Architecture Design

Insurance Premium Prediction

Written By	Joji Samuel
Document Version	1.1
Last Revised Date	29-Nov-2023

Document Version Control

Version	Date	Author	Comments
1.0	27-Nov-2023	Joji Samuel	Architecture data creation
1.1	28-Nov-2023	Joji Samuel	Document updated

Contents

Abstract

Machine Learning is a category of algorithms that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. The basic premise of machine learning is to build models and employ algorithms that can receive input data and use statistical analysis to predict an output while updating outputs as new data becomes available. These models can be applied in different areas and trained to match the expectations of management so that accurate steps can be taken to achieve the organization's target. In this project, we will estimate the amount of insurance premium on the basis of personal health information. Taking various aspects of a dataset collected from people, and the methodology followed for building a predictive model.

1. Introduction

1.1. What is an Architecture Design?

Architecture Design as the process of creating a structured solution that meets technical and operational requirements while optimizing for desired qualities. It serves as a blueprint for the system's structure and behavior, guiding implementation and evolution. The importance of a well-defined architecture is underscored for ensuring a robust and scalable foundation in the development and deployment of complex systems.

1.2. Scope

The scope of Architecture Design involves identifying and defining the system's boundaries, components, modules, and their interactions. It includes considerations for system components, data flow, interfaces, and external dependencies, taking into account both functional and non-functional requirements. A clearly defined scope is emphasized for efficient resource utilization and focused development efforts.

1.3 Constraints

We only predict the expected estimated cost of expenses to customers based on some personal health information.

2. Technical Specification

2.1 Dataset

The dataset containing verified historical data, consisting of the information and the actual medical expenses incurred by over 1338 customers. The objective is to find a way to estimate the value in the "expenses" column using the values in the other columns like their Age, Sex, BMI, No. of children, Smoking habits and Region. Using all the observations it is inferred what role certain properties of user and how they affect their expenses.

The provided dataset consists of the following columns:

age : Represents the age of individuals. It is of integer type.
sex : Represents the gender of individuals. It is of object type, suggesting categorical data.
bmi : Represents the Body Mass Index of individuals. It is of float type, indicating numerical data.
children: Indicates the number of children or dependents covered by the insurance. It is of integer type.
smoker: Indicates whether an individual is a smoker or not. It is of object type, suggesting categorical data.
region: Represents the geographical region of individuals. It is of object type, indicating categorical data.
expenses: Represents the insurance expenses. It is of float type, indicating numerical data.

Pre-processing of this dataset includes doing analysis on the independent variables like checking for null values in each column and then replacing or filling them with supported appropriate data types so that analysis and model fitting is not hindered from their way to accuracy. Shown above are some of the representations obtained by using Pandas tools which tell about variable count for numerical columns and model values for categorical columns. Maximum and minimum values in numerical columns, along with their percentile values for the median, play an important factor in deciding which value to prioritize for further exploration tasks and analysis. Data types of different columns are used further in label processing and a one-hot encoding scheme during the model building.

2.2 Logging

We should be able to log every activity done by the user

- The system identifies at which step logging require.
- The system should be able to log each and every system flow.
- The system should not be hung even after using so much logging. Logging is just because we can easily debug issuing so logging is mandatory to do.

2.3 Deployment

For the hosting of the project, we use AWS EC2

GitHub GitHub Actions docker + AWS ECR Amazon EC2

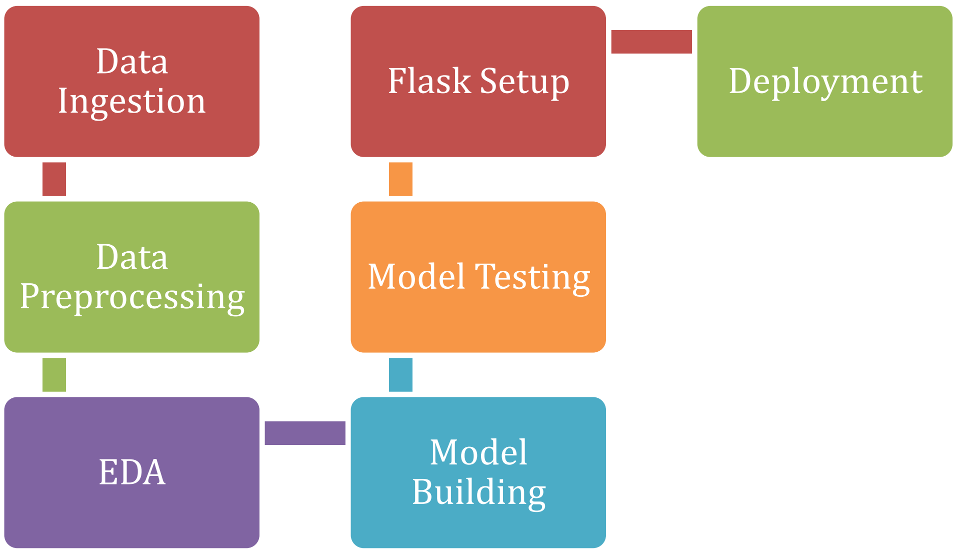
3. Technology Stack

Front End	HTML
Backend	Python/ Flask
Deployment	AWS

4. Proposed Solution

We will use performed EDA to find the important relation between different attributes and will use a machine-learning algorithm to estimate the cost of expenses. The client will fill in the required feature as input and get results through the web application. The system will get features and it will be passed into the backend where the features will be validated and pre-processed and then it will be passed to a hyperparameter-tuned machine learning model to predict the final outcome.

5. Architecture



5.1 Data Gathering

Insurance.csv file was obtained from the Machine Learning course website (Spring 2017) from Professor Eric Suess - CSU East Bay College of Science.

5.2 Raw Data Validation

After data is loaded, various types of validation are required before we proceed further with any operation. Validations like checking for zero standard deviation for all the columns, checking for complete missing values in any columns, etc. These are required because the attributes which contain these are of no use. It will not play a role in contributing to the estimated cost of the premium.

5.3 Exploratory Data Analysis

Visualized the relationship between the dependent and independent features. Also checked the relationship between independent features to get more insights about the data.

5.4 Feature Engineering

After pre-processing standard scalar is performed to scale down all the numeric features. Even one hot encoding is also performed to convert the categorical features into numerical features. For this process, the pipeline is created to scale numerical features and encode the categorical features.

5.5 Model Building

After doing all kinds of pre-processing operations mentioned above and performing scaling and encoding, the data set is passed through a pipeline to all the models, Linear Regression, and Random Forest using ML. It was found that Random Forest performs best with the smallest RMSE value i.e., 4652.33, and the highest R2 score equals 0.8527 on test data So ‘Gradient boosting’ performed well in this problem.

5.6 Model Saving

The model is saved using the joblib library in ‘joblib’ format.

5.7 Flask Setup for Web Application

After saving the model, the API building process started using Flask. Web application creation was created in Flask for testing purposes. Whatever user will enter the data and then that data will be extracted by the model to estimate the premium of insurance, this is performed in this stage

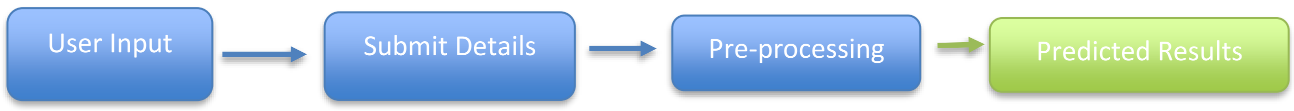
5.8 GitHub

The whole project directory will be pushed into the GitHub repository.

5.9 Deployment

The deployment strategy for the Insurance Premium Predictor ML Project involves a streamlined and automated process using GitHub Actions, Amazon Elastic Container Registry (ECR), and Amazon EC2. GitHub Actions facilitates automated builds, while ECR serves as a secure and scalable repository for Docker container images. The deployment is carried out on Amazon EC2 instances, providing scalable and on-demand computing power. This strategy ensures efficient, consistent, and version-controlled deployment, enhancing scalability and facilitating an automated workflow for the project. A diagram is included for visual clarity, illustrating the seamless flow from GitHub Actions to ECR and deployment on EC2.

6. User Input / Output Workflow:


User Input
Submit Details
Pre-processing
Predicted Results