Architecture Design

Insurance Premium Prediction

Document Version Control

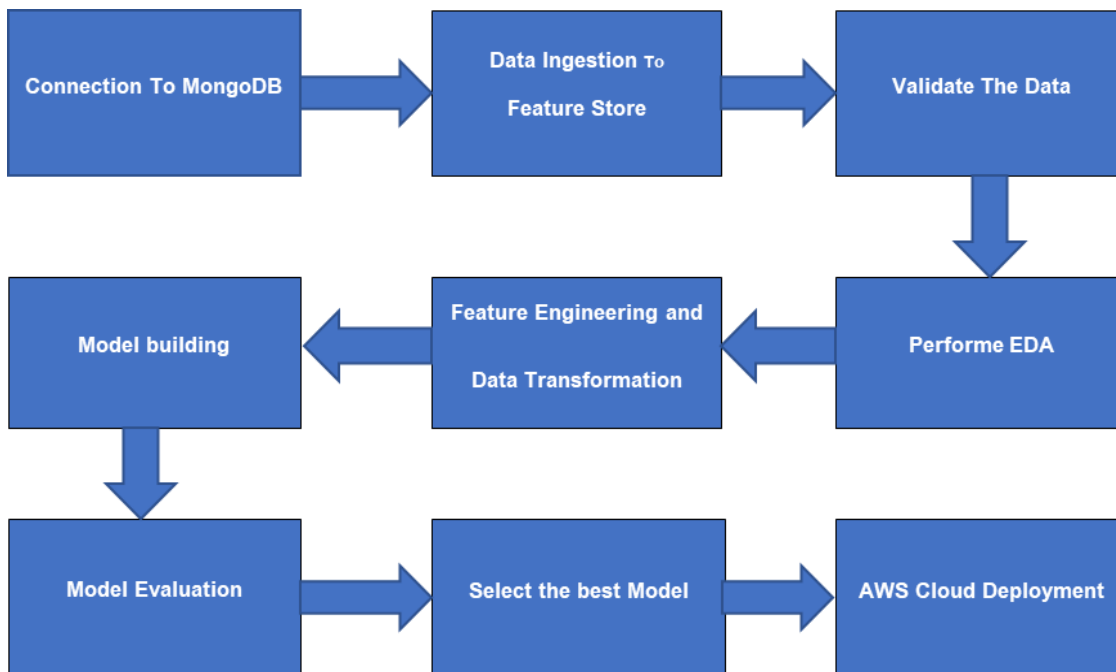| Date	Version	Description	Author
11 – 12 - 2023	1.0	Architecture Design	Dhananjay Gurav

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.

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 directlycode after reading each module description in the documentation.

2. Architecture

```
┌──────────────────────┐      ┌──────────────────────┐      ┌──────────────────────┐
│                      │      │  Data Ingestion To   │      │                      │
│ Connection To MongoDB│ ───► │                      │ ───► │   Validate The Data  │
│                      │      │   Feature Store      │      │                      │
└──────────────────────┘      └──────────────────────┘      └──────────────────────┘
                                                                        │
                                                                        ▼
┌──────────────────────┐      ┌──────────────────────┐      ┌──────────────────────┐
│                      │      │ Feature Engineering  │      │                      │
│    Model building    │ ◄─── │        and           │ ◄─── │    Performe EDA      │
│                      │      │ Data Transformation  │      │                      │
└──────────────────────┘      └──────────────────────┘      └──────────────────────┘
          │
          ▼
┌──────────────────────┐      ┌──────────────────────┐      ┌──────────────────────┐
│                      │      │                      │      │                      │
│   Model Evaluation   │ ───► │ Select the best Model│ ───► │ AWS Cloud Deployment │
│                      │      │                      │      │                      │
└──────────────────────┘      └──────────────────────┘      └──────────────────────┘
```

3. Architecture Design

3.1 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

3.2 Data Description

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

3.3 Data Pre-processing

- Checked for info of the Dataset, to get idea about 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 categorical columns.
- Checking the distribution of the numerical columns to interpret its importance. Now, the info is prepared to train a Machine Learning Model.

3.4 Model Training Process

After pre-processing & transforming the data, we split the data into two parts, train and test data. After splitting the data, we applied different Machine Learning Models like Linear Regression, Random Forest Regressor, Decision Tree Regressor and XGB Regressor to predict the Insurance Premium Price.
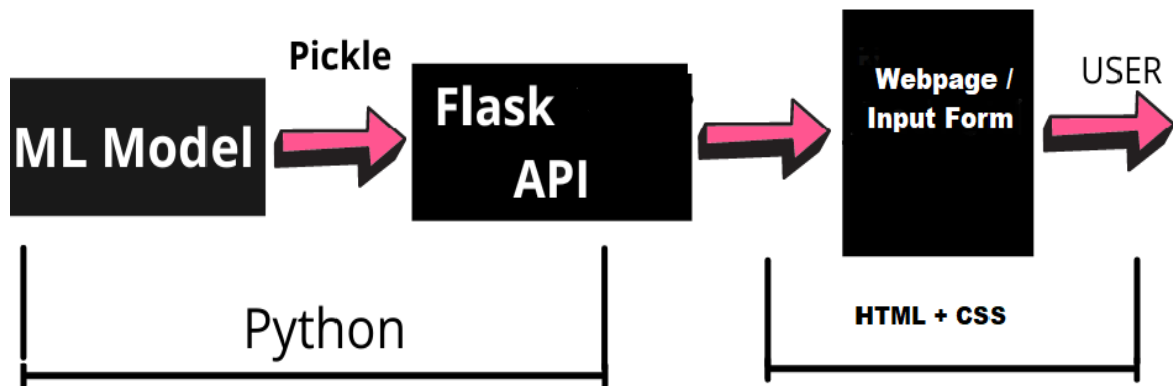
3.5 UI Integration

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

3.6 Data from User

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

3.7 Data Preprocessing

The data provided by the user is then processed by Preprocessor object loaded in app.py file. The transformed. data is then sent to the prepared model for the prediction.

3.8 Rendering the Results

The Prediction result then returned to user through result.html page

3.9 Deployment

The tested app is then deployed to AWS cloud platform. So, users can access the project from anyinternet device.