**Architecture Design**

**Insurance Premium Prediction**

Revision 1.0.1

Date: 17/11/2023

Description: Initial revision HLD

**Document Version Control**

* **Reviewers:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Comments** | **Reviewers** |
| 17/11/2023 | 1.0.1 | Initial revision (1.0.1) | Robin Mitra |
|  |  |  |  |
|  |  |  |  |

* **Change Information:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Revision** | **Description** | **Author** |
| 17/11/2023 | 1.0.1 | Initial revision (1.0.1) | Robin Mitra |
|  |  |  |  |
|  |  |  |  |

**Contents**

Document Version Control…………………………………………………………………………………………………………… 2

Abstract……………………………………………………………………………………………………………………………………….. 4

1 Introduction…………………………………………………………………………………………………………………….. 5

1.1 Why this Architecture Design Document?............................................................... 5

1.2 Definitions………………………………………………………………………………………………………….. 5

2 Technical specification ……………………………………………………………………………………………………..5

3.1 Dataset overview………………………………………………………………………………………………… 5

3.2 Predicting the Insurance premium…………………………..…………………………………………. 5

3.3 Logging………………………………………………………..……………………………………………………… 6

3 Technology Stack……………………………………………………………………………………………………….……..6

4 Proposed Solution…………………………………………………………………………………………………..………..6

5 Workflow……………………………………………………………………………………………………………………..…..7

6 Key Performance indicators (KPI)…………………………………………………………………………….………..8

**Abstract**

An insurance premium is the regular payment made by an individual or entity to an insurance company in exchange for coverage. The cost of the premium is calculated based on a variety of factors, including the type of insurance (such as auto, health, life, or property), the level of coverage, and the risk profile of the policyholder (such as age, health status, or driving history). These payments allow the insurer to maintain financial reserves to fulfill its contractual obligations when policyholders make claims. Understanding how premiums are determined is crucial for individuals and businesses to select appropriate coverage and manage their financial protection efficiently.

**1** **Introduction**

**1.1** **Why this architecture Design Document?**

The goal of architecture Design Document is to give an internal logical design of the actual program code for the Insurance Premium Prediction System. Architecture Design Document describes the class diagrams with the methods and relations between classes and program specs. It describes the modules so that the programmer can directly code the program from the document.

**1.2 Definitions**

|  |  |
| --- | --- |
| Term | Description |
| Database | Collection of all the information monitored by this system |
| IDE | Integrated Development Environment |
| EDA | Exploratory Data Analysis |

**2** **Technical Specifications**

**2.1 Dataset Overview**

For training and testing the model, I used the public data set available in Kaggle, “Insurance Premium Prediction” by nursnaaz

URL: <https://www.kaggle.com/noordeen/insurance-premium-prediction>

Data dictionary as follows:

|  |  |  |
| --- | --- | --- |
| **Name** | **Data Type** | **Description** |
| Age | Integer | Input variable |
| Sex | String | Input variable |
| BMI | Decimal | Input variable |
| Children | Integer | Input variable |
| Smoker | String | Input variable |
| Region | String | Input variable |
| Expenses | Decimal | Output variable |

**2.2 Predicting the Insurance Premium**

• The web application must be loaded properly for the users without any technical glitches like server timeouts.

• It must display the input fields and the “Predict” button to the users who accessed the application and allow the user to enter the values with respect to the personal information.

• The user gives the required information.

• Then the application should be able to predict the insurance premium based on the information given by the user.

**3.3 Logging**

We should be able to log every activity done by the user.

• The system should be able to log every step in the program flow.

• System should not be hung even after using so many loggings.

• Logging makes debugging much easier, like we can directly go to

that specific line of code, having bugs.

• In this project, logs will be written in the files “development\_logs.log” and the “deployment\_logs.log”

respectively.

**3** **Technology Stack**

|  |  |
| --- | --- |
| **Front-End** | **HTML 5** |
| **Back-End** | **Python version 3.8**  **Flask version 3.0.0** |

**4** **Proposed solution**

The solution proposed here is a web application, which takes the details of the personal information which contributes to insurance premium and those details will be taken by an Xgboost regressor model in the backend, which predicts the premium in dollars and displays in the front-end page to the user.

**5** **Workflow**

**6** **Key Performance indicators (KPI)**

• Time and workload reduction using the regression models.

• Comparison of the R2 scores and the Adjusted R2 scores of the model on both the training and the testing data.

• Comparison of the RMSE scores of the model on both the training and the testing data.

• Feature importance using Gradient Boost regressor: