Architecture Design

Insurance Premium Prediction

Document Version Control

Date	Version	Description	Author
05-02-2023	V1.0	Initial architecture design	Satya Nerurkar

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 Architecture Design?

The goal of Architecture Design (AD) is to give the internal design of the actual program code for the 'Health 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 predict the expected estimating cost of expenses customers based on some personal health information.

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 following fig 1.

Profile of the Data

```
df.head()
```

	age	sex	bmi	children	smoker	region	expenses
0	19	female	27.9	0	yes	southwest	16884.92
1	18	male	33.8	1	no	southeast	1725.55
2	28	male	33.0	3	no	southeast	4449.46
3	33	male	22.7	0	no	northwest	21984.47
4	32	male	28.9	0	no	northwest	3866.86

```
print("This dataset contains {} rows and {} columns.".format(df.shape[0], df.shape[1]))
```

```
This dataset contains 1338 rows and 7 columns.
```

Fig 1

The data set consists of various data types from integer to floating to object as shown in Fig. 2

```
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
```

The important statistics like mean, standard deviation, median, count of values and maximum value, etc. are shown below for numerical attributes.

	count	mean	std	min	25%	50%	75%	max
age	1338.0	39.207025	14.049960	18.00	27.0000	39.00	51.000	64.00
bmi	1338.0	30.665471	6.098382	16.00	26.3000	30.40	34.700	53.10
children	1338.0	1.094918	1.205493	0.00	0.0000	1.00	2.000	5.00
expenses	1338.0	13270.422414	12110.011240	1121.87	4740.2875	9382.03	16639.915	63770.43

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 the median, play an important factor in deciding which value to be chosen as a 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 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

The entire code has first been pushed into GitHub Repository, which has then been deployed to the AWS cloud platform using Docker and GitHub Actions.

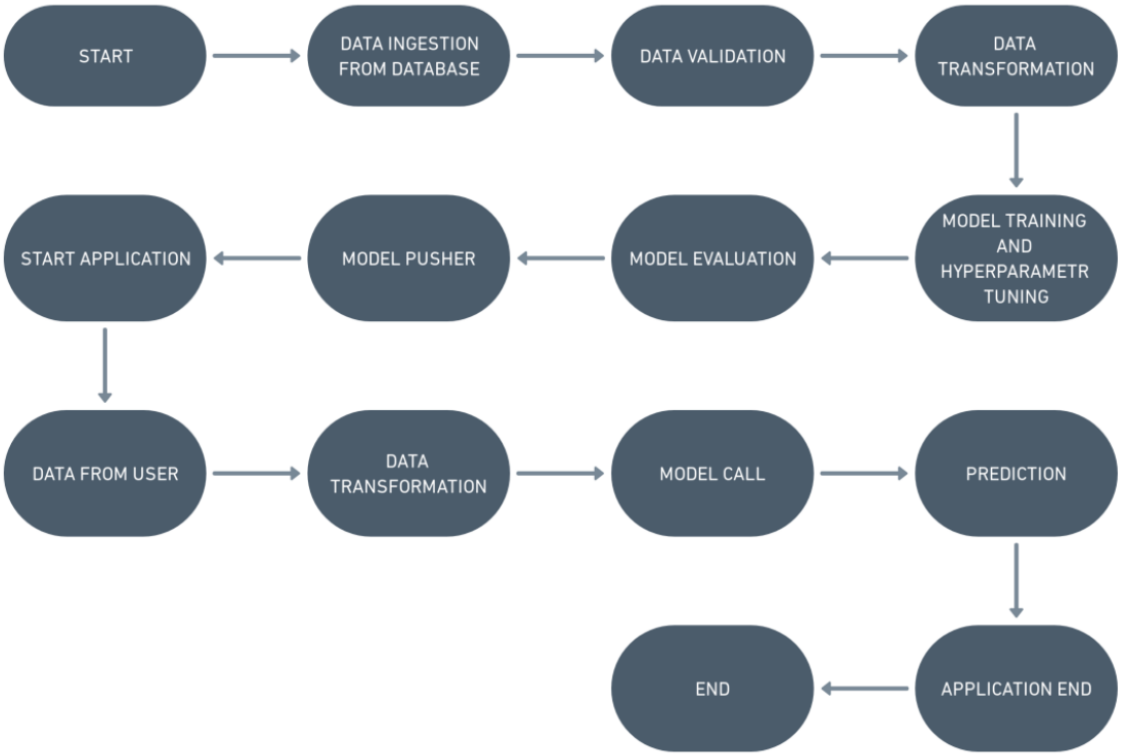
3. Technology Stack

Database	MongoDB
Front End	HTML/CSS
Backend	Python/Flask
Deployment	AWS

4. Proposed Solution

We will use performed exploratory data analysis 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 outcome.

5. Architecture



5.1 Data Gathering

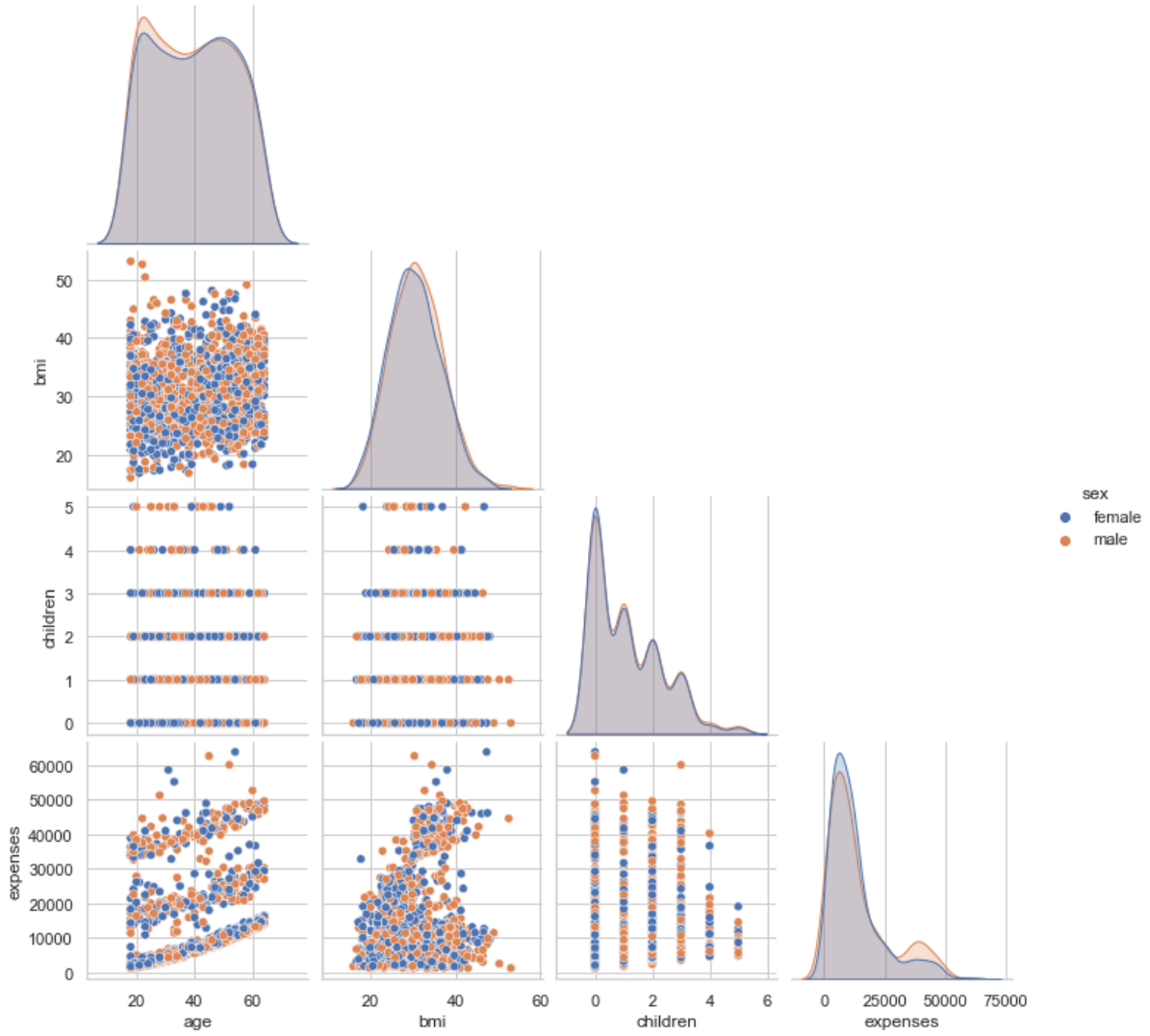
Data source: https://www.kaggle.com/noordeen/insurance-premium-prediction Dataset is stored in MongoDB database.

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.

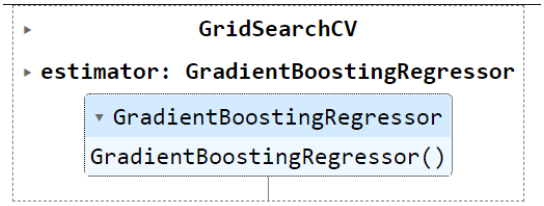


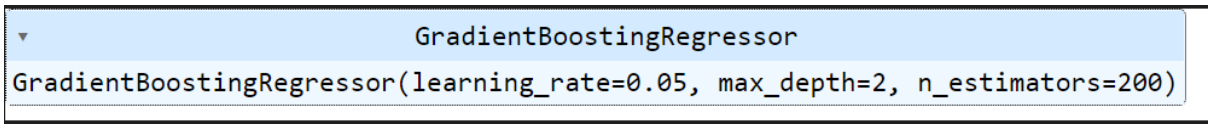
5.4 Feature Engineering

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, and Gradient booster. It was found that Gradient boosting performs best with the on-test data after that we perform Gradient SearchCV on gradient booster to find out best parameters, So after 'Gradient SearchCV' on gradient booster our model performance increases.





5.6 Model Saving

Model is saved using dill library in .pkl 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 to AWS cloud platform using Docker and GitHub Actions.

6. User Input / Output Workflow.