Detail Project Report

Insurance Premium Prediction

1. Introduction

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.

DPR serves as a comprehensive blueprint for a project. It outlines the project's objectives, scope, feasibility, timelines, costs, risks, and other essential details. The purpose of the DPR document is to provide a detailed analysis and justification for the project, enabling stakeholders to make informed decisions regarding its implementation.

Key points:

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

2. General Description

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.

Problem Statement

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

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

For the virtualization of the application, specialized hardware is not necessary. Users only need a device with web access and a basic understanding of providing input. On the backend, a server is required to run all the necessary packages for processing the input and generating the desired output

Tools Used

The entire model is built using the Python programming language and various frameworks such as NumPy, Pandas, Scikit-learn, Flask, and VS Code is utilized as the integrated development environment (IDE).

For visualization purposes, Matplotlib, Seaborn, and Plotly are employed to generate plots and charts.

Stream lit is utilized for deploying the model and serving as the front-end development tool.

Python Flask is used for backend development, handling the processing of data and model predictions.

Git Hub serves as the version control system for the project, enabling collaboration and code management.

.

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).

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

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.

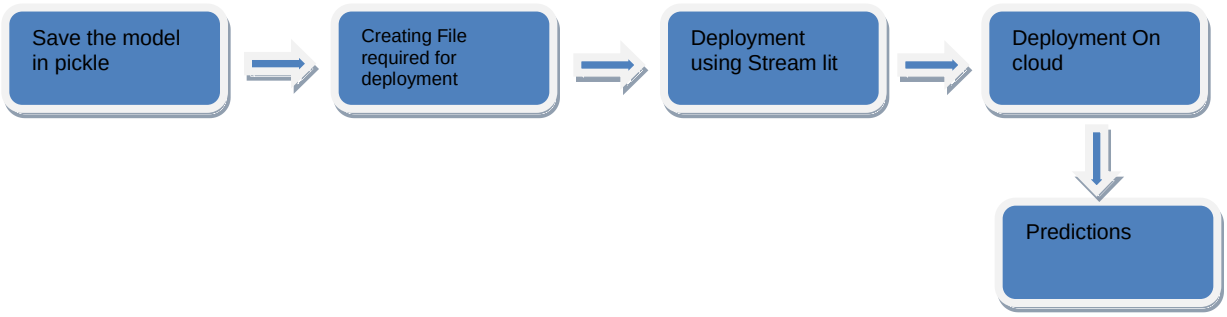
Data Pre-processing

- 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

Deployment Process



Logging

Each time an error or exception occurs in logging, the event is recorded in the system log file along with the reason and timestamp. This practice assists developers in debugging system bugs and rectifying errors.

Data from User

The data from the user is fetched from the Stream lit web page that has been created.

Data Validation

- The user-provided data is processed and validated by the app.py file.
- Once the data is validated, it is forwarded to the prepared model for prediction.

Rendering the Results

The predicted data is then displayed or rendered on the web page for the user to see..

6. Deployment

The tested model is deployed to Streamlit, enabling users to access the project from any internet-connected device..

7. Conclusion

The Insurance Premium Prediction system utilizes trained knowledge and a set of rules to predict the price, assisting customers in estimating the approximate value of their insurance premium. Users can leverage this system to gain insights into the expected cost of their insurance coverage

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..