ARCHITECTURE

Insurance Premium Prediction

Written By	**Manthan Takalkar, Adityaraj Chaudhari**
Document Version	**0.2**
Last Revised Date	**18 - 10 - 2021**

Document Control

Version	Date	Author	Comments
0.1	21 – August - 2021	Manthan Takalkar Adityaraj Chaudhari	Introduction, Architecture Workflow, Architecture Description Defined
0.2	18-October-2021	Manthan Takalkar Adityaraj Chaudhari	User Work Flow Architecture Added

Index

1. Introduction

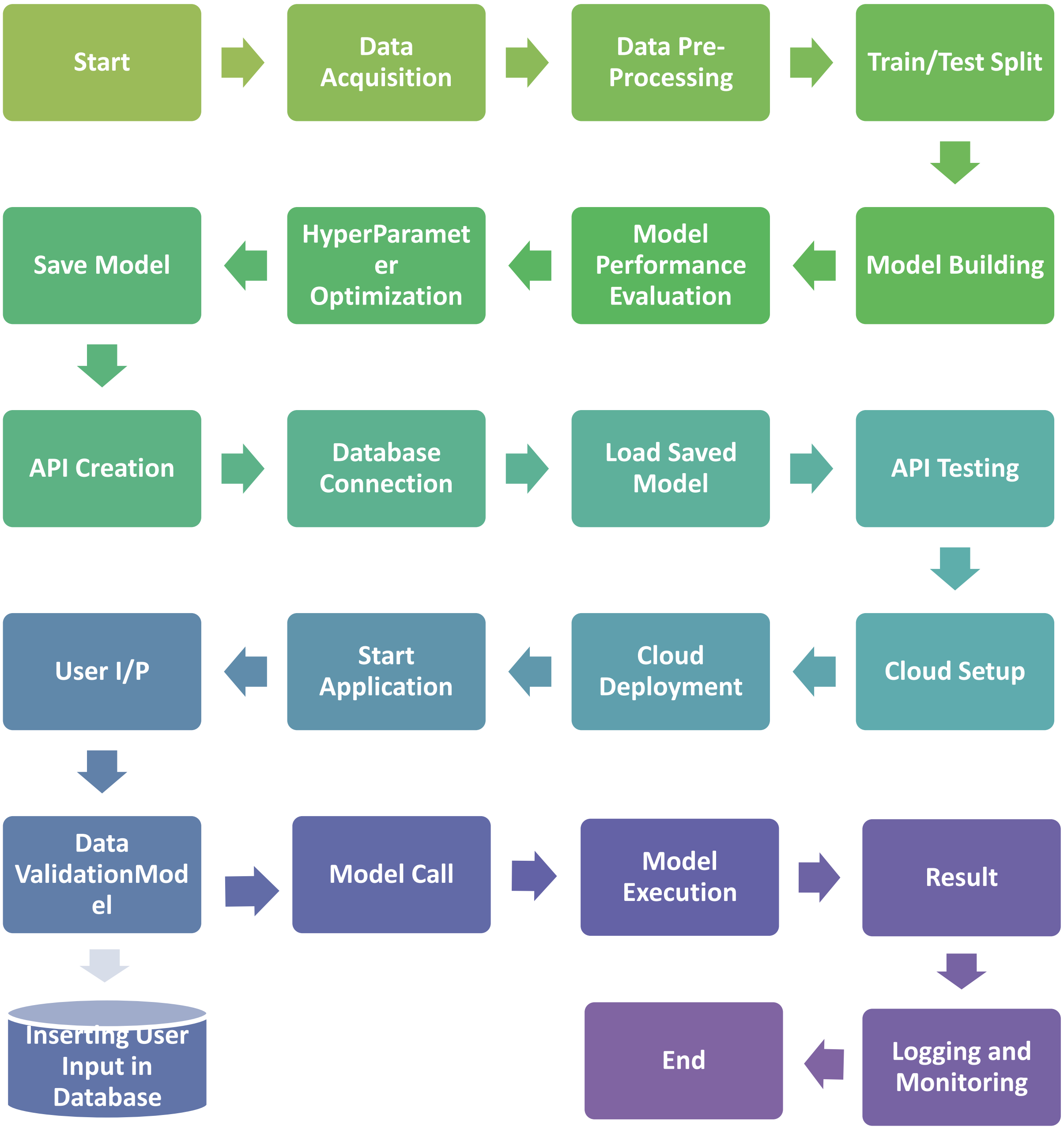
1.1. What is Architecture ?

- The Software architecture is, simply, the organization of a system. This organization includes all components, how they interact with each other, the environment in which they operate, and the principles used to design the software.
- Architecture serves as a blueprint for a system. It defines a structured solution to meet all the technical and operational requirements.

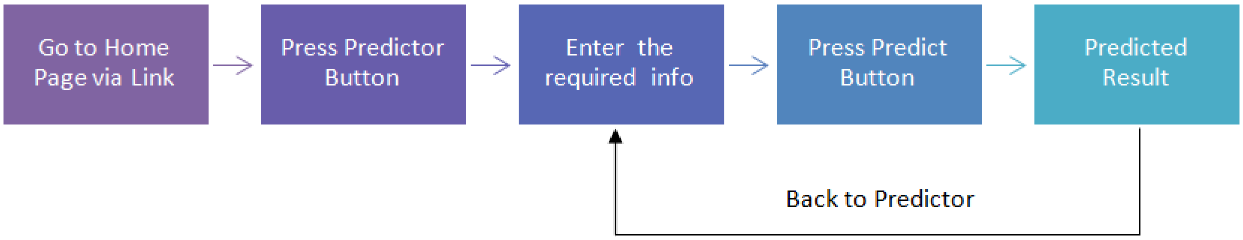
1.2. Benefits

- Its builds a solid foundation
- Helps to manage complexity
- It builds better interpretability by dividing the process in bits and pieces
- Better code maintainability

2.1 General Architecture



2.2 User Work Flow Architecture



3 Architecture Description

3.1 Data Description

The dataset contains 1338 observations (rows) and 7 features (columns). The dataset contains 4 numerical features (age, BMI, children and expenses) and 3 nominal features (sex, smoker and region). This features can help for predicting future medical expenses of individuals that help medical insurance to make decision on charging the premium.

3.2 Data Acquisition

The Dataset we acquired is in CSV(Comma Separated Format) format and was accessed from the given source

3.3 Data Pre-Processing

Data pre-processing step can be used for handling categorical values, statistical analysis, Outlier detection etc.

3.4 Train/Test Split

Spliting the dataset into two sets I.e into Training set and Testing set. This will help us for model training and evaluation. Here we can follow hold-out cross validation approach, k-fold cross validation or stratified k-fold validation approaches which allow us to compare the performance of machine learning algorithms for our predictive modeling problem.

3.5 Model Building

After performing Train/Test split we can build model by using various machine learning algorithm on training set and testing set can be used for checking model performance and model evaluation.

3.6 Model Performance and Evaluation

After performing hyper parameter optimization technique we can proceed further to check model performance to see how model performs with test data. In case of supervised ML for regression use cases we can use evaluation metrics like R-square, Adjusted R-square, mean squared error(MSE), etc. and for classification use cases we can use evaluation metrics like confusion matrix, sensitivity, specificity, accuracy score, etc.

3.7 Hyper Parameter Tuning

To increase model accuracy, to find set of right hyper parameters and to get generalized model we can use hyper parameter tuning. We can perform hyper parameter tuning using techniques like Randomized Search CV, Grid Search CV, Genetics algorithm, bayesian optimization,etc.

3.8 Saving Created Model

Saving the created Machine Learning model into pickle file format,which will increase the reusability.

3.9 API Creation

API(Application Programming Interface) creation is the process of creating and exposing APIs. We can use flask frame-work to create API which act as an interface between web application. To design a web page we can use HTML, CSS, etc.

3.10 Database Connection

For data storing purpose we have used PostgreSQL database. With the help of psycopg2 we can connect PostgreSQL database to the application.For storing entered data from created web application the database connection is required.

3.11 Loading Saved Model

After database connection we load the model which is saved to the disk in the pickle file for use of further prediction.

3.12 API Testing

To check whether API gives desired result we perform API testing. API testing is a type of software testing that involves testing application programming interfaces directly and as part of integration testing to determine if they meet expectations for functionality, reliability, performance, and connections.

To perform API testing we can use tools like Postman, Soap UI, etc.

3.13 Cloud Setup

We will be deploying the model to **Heroku** cloud platform. The process of deployment of Heroku starts with creating the app with any desired name and choosing the region. Then we create Heroku PostgreSQL (Database) instance. After this we connect our app with the Github and fetch required repository to successfully setup the cloud environment.

3.14 Cloud Deployment

Once the cloud setup process is successfully completed then we are ready to deploy our app to the cloud.

3.15 Start Application

TO start the web application we click the URL received after deployment. This will help to verify whether web application is running properly.

3.16 User I/P

After opening the application the user can enter the data such as Age, BMI, Region,etc. in input fields to get the rough estimate amount of insurance policy to be selected.

3.17 Data Validation

Data Validation is a process to restrict user to enter the data within the specified range to get valid output.

3.18 User Data Inserting into Database

Collecting the data from the user and storing it into the database. The database used here is Heroku PostgreSQL.

3.19 Model Call

Once user input is given then machine learning model is loaded internally to give prediction/output based on inputs given.

3.20 Model Execution

Model execution is nothing but only the user input we received is given to the model that we have called earlier for prediction purpose.

3.21 Result

Finally after the model execution the rough estimate amount of insurance premium is displayed on the user screen.

3.22 Logging and Monitoring

To keep the track of end to end application we perform logging. To store these logs we have use file logging method in which we store our logs in text file for monitoring purpose.

4. User Work Flow Architecture

4.1 Go to Home Page via Link

The link/url of deployed application can be used to visit the home page of the application.

4.2 Press Predictor Button

There is the Premium Predictor Button on the Home Page of the application that can help user navigate to the predictor page.

4.3 Enter the required info

On the predictor page in the respected fields the user is expected to give the proper information to get good results.

4.4 Press Predict Button

There is the Predict Button on the Predictor Page of the application that can help user get the predicted results on the results page.

4.5 Predicted Results

By Pressing the Predict button user submits the responses and gets the predicted results on the new results page.

4.6 Back to Predictor Page

By pressing the Back to Predictor from the slide bar user can go back to the predictor page if he wants to submit other responses.

5 Test Cases

Sr.No	Test Case Description	Expected Result	Actual Result	Outcome
1.	To verify whether URL is working	The URL should be accessible to the user	The URL is accessible by the user	Pass
2.	To verify whether the slider-bar is working.	The slide-bar should slide from the left side and display the navigation icons.	The slide bar slides from the left and shows the navigation icons.	Pass
3.	To verify whether the Home icon from the slide-bar works.	The Home icon on the slide bar should navigate the user on the home page.	The home icon navigates the user to the home page.	Pass
4.	To verify whether the About icon from the slide-bar works.	The About icon from the slide bar should navigate the user to the about (app info)page.	The about icon navigates the user to the about page.	Pass
5.	To verify whether the Contact icon from the slide-bar works.	The Contact icon from the slide bar should navigate the user to the contacts page.	The about icon navigates the user to the contact page.	Pass
6.	To verify whether the back/home button on Contact and About page works.	The back/home button on the Contact Page and About Page should navigate the user to the Home Page	The back button on these pages navigates the user to the Home Page.	Pass
7.	To check whether the Premium Predictor button works.	The Premium Predictor button should navigate the user to main application page.	The Premium Predictor button navigates the user to main application page.	Pass
8.	To check whether input data can be entered into input fields.	User must be able to enter the data into input field	User is able to enter the data into input field	Pass
9.	To verify whether the drop-down list from some of the input fields is working.	The drop-down menu from some input fields should slide down to help use choose correct option.	The drop-down menu from some input fields slides down.	Pass
10.	To check whether user input is valid	The input entered by user should be valid	The input entered by the user is valid	Pass
11.	To check whether user is able to reset the input fields	By clicking the reset button user must be able to reset input field	User is successfully able reset the input fields on clicking the reset button	Pass

12.	To verify whether user is able to submit the inputs by clicking submit button.	By clicking the submit button user must be able to submit inputs	User is able to submit the actual data	Pass
13.	To verify whether user is able to see the output page.	User must be able to see the output page.	User is able to see the output page.	Pass
14.	To check whether user is able to see the predicted result	User must be able to see the predicted result	User is able to see the predicted result	Pass
15.	Verify whether the predicted results are in accordance to the entered inputs by user.	The Predicted result should be correct	The predicted results are in accordance to the entered inputs by user	Pass
16.	To verify whether user is able to navigate back to Home Page or Predictor Page.	By using the slide-bar user should be able to navigate to Home Page and Predictor Page.	By using slide bar user is able to navigate to Home Page and Predictor Page.	Pass
17.	To check whether user is able to submit form again.	By navigating back to Predictor Page user should be able to input the data and should get the desired results.	User is able to get desired results.	Pass.
18.	Finally to check whether all links and buttons works fine.	By navigating to any page the user should be able to access any link or should be able to do certain operation by pressing the button.	User is able to access all links and is able to perform operations by pressing buttons.	Pass.