High Level Design (HLD)

INSURANCE PREMIUM PREDICTION USING ML ALGORITHMs

Revision Number : 3.0

Last date of Revision : 12/08/2022

Ankith Patil

Document Version Control

Date Issued	Version	Description	Author
08/08/2022	1	Initial HLD -V1.0	Ankith Patil
09/08/2022	2	Updated API -V2.1	Ankith Patil
12/08/2022	3	Final HLD -V2.0	Ankith Patil

1. CONTENTS

Document Version Control 2
Abstract 4

1. Introduction 5

1.1. Why this high-Level design document? 5
1.2. Scope 5

2. General Description 6

2.1. Product Perspective 6
2.2. Problem Statement 6
2.3. Proposed Solution 6
2.4. Further improvements 6
2.5. Data Requirements 7
2.6. Tools Used 7

3. Design Details 8

3.1. Process Flow 8
3.1.1. Model Training & Evaluation 8
3.1.2. Deployment Process 9
3.2. Event log 9
3.3. Error Handling 10
3.4. Performance 10
3.5. Reusability 10
3.6. Application Compatibility 11
3.7. Resource Utilisation 11
3.8. Deployment 11

4. Dashboard 12

5. Conclusion 13

2. Abstract

Traditionally most insurance companies **employ actuaries** to calculate the insurance premiums. Actuaries are business professionals who **use mathematics** and statistics to assess the **risk of financial loss** and **predict the likelihood of an insurance premium** and claim, based on the factors/features like age and gender, etc.

They typically produce something called an actuarial table provided to an insurance company's underwriting department, which uses the input to set insurance premiums. The insurance company calculates and writes all the programs, but it becomes **much simpler** by **using Machine Learning to predict the insurance premium** and even for the **end consumer to predict the insurance premiums to manage their insurance cover** and also **their monthly expenses**.

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:
  - Security
  - Reliability
  - Maintainability
  - Portability
  - Reusability
  - Application compatibility
  - Resource utilisation
  - 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
IDE	Integrated development environment

2. General Description

2.1. Product perspective

An insurance price depends on various features such as **age**, type of coverage, amount of coverage needed, **gender, body mass index** (**BMI**), **region**, and other special factors like **smoking** to determine the price of the insurance.

Insurance premium prediction model is a machine learning based model which will help us predict or estimate the cost/expenses of the insurance policies one buys and the subsequent premium one has to pay to get themselves insured.

2.2. Problem statement

To create an **AI solution to predict the insurance premium** to fulfil the following tasks:

- To give people an estimate of how much they need based on their individual health situation.
- Customers can work with any health insurance carrier and its plans and perks while keeping the projected cost from our study in mind.

2.3. Proposed Solution

The insurance premium predictor model estimates the insurance premium one has to shell out based on their **particular background** and their **personal details**.

This aids the end consumer to predict the expenses he/she has to bear on the insurance premium and also helps the insurance companies to make their insurance **premiums predictable, scientific and rule based**. This can assist a person in

concentrating on the health side of an insurance policy rather than the ineffective part.

Based on factors which really matter - Gender, Age , BMI , Region , Smoker/Non-smoker , number of children.

2.4. Further improvements

This model can be further scaled to estimate the spectrum of the people mostly requiring which type of **insurance cover** and also **can aid the Governments** to estimate which types of insurance cover with minimal insurance premiums can **provide maximum benefit to wide sections of the society.**

This can also be used to **increase the insurance coverage** and **reduce the insurance premium** that the insurance company's charges as we scientifically based on the background of the insurance applicants can create a **tailormade insurance policies** best suited for that particular use cases becomes more **intelligent** and **smarter** in predicting the exact price

2.5. Data Requirements

Data requirements completely depends on the problem statement

- We need to create the **schema file** if not provided from the client (showing the number of **numerical columns** and **categorical columns**)
- The **naming convention** followed for naming the training and testing data files
- The **minimum acceptable accuracy parameter** for the model
- The **minimum acceptable level of difference of the train and test model accuracy**
- Parameters to populate in case there are multiple null values
- The tolerance level of the collinearity between the features.

2.6. Tools Used

- **VS code** is used as IDE.
- For visualisation of the plots, **Matplotlib**, **Seaborn** and **Plotly** are used.
- **Heroku** is used for deployment of the model
- Front end development is done using **HTML/CSS**
- **Python** is used for backend development
- **Github** is used as version control system.

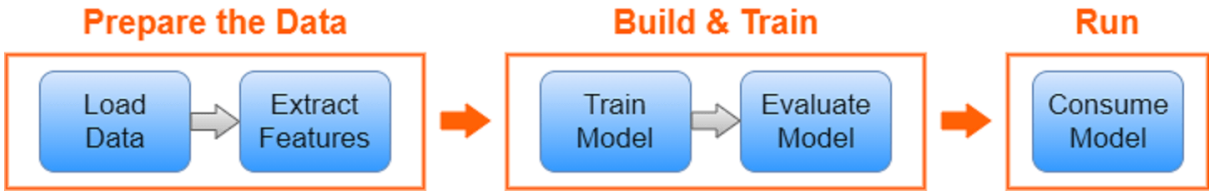
3. Design Details

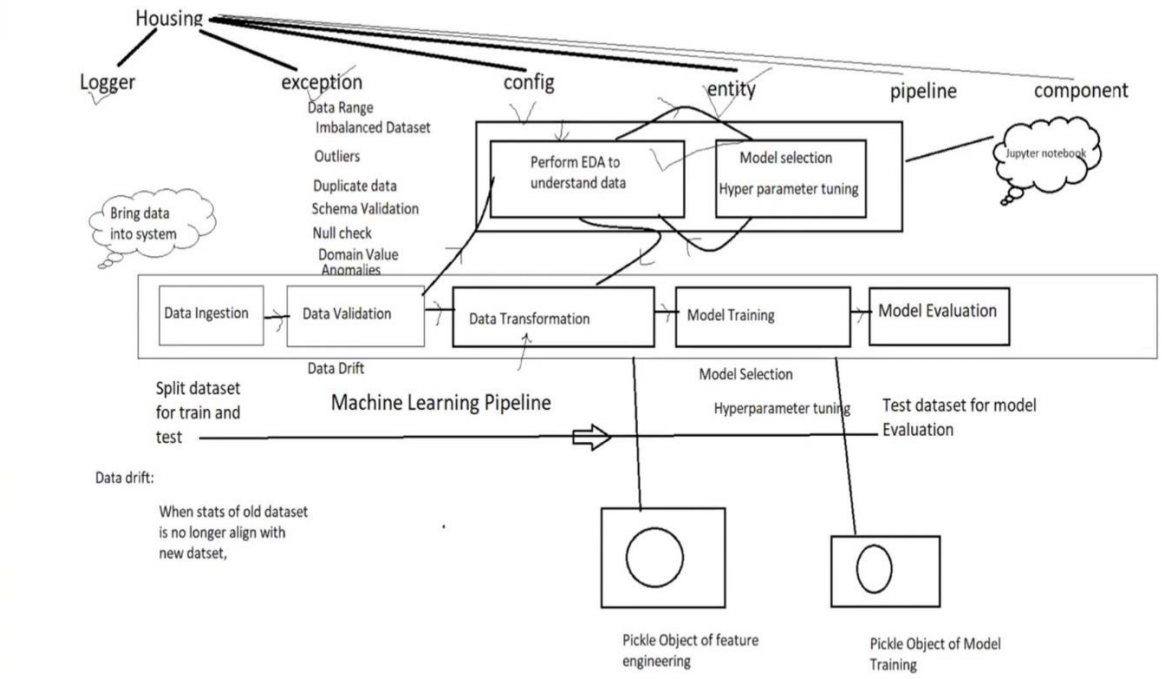
There are three key tenants of ML workflow:

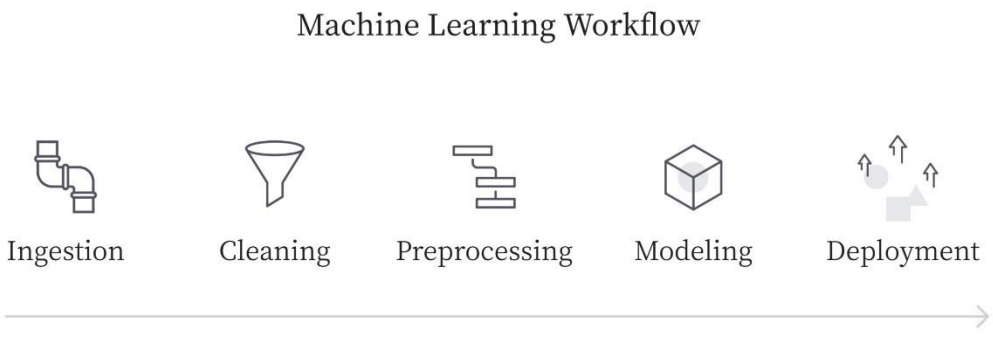
- **Prepare the data**. **Load the data** from the database or CSV files. Extract/Identify the key features (input and output parameters) relevant to the problem you will solve or predict the outcome.
- **Build and train ML model**. Here you can evaluate different algorithms, settings and see which model is best for your scenario.
- Once the model is ready, **consume the model in your application**

age	sex	bmi	children	smoker	region	price
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.471
32	male	28.88	0	no	northwest	3866.8552

Features (Input) **Label (Output)**


Prepare the Data
Load Data
Extract Features
Build & Train
Train Model
Evaluate Model
Run
Consume Model


Housing
Logger
exception
Data Range
Imbalanced Dataset
Outliers
Duplicate data
Schema Validation
Null check
Domain Value Anomalies
config
entity
pipeline
component
Perform EDA to understand data
Model selection
Hyper parameter tuning
Jupyter notebook
Bring data into system
Data Ingestion
Data Validation
Data Transformation
Model Training
Model Evaluation
Data Drift
Model Selection
Split dataset for train and test
Machine Learning Pipeline
Hyperparameter tuning
Test dataset for model Evaluation
Data drift:
When stats of old dataset is no longer align with new datset,
Pickle Object of feature engineering
Pickle Object of Model Training


Machine Learning Workflow
Ingestion
Cleaning
Preprocessing
Modeling
Deployment

3.2. Event Log

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

We have created our own **custom logging** by informing the user about what the **timestamp** and **message** we want to convey at what point of the programme flow.

3.3. Error Handling

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

We have created a **custom exception class** wherein we have systematically created the format in which the system **displays the location of the error** along with **exact line** and **file location** and the exact **type of error**. This creates the error response which is very **reader friendly** and would also streamline the **debugging process** the code for future references easy.

4. Performance

The insurance premium prediction **AI based solution** helps in creating a fairly accurate model with less difference in the training and test data accuracies. As the expenses of the people are involved and it involves fairly complex mathematical calculations which is traditionally calculated.

The ML based model should come close to the actual calculations and have **minimal misjudgements, not have over-fitting or underfitting issues**, as this jeopardises the estimate insurance companies and the insurance applicants predict.

Also, constant check of **data drift** is necessary to take into **account the changes in the lifestyles of the people** and reinforce the believe in the weightage of the present features in the final prediction of the insurance premium prediction.

Thus, there is need to have **constant re-training of the model to achieve near zero error** in the estimate prediction.

4.1. Reusability

The code written and the components used have **ability to reuse** as the code is written in a **modular fashion in OOPs**. The pipeline is created in such a fashion that streamlines the whole process and is **flexible** to new changes and also is a **scalable** to **tackle new challenges**.

4.2. Application Compatibility

The different components for this project will be using **Python** as an **interface** between them. Each Component will have its own task to perform, and it is the job of the Python to ensure **proper transfer of information**.

4.3. Resource Utilisation

We have used the **threading technique** to ensure the application uses the processing power available in the most efficient manner. Thus, we have programmed the threading process in a way that a single heavy process does not hamper the **execution of the simultaneous tasks** at hand.

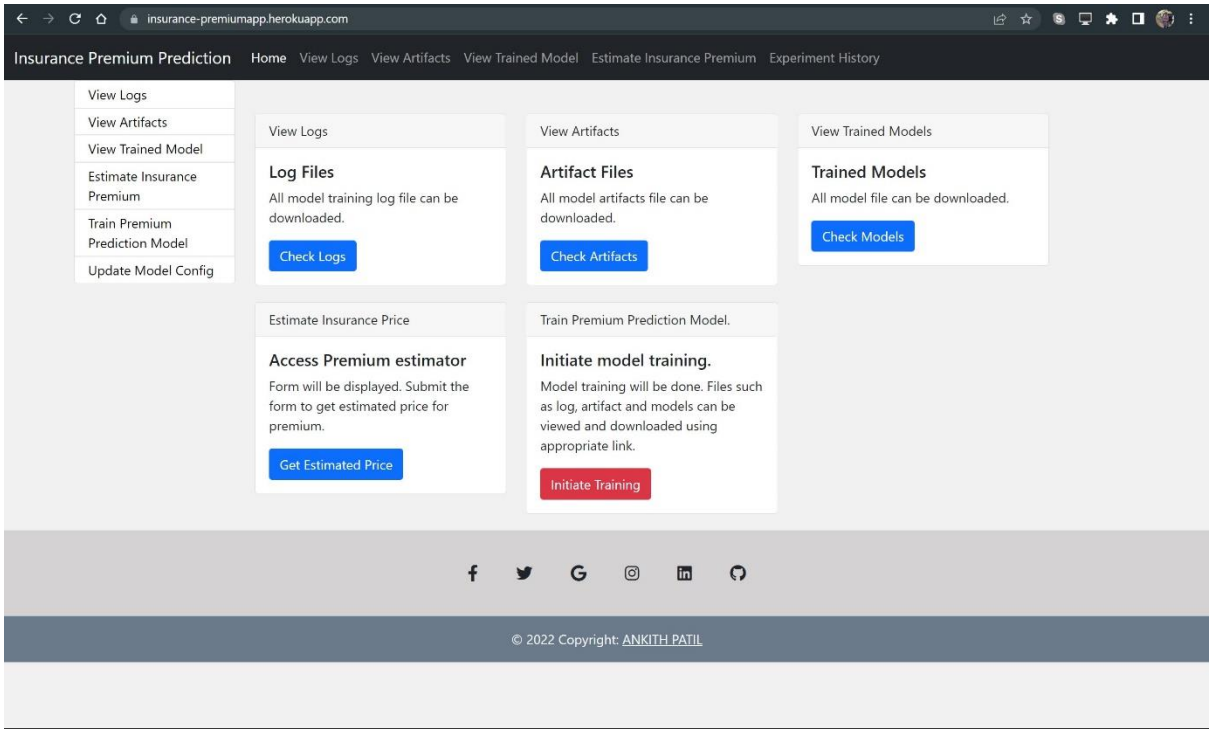
4.4. Deployment

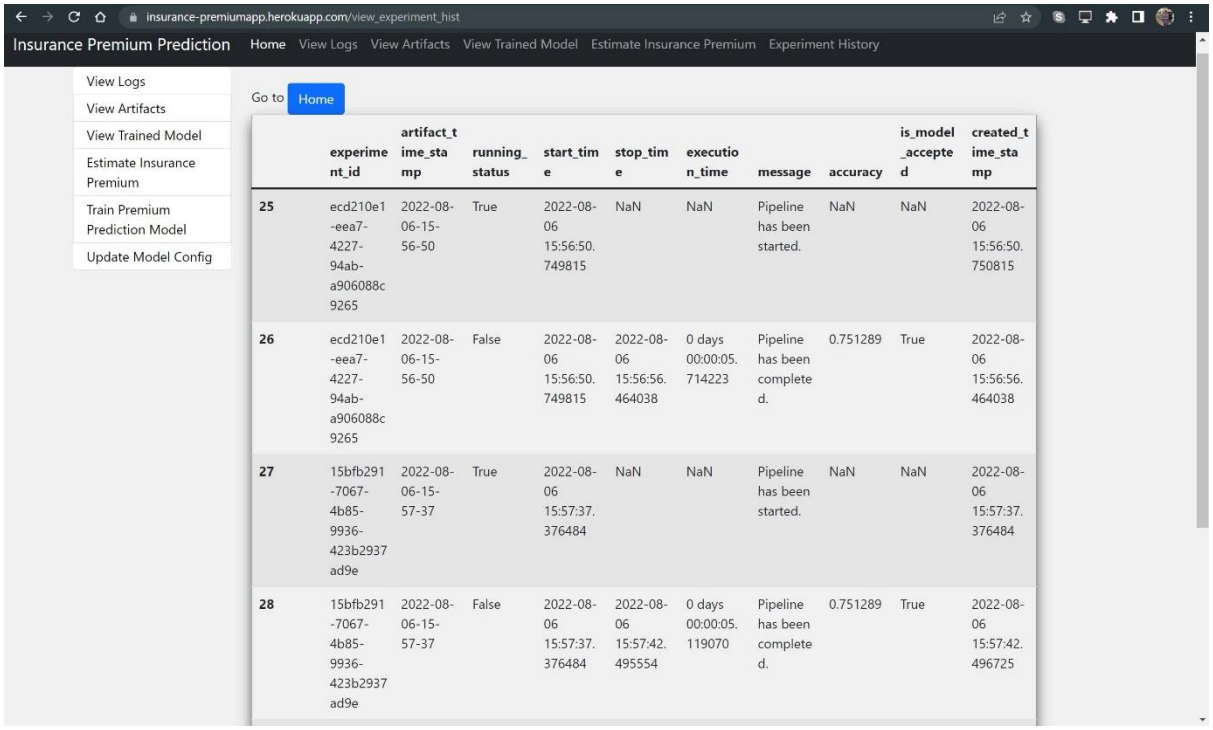
We have built a **CI-CD pipeline** wherein when a defined **trigger** (when we push the code to the github repository) Is triggered, the code is automatically deployed over the cloud taking into account the changes along with it.

This creates a automated CI-CD pipeline, which is **scalable** also increase the **productivity** of the application to accommodate any changes without major disruptions.

5. Dashboard

Dashboard will be displayed to ensure that the application can be user friendly in accessing different objectives from the application.





	experiment_id	artifact_time_stamp	running_status	start_time	stop_time	execution_time	message	accuracy	is_model_accepted	created_time_stamp
25	ecd210e1-eea7-4227-94ab-a906088c9265	2022-08-06-15-56-50	True	2022-08-06 15:56:50.749815	NaN	NaN	Pipeline has been started.	NaN	NaN	2022-08-06 15:56:50.750815
26	ecd210e1-eea7-4227-94ab-a906088c9265	2022-08-06-15-56-50	False	2022-08-06 15:56:50.749815	2022-08-06 15:56:56.464038	0 days 00:00:05.714223	Pipeline has been completed.	0.751289	True	2022-08-06 15:56:56.464038
27	15bfb291-7067-4b85-9936-423b2937ad9e	2022-08-06-15-57-37	True	2022-08-06 15:57:37.376484	NaN	NaN	Pipeline has been started.	NaN	NaN	2022-08-06 15:57:37.376484
28	15bfb291-7067-4b85-9936-423b2937ad9e	2022-08-06-15-57-37	False	2022-08-06 15:57:37.376484	2022-08-06 15:57:42.495554	0 days 00:00:05.119070	Pipeline has been completed.	0.751289	True	2022-08-06 15:57:42.496725

6. Conclusion

The insurance premium predictor model estimates the insurance premium one has to shell out based on their particular background and their personal details.

This model will go great lengths in **easing the manual mathematical calculations** involved in calculating the insurance premiums

This also aids the end consumer to **predict the expenses** he/she has to bear on the insurance premium and also helps the insurance companies to make their insurance **premiums predictable, scientific and rule based**. This can assist a person / insurance company in concentrating on the health side of an insurance policy rather than the ineffective part.