High Level Design
(HLD)

Insurance Premium Prediction

Document Version Control

Date Issued	Version	Description	Author
28/04/2023	1	Initial HLD — V1.0	Parag Darade Prince Katiyar Kushal Bhadra

Contents

Abstract

In this project, we aim to predict insurance premiums for individuals by analyzing their health data. We employed different model's algorithm to accomplish this. We used these models to compare and contrast their performance.

The training dataset was used to train the model, and the predictions made by the model were compared with actual data to test and verify the model's accuracy. After reaching the accuracy of all the models ie LinearRegression, KNeighborsRegressor, DecisionTreeRegressor, XGBRegressor, AdaBoostRegressor, ExtraTreesRegressor, RandomForestRegressor , **CatBoostRegresso**r, algorithm performed better than the remaining models.

Ultimately, we found that the **CatBoostRegresso**r algorithm was the best suited for this task as it provided the best evaluation score compared to the other models.

1 Introduction

1.1 Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

- Present all of the design aspects and define them in detail
- Describe the user interface being implemented
- Describe the hardware and software interfaces
- Describe the performance requirements
- Include design features and the architecture of the project
- List and describe the non-functional attributes like:
  - Reliability
  - Security
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilization
  - Serviceability

1.2 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.

1.3 Definitions

Term	*Description*
UGV	Unmanned Ground Vehicle
Database	Collection of all the information from MongoDb
IDE	Integrated Development Environment
RENDER	RENDER

2 General Description

2.1 Product Perspective

The Insurance premium estimation is a machine learning-based predictive model which will help us to predict the premium of the person for health insurance.

2.2 Problem Statement

To develop an API interface to predict the premium of insurance using people's individual health data and analyze the following:

- To detect BMI value affects the premium.
- To detect smoking affects the premium of the insurance.
- To create an API interface to predict the premium.

2.3 Proposed Solution

The solution proposed here is to estimate the premium of insurance based on people's health data. This can be implemented to perform the use cases mentioned above. In the first case, we will analyze how the BMI value affects people's health as well as the premium of the insurance. In the second case, if the model detects that smoking affects the premium, we will inform those people. And in the last use case, we will create an interface to predict the premium.

2.4 Technical Requirements

The solution proposed here can be implemented as a cloud-based solution or as an application hosted on an internal server, or even on a local machine. To access this application, the following minimum requirements are necessary:

- A good internet connection.
- A web browser.

For training the model, the following system requirements are preferred:

- 4 GB RAM or more.
- An operating system such as Windows, Linux or Mac.
- Visual Studio Code or Jupyter notebook.

2.5 Data Requirements

The data requirements for this project will depend on the specific problem statement. A CSV file will be used as the input file, and the feature/field names and sequence should be followed as decided. It's important to have a clear understanding of the problem statement and the data that is required to solve it, to design a suitable data pipeline, and to train the model effectively.

2.6 Tools used

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, Flask , Pymongo, and Jinga are used to build the whole model.


- Pandas is an open-source Python package that is widely used for data analysis and machine-learning tasks.
- NumPy is the most commonly used package for scientific computing in Python.
- Plotly is an open-source data visualization library used to create interactive and quality charts/graphs.
- Scikit-learn is used for machine learning.
- Flask is used to build API.
- VS Code is used as an IDE (Integrated Development Environment)
- GitHub is used as a version control system.
- Front-end development is done using HTML and CSS.
- Railway is used for the deployment of the model.

2.7 Constraints

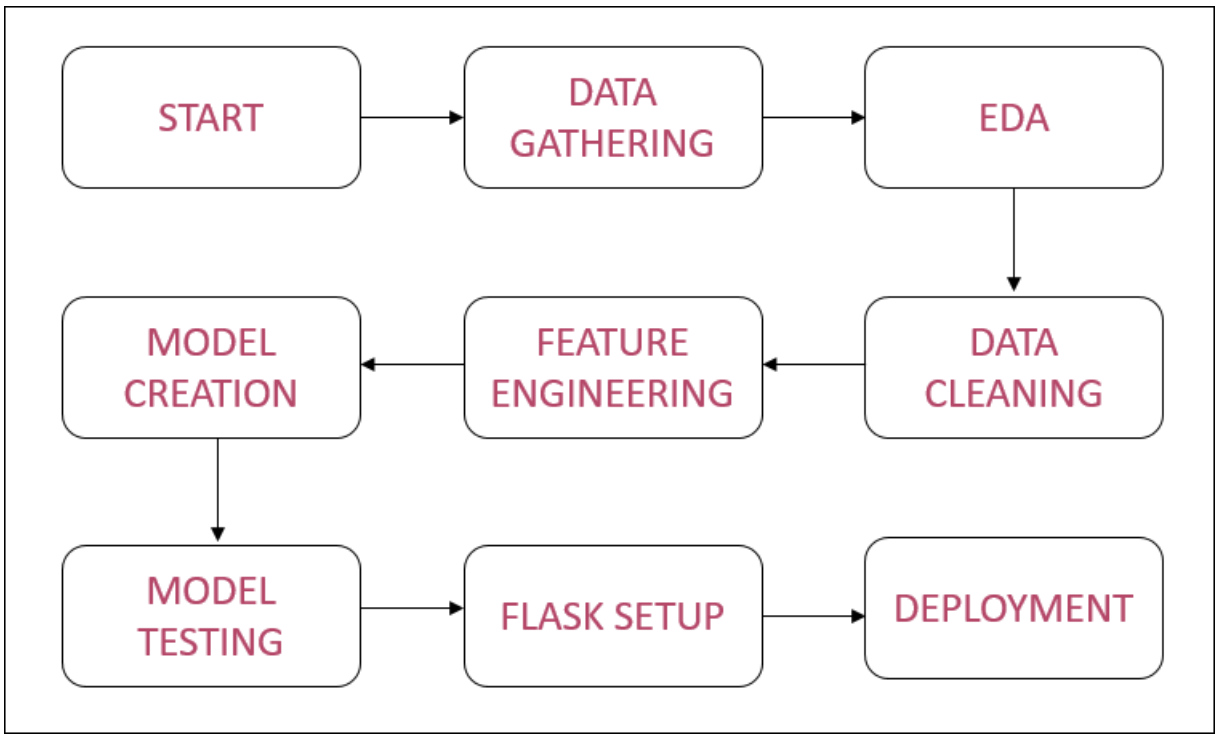
It is useful for the user by predicting Insurance prices based on their provided details for ex: - Bmi, sex, smoker, yes/no, age, etc.

2.8 Assumptions

The main objective of the project is to develop an API to predict the premium for people based on their health information. A machine learning-based regression model is used for predicting the above-mentioned cases on the input data.

3 Design Details

3.1 Process Flow



3.2 Event log

The system should log every event so that the user will know what process is runninginternally.

Initial Step-By-Step Description:

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developer can choose logging method. You can choose database logging / File logging as well.
- System should not hang even after using so many loggings.

3.3 Error Handling

Should errors be encountered, an explanation will be displayed as to what went wrong? Anerror will be defined as anything that falls outside the normal and intended usage.

4 Performance.

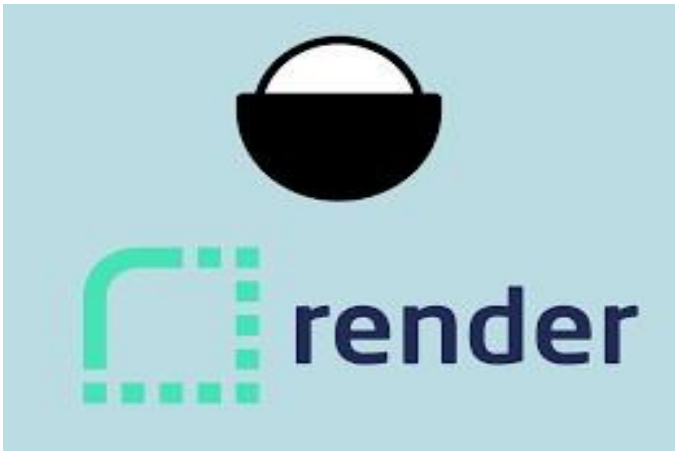
4.1 Reusability

The entire solution will be done in a modular fashion and will be API oriented. So, in the case of scaling the application, the components are completely reusable.

4.2 Application Compatibility

The interaction with the application is done through the designed user interface, which the end user can access through any web browser.

4.3 Deployment



5 Conclusion

This system shows us the different techniques that are used to estimate the how much amount of premium required based on individual health situations. After analysis, it shows how a smoker and non-smokers affect the amount of estimate. Also, a significant difference between male and female expenses. Accuracy plays a key role in prediction-based systems. From the results, we could see that Gradient Boosting turned out to be the best working model for this problem in terms of accuracy. Our predictions help users to know how much amount premium they need based on their current health situation.