Architecture

Insurance Premium Prediction

Prepared By: Mangesh Devkate

Data Science Intern

iNeuron.ai

Document Version Control

Version	**Date**	**Author**	**Comments**
1	1-12-2022	Mangesh Devkate	

Contents

Abstract

Machine learning is a part of Artificial Intelligence and it is a process in which machine is having ability to learn without being explicitly programmed. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as in medicine, email filtering, speech recognition, agriculture, health, insurance and computer vision, where it is difficult or unfeasible to develop conventional algorithms to perform the needed tasks. 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?

The machine learning architecture defines the various layers involved in the machine learning cycle and involves the major steps being carried out in the transformation of raw data into training data sets capable for enabling the decision making of a system.

1.2 Scope

Architecture 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.

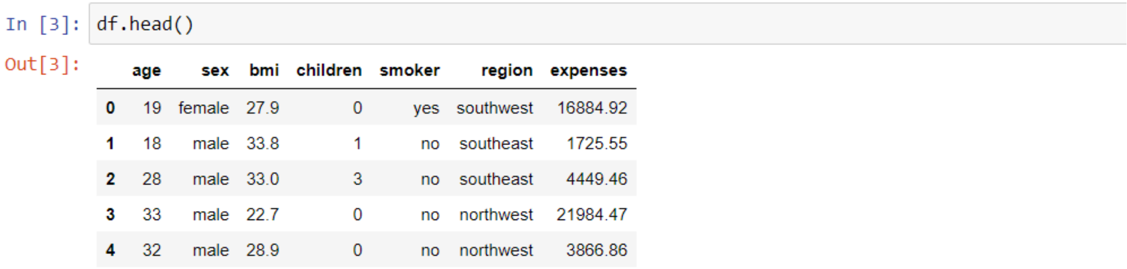
1.3 Constraints

We only predict the expected estimating cost of insurance premium of customers based on their personal health information.

2. Technical Specifications

2.1 Dataset

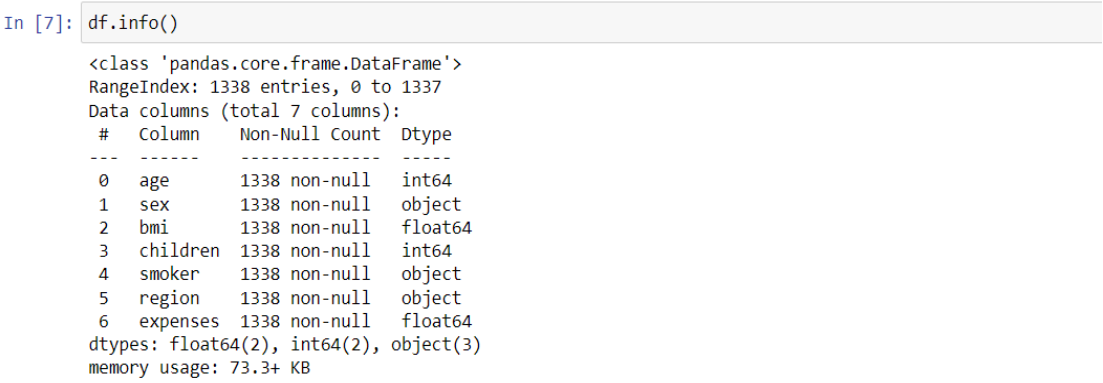
The dataset containing historical data of 1338 customers personnel information and their actual medical expenses. It consists 1338 rows and 7 columns. The objective is to estimate the customers expenses based on their personnel information like age, sex, bmi, children, smoker, region etc. The dataset looks like as follow



```
In [3]: df.head()
Out[3]:
```

	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

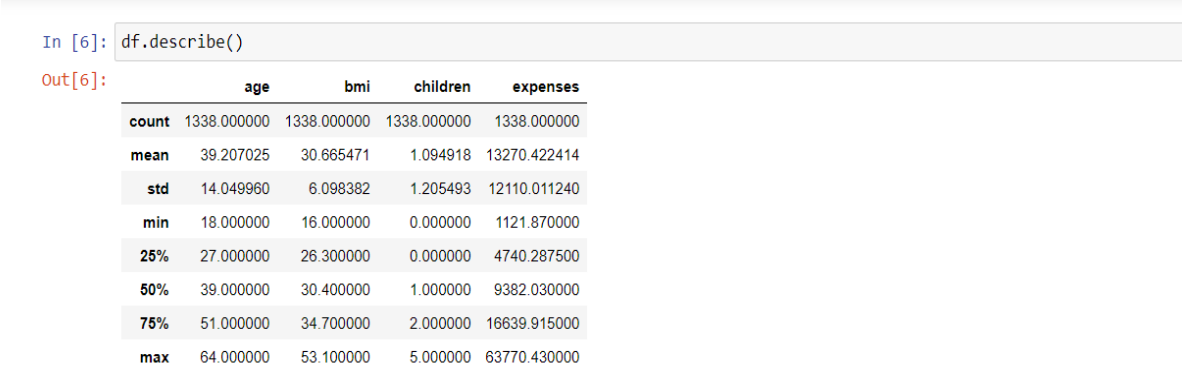
Data contains numerical as well as categorical columns. The columns in the dataset consists of various data types like int, float, object as shown in fig.



```
In [7]: 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 categorical variables are sex, smoker, region and the numerical variables are age, bmi, children and expenses. Below is the summary of the numerical variables which includes statistical information of the variables like, mean, std, min, max, percentile value of the numerical variables. as shown in below fig.



```
In [6]: df.describe()
```

Out[6]:

	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

Exploratory Data Analysis is doing analysis of dataset. Doing univariate analysis and bivariate analysis is the best way to analyse and understand the dataset. Univariate analysis can be done by using countplot, distplot, histogram, boxplot and bivariate analysis can be done by using scatterplot, pairplot. Feature Engineering includes removing duplicate rows in the dataset, filling NULL/missing values in the variables by imputing the variables, checking data distribution of variables by which we get to know that whether data distribution is normal or schewed, outlier detection and handling outliers, transformation of categorical variables into numerical variables by encoding techniques.

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

We have used Heroku for project deployment.

HEROKU

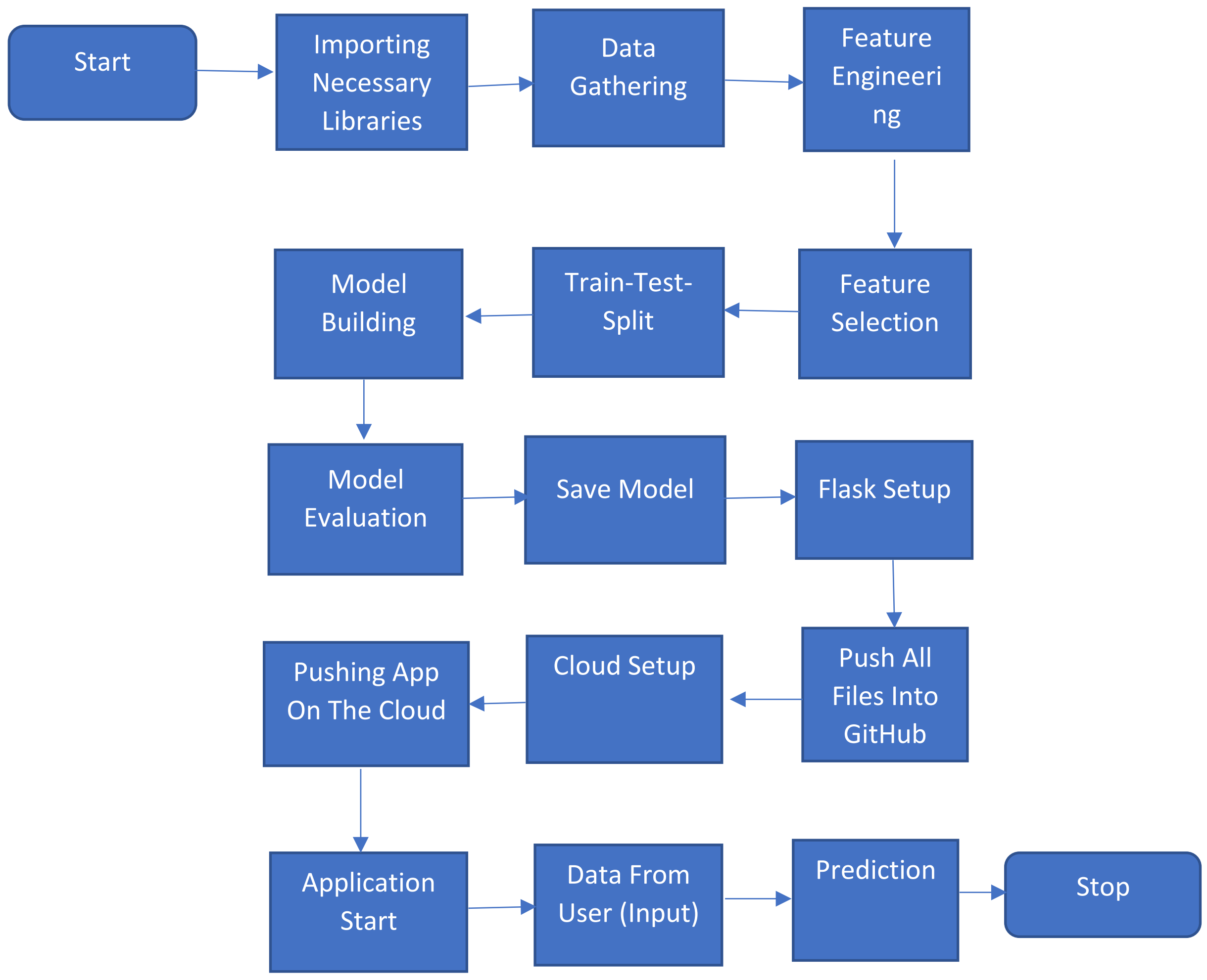
3. Technology Stack

Front End	HTML/CSS
Back End	Python/Flask
Deployment	Heroku

4. Proposed Solution

We have used EDA to find the important relation between different attributes and machine-learning algorithms to estimate the cost of expenses. The user will provide the required feature values as input and will get results through the web application. The system will get features values 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

Data Source : https://www.kaggle.com/datasets/noordeen/insurance-premium-prediction

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

In EDA we have done analysis of dependent and independent variables. We have done univariate analysis and bivariate analysis to get insights of the data.

5.4 Feature Engineering

Feature Engineering consists various steps like removing duplicate rows in the dataset, filling NULL/Missing values, outlier detection and handling outliers, transformation of categorical columns into numerical columns using encoding techniques like label encoding, one hot encoding. Adding to these feature engineering consists other steps like feature scaling, feature binning.

5.5 Feature Selection

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data. It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve. Feature selection techniques are wrapper method, filter method, embedded method.

5.6 Train-Test-Split

The train_test_split() method is used to split our data into train and test sets. First, we need to divide our data into features (X) and labels (y). The dataframe gets divided into X_train, X_test , y_train and y_test. X_train and y_train sets are used for training and fitting the model.

5.7 Model Building

Building an ML Model requires splitting of data into two sets, such as ‘training set’ and ‘testing set’ in the ratio of 80:20 or 70:30. A set of supervised (for labelled data) and unsupervised (for unlabeled data) algorithms are available to choose from depending on the nature of input data and business outcome to predict.

5.8 Model Evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance.

5.8 Model Saving

Model is saved using pickle library in pickle` format.

5.9 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. Data entered by user extracted by the model to estimate the premium of insurance.

5.10 GitHub

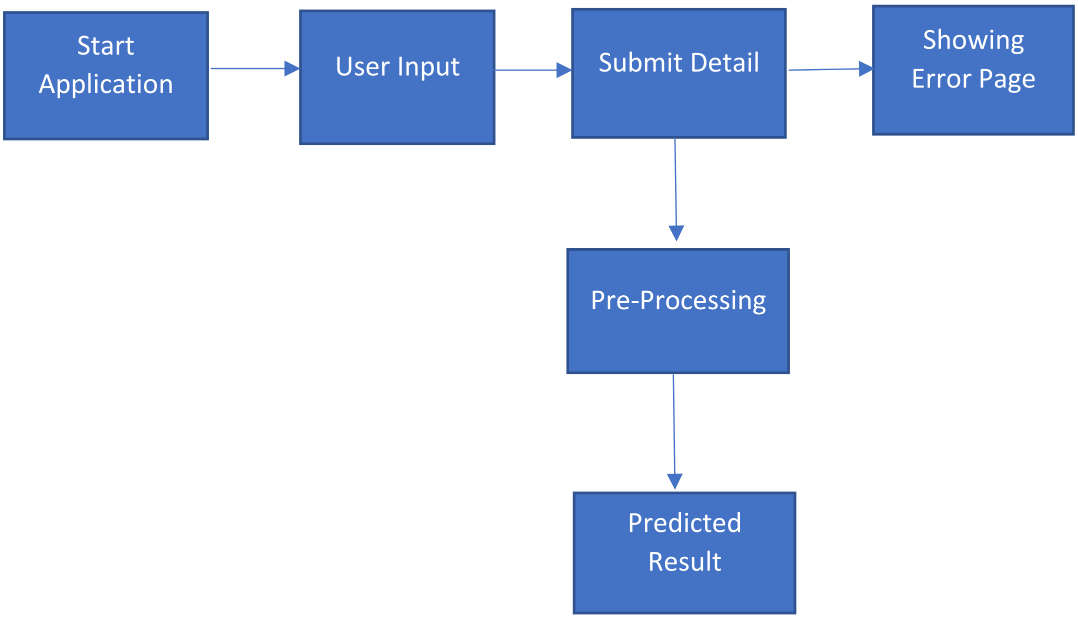
The whole project directory pushed into the GitHub repository.

5.11 Deployment

The project was deployed from GitHub into the Heroku platform.

HEROKU

6. User Input/Output Workflow


Start Application
User Input
Submit Detail
Showing Error Page
Pre-Processing
Predicted Result