

Architecture Design

Insurance Premium Prediction

Contents

Abstract

1. Introduction

2. Architecture

3. Architecture Design

3.1 Data Collection

3.2 Data Description

3.3 Data Pre-processing

3.4 Modelling Process

3.5 UI Integration

3.6 Data from user

3.7 Data validation

3.8 Rendering the results

Abstract

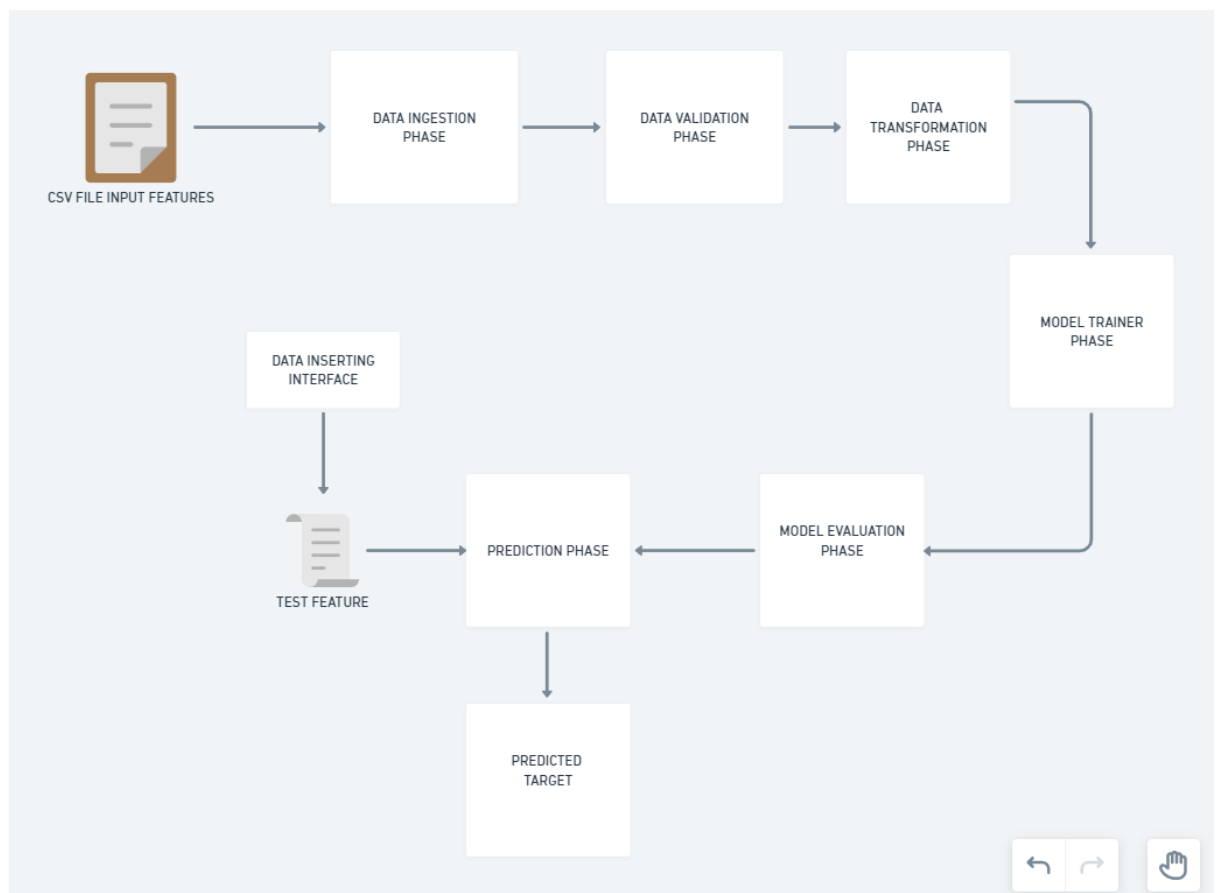
To give people an estimate of how much they need based on their individual health situation. After that, customers can work with any health insurance carrier and its plans and perks while keeping the projected cost from our study in mind. We are considering variables as age, sex, BMI, number of children, smoking habits and living region to predict the premium amount. 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 Architecture Design Document?

The main objective of the Architecture design documentation is to provide the internal logic understanding of the Insurance Premium Prediction code. The architecture design documentation is designed in such a way that the programmer can directly code after reading each module description in the documentation.

2.Architecture



3. Architecture Design

3.1 Data collection

The data for this project is collected from the Kaggle dataset.

<https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction>

3.2 Data Description

The dataset contains features related to any individual such as age, sex, number of children, smoking habit, region of residence, bmi and other.

3.3 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 label encoding to encode categorical variable.
- Checking the distribution of the columns to interpret its importance.

3.4 Modelling Process

After pre-processing the data, we visualize our data to gain insights and split into two parts, train and test data. After splitting the data, we use different machine learning models to predict the result.

3.5 UI integration

Streamlit files are being created and being integrated with the machine learning model. All the required files are then integrated to the app.py file.

3.6 Data from user

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

3.7 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.

3.8 Rendering the results

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