Architecture

Insurance Premium Prediction

Revision Number: 2.0
Last date of revision: 09/03/2022

Subhankar Roy

Contents

Abstract

Penetration of Insurance facility is very less till date in the world. People are unaware of the fact that they can insure their earning member's life and health by buying life and health insurance. For more awareness of insurance facility, many companies hired agents to go and make people aware in various corners of the world. In this system, the agents sometimes try to sell the most highly premium insurance products to the customers. It becomes very difficult for the people to buy such highly priced life and health insurances. The insurances are based on certain factors like age, sex, medical situation of the customer. Depending on various such factors, the premium of the insurance policy is fixed. Our prediction model will predict the exact premium amount one needs to pay to buy a life and health insurance.

1 Introduction

1.1 Why this Architecture Document?

The purpose of this document is to present a detailed description of the Insurance Premium Prediction. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. 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.

The main objective of the project is to predict what is the exact insurance premium one has to pay and act accordingly.

To create an Al solution for predicting insurance premium and to implement the following use cases.

- To predict premium based on age, sex and bmi
- To predict premium based on whether the customer is a smoker or not
- To predict premium based on number of children and region of the customer.

1.2 Scope

Low-level design (LLD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software architecture, source code and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work

1.3 Constraints

We will only be selecting a few of the features.

1.4 Risks

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

1.5 Out of Scope

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

2 Technical specifications

2.1 Dataset

Features	Finalized	Source
Age	yes	https://www.kaggle.com/noordeen/insurance-premium-prediction
Sex	Yes	
Gender	Yes	
Children, smoker and region	Yes	

2.1.1 Insurance dataset overview

The purposes of this exercise to look into different features to observe their relationship, and plot a multiple linear regression based on several features of individual such as age, physical/family condition and location against their existing medical expense to be used for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium

There are a total of 1338 observations in the training set.

- Feature table
-

	age	sex	bmi	children	smoker	region
0	19	0	27.9	0	1	1
1	18	1	33.8	1	0	2
2	28	1	33.0	3	0	2
3	33	1	22.7	0	0	3
4	32	1	28.9	0	0	3
...	...	...	...	...	...	...
1333	50	1	31.0	3	0	3
1334	18	0	31.9	0	0	4
1335	18	0	36.9	0	0	2
1336	21	0	25.8	0	0	1
1337	61	0	29.1	0	1	3

1338 rows × 6 columns

- Expense table

```
0        16884.92
1         1725.55
2         4449.46
3        21984.47
4         3866.86
           ...
1333     10600.55
1334      2205.98
1335      1629.83
1336      2007.95
1337     29141.36
Name: expenses, Length: 1338, dtype: float64
```

2.2 Predicting Premium

- The system displays the choices of the factors.
- The User puts values in the respected places.
- The system presents the set of inputs required from the user.
- The user gives required information.
- The system should be able to predict the premium amount based on the user information.

2.3 Logging

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

- The System identifies at what step logging required
- The System should be able to log each and every system flow.
- Developers can choose logging methods. You can choose database logging/ File logging as well.
- System should not be hung even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.

2.4 Database

System needs to store every request into the database and we need to store it in such a way that it is easy to retrain the model as well.

1. The User gives required information.

2. The system stores each and every data given by the user or received on request to the database. Database you can choose your own choice whether MongoDB/ MySQL.

2.5 Deployment

1. Google Cloud Platform

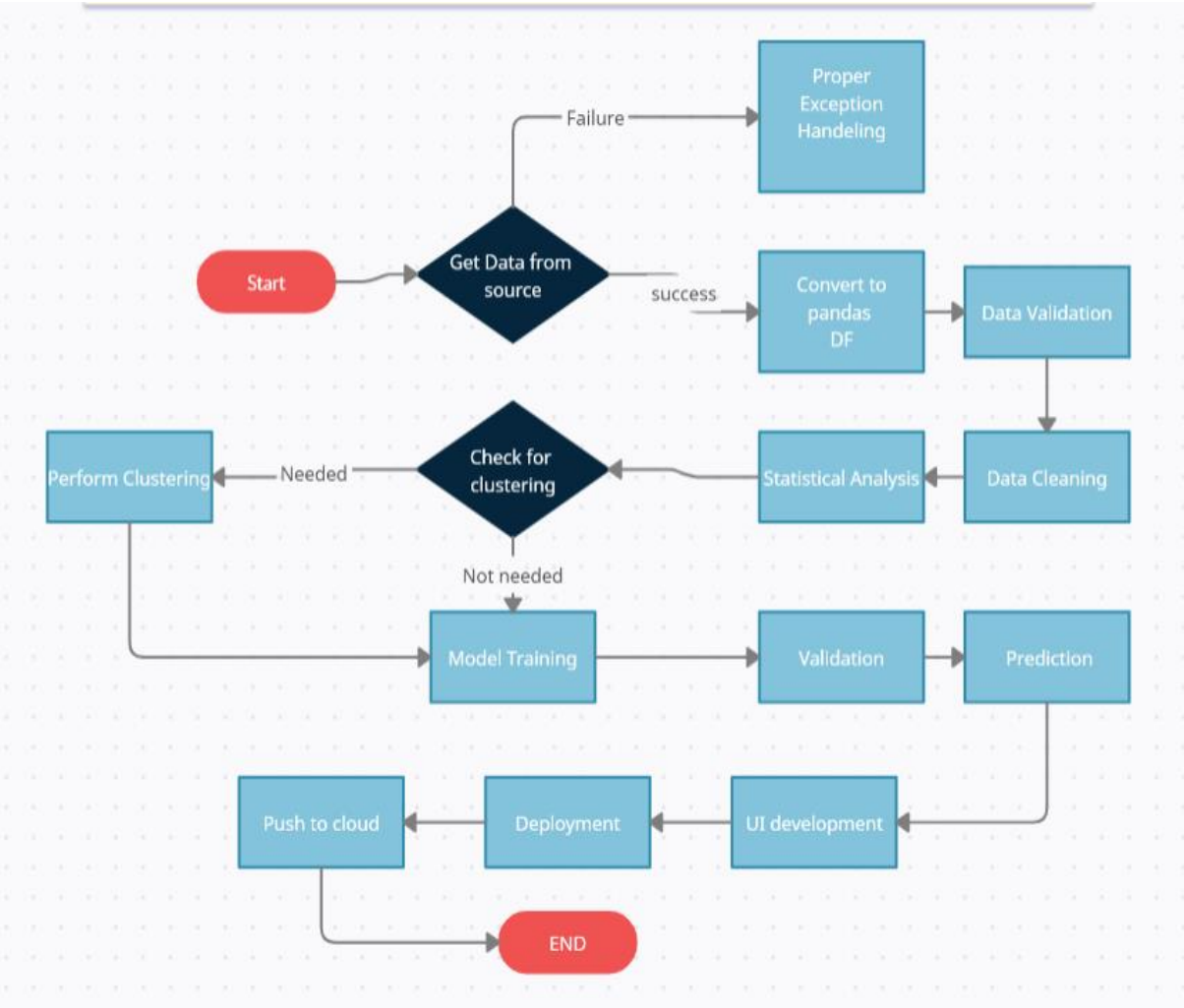
3 Technology stack

Front End	HTML
Backend	Python Streamlit
Database	Cassandra
Deployment	GCP

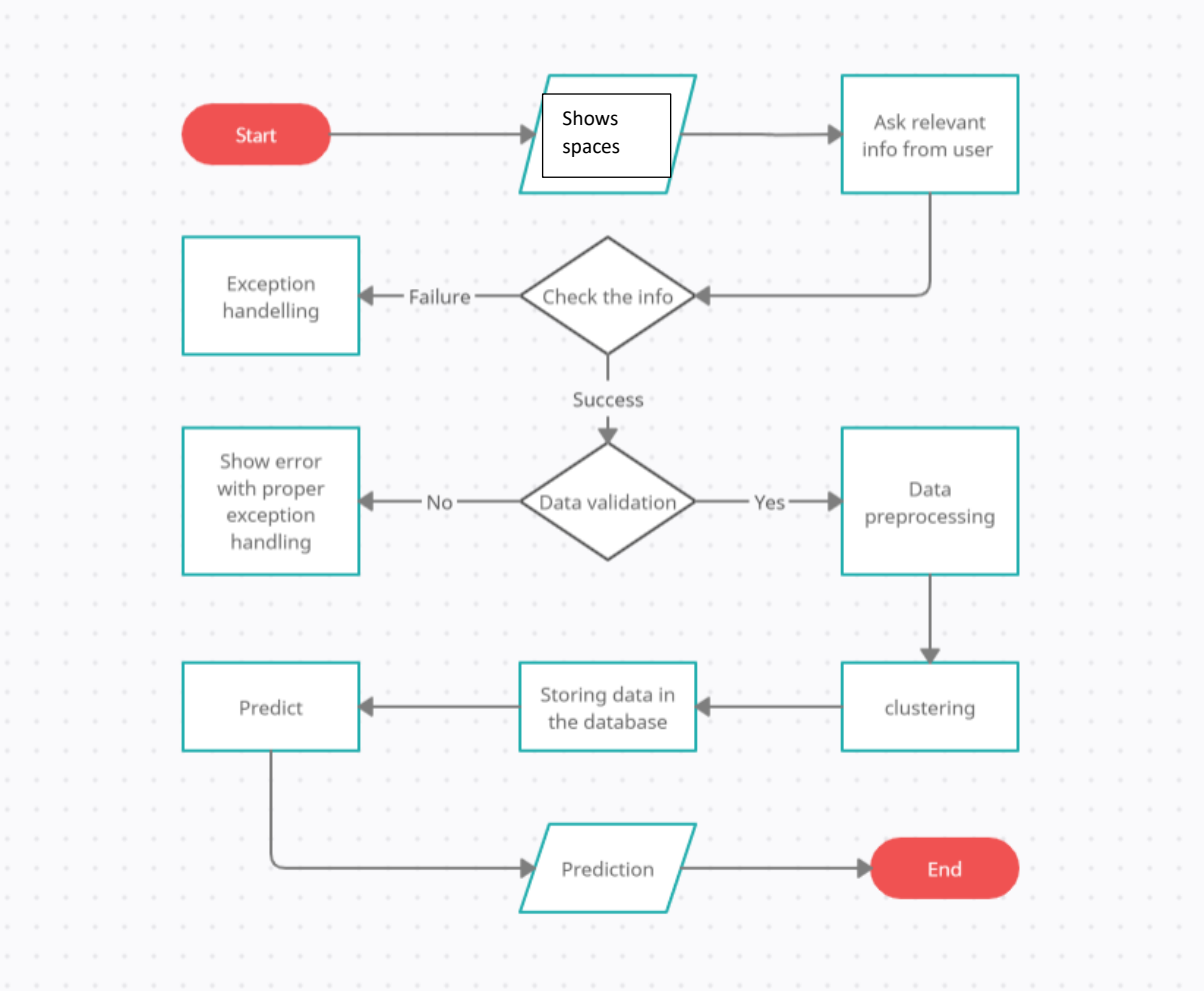
4 Proposed Solution

The solution proposed here is based on customer's age, sex, bmi, children, smoker and region. If the person is young then his premium will be lesser compared to a person who is old. Smoking also increases the premium rate. More a person is healthy, less will be the insurance premium.

5 Model training/validation workflow



6 User I/O workflow



7 Test cases

Test case	Steps to perform test case	Module	Pass/Fail