detail Project Report (DPR)

**INSURANCE PREMIUM PREDICTION**

Document Version: 0.3 Last Revised Date: 22-mar-2023

Kanishka Kumar

Document Version Control:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Description |
| 0.1 | 16-01-2023 | Kanishka | Introduction |
| 0.2 | 21-02-2023 | Kanishka | Deployment and  Process |
| 0.3 | 22-03-2023 | Kanishka | Q and A |

**Contents**

1. [Introduction 3](#_TOC_250015)
   1. [What is High-Level design document ? 3](#_TOC_250014)
   2. [Scope 3](#_TOC_250013)
2. [Description 3](#_TOC_250012)
   1. [Problem Perspective 3](#_TOC_250011)
   2. [Problem Statement 3](#_TOC_250010)
   3. [Purposed Solution 3](#_TOC_250009)
   4. [Solution Improvements 3](#_TOC_250008)
   5. [Technical Requirements 4](#_TOC_250007)
   6. [Data Requirements 4](#_TOC_250006)
   7. [Tool Used 4](#_TOC_250005)
   8. [Data Gathering 4](#_TOC_250004)
   9. [Data Description 4](#_TOC_250003)
3. [Data Pre-Processing 4](#_TOC_250002)
4. [Design Flow 5](#_TOC_250001)
   1. [Modelling 5](#_TOC_250000)
   2. Modelling Process 6
   3. Deployment Process 6
5. Data from User 6
6. Data Validation 6
7. Rendering Result 6
8. Conclusion 7

9 Q & A 7

# Introduction

## What is High-Level design document?

The main purpose of this HLD documentation is to feature the required details of the project and supply the outline of the machine learning model and also the written code. This additionally provides the careful description on however the complete project has been designed end-to-end.

## Scope

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

# Description

## Problem Perspective

## The Insurance premium prediction may be a machine learning model that helps to predict the price of the health insurance premium.

## Problem Statement

The goal of this project is to give people an estimate of how much they need based on

their individual health situation. After that, customers can work with any health

insurance carrier and its plans and perks while keeping the projected cost from our

study in mind. This can assist a person in concentrating on the health side of an

insurance policy rather han the ineffective part.

## Purposed Solution

Projected to require the desired input of user from the created interface and method all the provided information to satisfy the wants of the machine learning model and at last show the output oral communication so and then quantity is that the expected value.

## Solution Improvements

we will even predict the premium considering several health issue however, considering from the angle of business, if we have a tendency to method such information and predict the price of the discounted price tag it'll bring some loss to the insurance company. therefore, this technique isn't thought-about.

## Technical Requirements

There are not any hardware needs needed for victimization this application, the user should have AN interactive device that has access to the web and should have the fundamental understanding of providing the input. And for the backend half the server should run all the package that's needed for the process the provided information and to show the results.

## Data Requirements

The info demand is totally supported the matter statement. and also, the information set is accessible on the Kaggle within the type of standout sheet(.xlsx). because the main theme of the project is to induce the expertise of real time issues, we have a tendency to ar once more mercantilism {the information into the prophetess data base and commerce it into csv format.

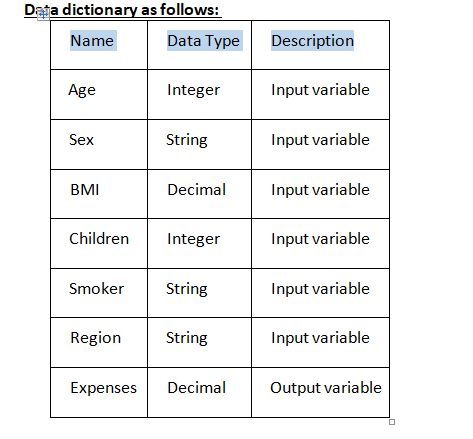
## Tool Used

* Python 3.9 is employed because the programming language and frame works like numpy, pandas, sklearn and alternative modules for building the model.
* PyCharm is employed as IDE.
* For visualizations seaborn and components of matplotlib are getting used.
* For information assortment prophetess info is getting used.
* Front end development is completed by Streamlit.
* GitHub is employed for version management.
* Docker is employed for deployment.

## Data Gathering

The data for the current project is being gathered from Kaggle dataset, the link to the data is: https://www.kaggle.com/noordeen/insurance-premium-prediction

## Data Description



1. Data Pre-processing

Steps performed in pre-processing are:

* First the info sorts square measure being checked and located solely the value column is of sort number.
* Checked for null values as there square measure few null values, those rows square measure born.
* Converted all the desired column into the date time format.
* Performed one-hot cryptography for the desired columns.
* Scaling is performed for needed information.
* And, the info is prepared for passing to the machine learning formula

# Design Flow

## Modelling

The pre-processed data is then visualized and all the required insights are being drawn. Although from the drawn insights, the data is randomly spread but still modelling is performed with different machine learning algorithms to make sure we cover all the possibilities. And finally, as expected random forest regression performed well and further hyperparameter tuning is done to increase the model’s accuracy.

## Modelling and Deployment Process

Import python libraries and read data

Data Processing

EDA

Handling categorical data

Saving Data in Pickle File

Hyper Parameter Tuning

Model Building

Feature Selection

Create a webpage Using Streamlit for Deployment

Deployment on Heroku Cloud

Input Value and Predict Final Result

End

* 1. **Data from User**

The data from the user is retrieved from the created HTML web page.

* 1. **Data Validation**

The data provided by the user is then being processed by app.py file and validated. The validated data is then sent for the prediction.

* 1. **Rendering Result**

The data sent for the prediction is then rendered to the web page.

* 1. **Deployment**

The tested model is then deployed to Docker. So, users can access the project from any internet devices.

1. Conclusion

The flight fare prediction will predict the worth supported the trained knowledge set within the rule. therefore, the user will recognize the approximate value for his or her journey.

1. Q & A

Q1) What’s the source of data?

The data for training is provided by the client in multiple batches and each batch contain multiple files.

Q 2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q 3) What’s the complete flow you followed in this Project?

Refer Page no 6 for better Understanding.

Q 4) After the File validation what you do with incompatible file or files which didn’t pass the

validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q 5) How logs are managed?

We are using different logs as per the steps that we follow in validation and modelling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q 6) What techniques were you using for data pre-processing?

* Removing unwanted attributes
* Visualizing relation of independent variables with each other and output variables
* Checking and changing Distribution of continuous values
* Removing outliers
* Cleaning data and imputing if null values are present.
* Converting categorical data into numeric values.

•

Q 7) How training was done or what models were used?

* Before dividing the data in training and validation set, we performed pre-processing over the data set and made the final dataset.
* As per the dataset training and validation data were divided.
* Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used based on the recall, final model was used on the dataset and we saved that model.

•

Q 8) How Prediction was done?

The testing files are shared by the client. We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q 9) What are the different stages of deployment?

* + First, the scripts are stored on GitHub as a storage interface.
  + The model is first tested in the local environment.
  + After successful testing, it is deployed on Docker.