Detail Project Report

Insurance Premium Prediction

Revision Number : 1.2

Last Date of Revision: 05/05/2023

Document Version Control

Date	Version	Description	Author
01/05/2023	1.0	Abstract Introduction General Description	Ambarish Singh
03/05/2023	1.1	Technical Requirements Data Requirements Data Pre-processing Design Flow	Ambarish Singh
05/05/2023	1.2	Data from User and its validation Rendering the Results Deployment Conclusion	Ambarish Singh

Contents

Abstract

To give people an estimate of how much they need based on their individual health situation. After that, customers can work with any health insurance carrier and its plans and perks while keeping the projected cost from our study in mind. I am considering variables as age, sex, BMI, number of children, smoking habits and living region to predict the premium. This can assist a person in concentrating on the health side of an insurance policy rather than the ineffective part.

1. Introduction

1.1 Why this DPR Document?

The main purpose of this DPR documentation is to add the necessary details of the project and provide the description of the machine learning model and the written code. This also provides the detailed description on how the entire project has been designed end-to-end.

Key points:

- Describes the design flow
- Implementations
- Software requirements
- Architecture of the project
- Non-functional attributes like:
- Reusability
- Portability
- Resource utilization

2. General Description

2.1 Problem Perspective

The Insurance premium prediction is a machine leaning model that helps users to understand their insurance premium price based on some input data.

2.2 Problem Statement

The main goal of this model is to predict Insurance premium price based on some input data like bmi, gender, age etc.

2.3 Proposed Solution

To solve the problem, we have created a user interface for taking the input from the user to predict insurance premium price using our trained ML model after processing the input and at last the predicted value from the model is communicated to the user.

3. Technical Requirements

As technical requirements, we don't need any specialized hardware for virtualization of the application. The user should have a device that has the access to the web and the fundamental understanding of providing the input. And for

the
backend, we need a server to run all the required packages to process the input and predict the desired output.

3.1 Tools Used

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

- VS Code is used as IDE.
- For visualization of the plots, Matplotlib, Seaborn and Plotly are used.
- Streamlit is used for deployment of the model.
- Front end development is done using Streamlit.
- Python Flask is used for backend development.
- GitHub is used as version control system.

4. Data Requirements

The Data requirements totally supported the matter statement and also the dataset is accessible on the Kaggle within the file format of (.zip).

4.1 Data Collection

The data for this project is collected from the Kaggle Dataset, the URL for the dataset is https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction

4.2 Data Description

Insurance Premium dataset publicly available on Kaggle. The information in the dataset is present in one csv files named as insurance.csv. Dataset contains 1338 rows which shows the information such age, bmi, children and expenses.

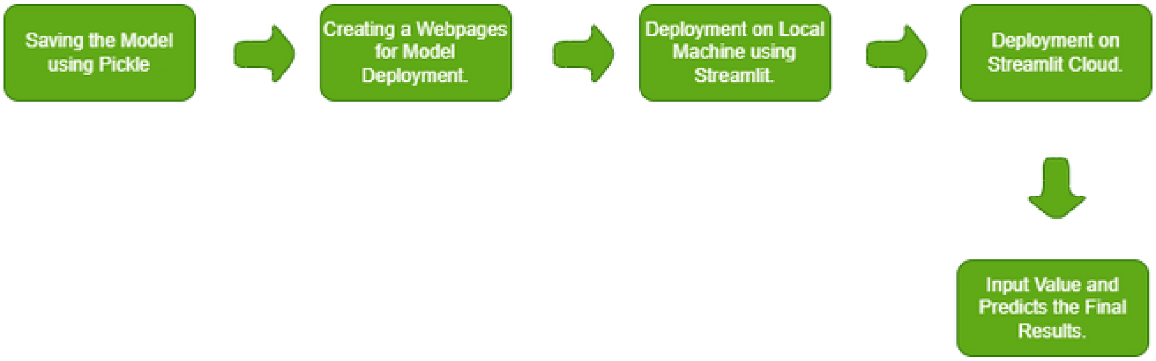
4.3. Data Pre-processing

- Checked for info of the Dataset, to verify the correct datatype of the Columns.
- Checked for Null values, because the null values can affect the accuracy of the model.
- Performed One – Hot encoding on the desired columns.
- Checking the distribution of the columns to interpret its importance.

Now, the info is prepared to train a Machine Learning Model

5. Design Flow

5.1 Deployment Process


Saving the Model using Pickle
Creating a Webpages for Model Deployment.
Deployment on Local Machine using Streamlit.
Deployment on Streamlit Cloud.
Input Value and Predicts the Final Results.

5.2 Logging

In logging, at each time an error or an exception occurs, the event is logged into the system log file with reason and timestamp. This helps the developer to debug the system bugs and rectify the error.

5.3 Data from User

The data from the user is retrieved from the created Streamlit web page.

5.4 Data Validation

- The data provided by the user is then being processed by app.py file and validated.
- The validated data is then sent to the prepared model for the prediction.

5.5 Rendering the Results

The data sent for the prediction is then rendered to the web page.

6. Deployment

The tested model is then deployed to Streamlit. So, users can access the

project from any internet device.

7. Conclusion

The Insurance Premium Prediction system will predict the price for helping the customers with the trained knowledge with set of rules. The user can use this system to recognize the approximate value of their insurance premium

8. Frequently Asked Questions (FAQs)

Q1) What’s the source of data?

The data for training is provided by the client in multiple batches and each batch contain multiple files.

Q2) What was the type of data?

The data was the combination of numerical and Categorical values.

Q3) What’s the complete flow you followed in this Project?

Refer Page no 6 for better Understanding.

Q4) After the File validation what you do with incompatible file or files which didn’t pass the validation?

Files like these are moved to the Achieve Folder and a list of these files has been shared with the client and we removed the bad data folder.

Q5) How logs are managed?

We are using different logs as per the steps that we follow in validation and modelling like File validation log, Data Insertion, Model Training log, prediction log etc.

Q6) What techniques were you using for data pre-processing?

- Removing unwanted attributes.
- Visualizing relation of independent variables with each other and output variables.
- Checking and changing Distribution of continuous values.
- Removing outliers
- Cleaning data and imputing if null values are present.
- Converting categorical data into numeric values.

Q7) How training was done or what models were used?

- Before dividing the data in training and validation set, we performed pre-processing over the data set and made the final dataset.
- As per the dataset training and validation data were divided.
- Algorithms like Linear regression, SVM, Decision Tree, Random Forest, XGBoost were used based on the recall, final model was used on the dataset and we saved that model.

Q8) How Prediction was done?

The testing files are shared by the client. We Performed the same life cycle on the provided dataset. Then, on the basis of dataset, model is loaded and prediction is performed. In the end we get the accumulated data of predictions.

Q9) What are the different stages of deployment?

- First, the scripts are stored on GitHub as a storage interface.
- The model is first tested in the local environment.
- After successful testing, it is deployed on Streamlit Cloud..