**Architecture Design**

**INSURANCE PREMIUM PREDICTION**

**Document Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Comments** |
| 1 | 10.09.23 | Asouk S R |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Index**

|  |  |
| --- | --- |
| **Content** | **Page No** |
| Abstract | 4 |
| 1. Introduction | 4 |
| 1.1 What is Architecture Design? | 4 |
| 1.2 Scope | 4 |
| 1.3 Constraints | 4 |
| 2. Technical Specification | 5 |
| 2.1 Dataset | 6 |
| 2.2 Logging | 6 |
| 2.3 Deployment | 6 |
| 3. Technology Stack | 7 |
| 4. Proposed Solution | 7 |
| 5. Architecture | 7 |
| 5.1 Architecture Description | 8 |
| 6. User Input/Output Workflow | 10 |

# Abstract

Machine Learning is a category of algorithms that empowers software applications to enhance their accuracy in predicting outcomes without the need for explicit programming. The fundamental concept behind machine learning is the creation of models and the utilization of algorithms capable of taking input data and utilizing statistical analysis to predict outputs, all while continuously adapting these outputs as new data becomes available.

These models find applications in diverse domains and can be tailored to align with management expectations, thereby enabling informed decision-making to achieve organizational objectives.

In the context of this project, our objective is to estimate insurance premiums based on personal health information. We'll accomplish this by considering various aspects of a dataset derived from individuals and implementing a well-defined methodology to construct a predictive model."

# 1. Introduction

**1.1 What is Architecture Design?**

The goal of Architecture Design (AD) is to give the internal design of the actual program code for the `Insurance Premium Prediction`. AD describes the class diagrams with the methods and relation between classes and program specification. It describes the modules so that the programmer can directly code the program from the document.

## 1.2 Scope

Architecture Design (AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during requirement analysis and then refined during data design work. And the complete workflow.

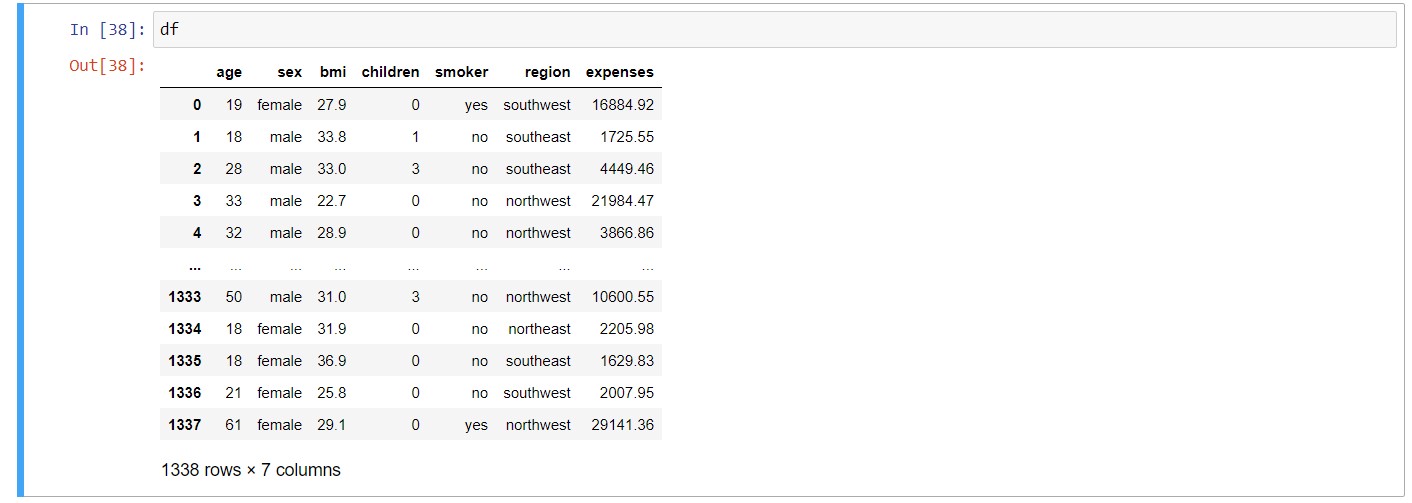
## 1.3 Constraints

We only predict the expected estimating cost of expenses customers based on some personal health information. We did not use previous existing Conditions as a feature

# 2. Technical Specification

## 2.1 Dataset

The dataset containing verified historical data, consisting of the aforementioned information and the actual medical expenses incurred by over 1300 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 dataset looks like as follow:

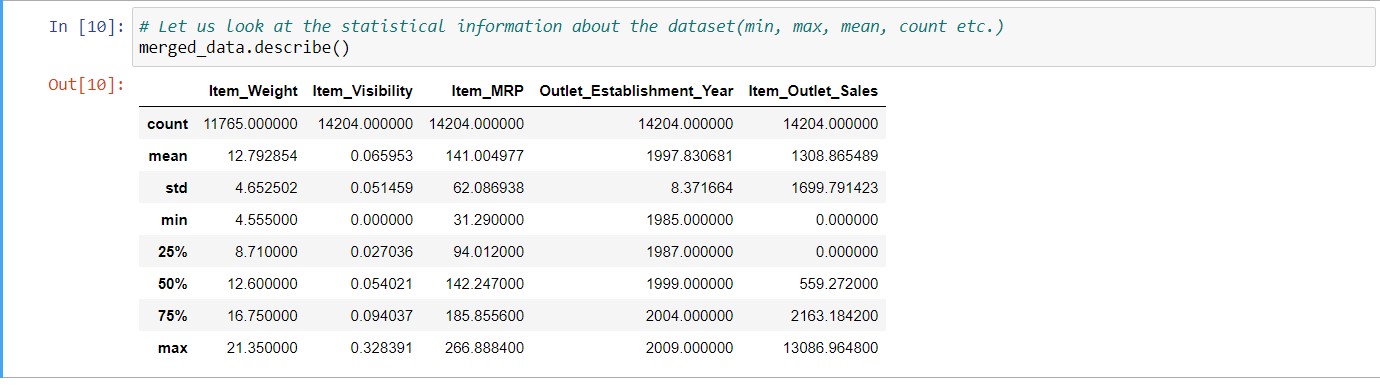


The data set consists of various data types from integer to floating to object as shown in Fig.



In the dataset, there can be various types of underlying patterns which also gives an in-depth knowledge about the subject of interest and provides insights into the problem. Looks like ‘age’, ‘children’, ‘bmi’ (body mass index) and ‘expenses’ are numbers, whereas ‘sex’, ‘smoker’, and ‘region’ are strings (possibly categories).

Various factors important by statistical means like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes



Preprocessing 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 median, play an important factor in deciding which value to be chosen at priority 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 be not be hung even after using so much logging. Logging just because we can easily debug issuing so logging is mandatory to do.

**2.3 Deployment**

For the hosting of the project, we will use Heroku.



## 3. Technology Stack

|  |  |
| --- | --- |
| **Front End** | HTML/CSS |
| **Backend** | Python/ Flask |
| **Deployment** | Heroku |

## 4. Proposed Solution

"In this project, we will leverage Exploratory Data Analysis (EDA) to identify significant relationships among various attributes. Our goal is to employ a machine learning algorithm to estimate the cost of expenses accurately.

The client will interact with a user-friendly web application by inputting the necessary features. These features will then be processed within the system.

Upon receiving the features, our backend system, powered by Azure endpoint deployment, will undertake a series of tasks. It will validate and preprocess the input features before passing them to a machine learning model. This model has been fine-tuned with hyperparameters to ensure the most accurate predictions.

Ultimately, the client will receive the final outcome, providing valuable insights into the estimated cost of expenses."

## 5. Architecture

Start

Data

Fetching

EDA

Data

Cleaning

Feature

Engineering

Model

Building

Model

Testing

Azure

Endpoint

Flask

Deployment

### 5.1 Data Gathering

Data source: <https://www.kaggle.com/noordeen/insurance-premium-prediction>Dataset is stored in .csv format.

### 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 role in contributing to the estimating cost of the premium.

### 5.3 Exploratory Data Analysis

Visualized the relationship between the dependent and independent features. Also checked 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, pipeline is created to scale numerical features and encoding the categorical features.

### 5.5 Model Building

After doing all kinds of pre-processing operations mention above and performing scaling and encoding, the data set is passed through a pipeline to all the models, Linear Regression, Decision tree, Random Forest, Gradient boost, KNN and XGBoost regressor using EvalML. It was found that Gradient boosting 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**

Model is saved using pickle library in pickle` 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 purpose. 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 project was deployed from GitHub into the Heroku platform.

**6. User Input / Output Workflow.**

