Architecture Design

Insurance Premium Prediction

Document Control

Date issued	version	author	comments
5/9/2024	V1.0	Madhuri Chandanbatwe	

Abstract

In this project,we analyze the personal health data to predict insurance premium of individuals. Predictive insurance is an advanced type of analysis that allows insurance companies to make forecasts using their historical data, combining statistical models, data mining techniques, and machine learning. Machine learning can significantly enhance the

prediction of insurance premiums through various techniques and methodologies through data analysis, selecting the most relevant features that contribute to premium calculations and predictive modelling.In this project we mostly focuss on building a premium prediction application that would help the insurance companies predict premiums for their customers.

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

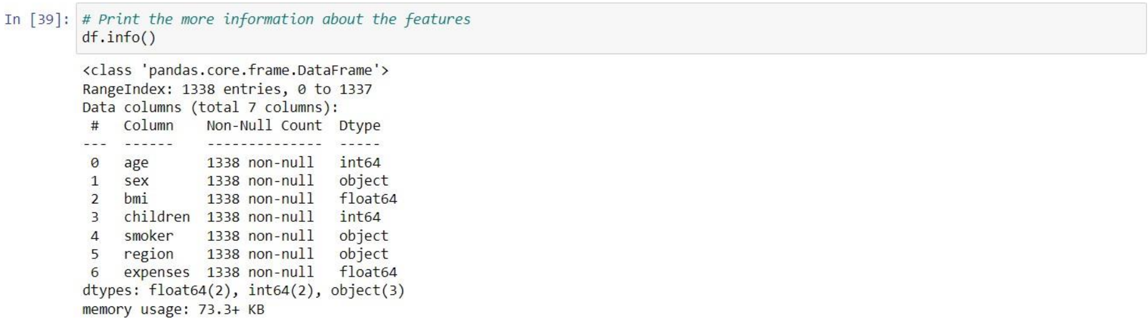
Architecture Design (AD) is a component-level design process that follows a step-by-step refinement process. This process can be used for designing data structures, required software, architecture, source code, and ultimately, performance algorithms. Overall, the data organization may be defined during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

2. Technical Specification

2.1 Dataset

The dataset containing verified historical data, consisting of the mentioned 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 data set consists of various data types from integer to floating to object as shown in Fig.

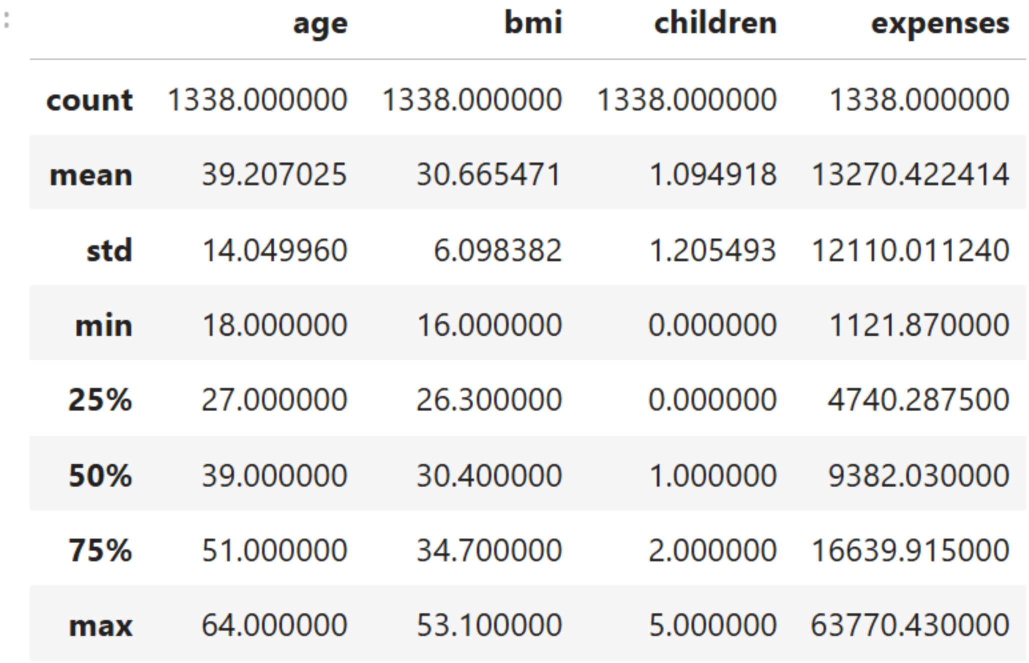


```
In [39]: # Print the more information about the features
         df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   expenses  1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

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



	age	bmi	children	expenses
count	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	30.665471	1.094918	13270.422414
std	14.049960	6.098382	1.205493	12110.011240
min	18.000000	16.000000	0.000000	1121.870000
25%	27.000000	26.300000	0.000000	4740.287500
50%	39.000000	30.400000	1.000000	9382.030000
75%	51.000000	34.700000	2.000000	16639.915000
max	64.000000	53.100000	5.000000	63770.430000

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.

3.0 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.

4.0 Deployment

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

4.1 Technology Stack

Front End	HTML/CSS
Backend	Python/ Flask
Deployment	AWS

4.2 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 be filled the required feature as input and will 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 preprocessed and then it will be passed to a hyperparameter tuned machine learning model to predict the final outcome.

4.3 Data Gathering

Data was gathered from Kaggle Insurance Premium Prediction data.It was in csv format.

4.4 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.

4.5 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.

4.6 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.

4.7 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, Adaboost,Ridge,Random Forest, Gradient boost, KNN and XGBoost regressor using EvalML. It was found that Gradient boosting performs best with the smallest RMSE value and the highest R2 score on test data So ‘Gradient boosting’ performed well in this problem.

4.8 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.

4.9 GitHub

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