DEVI AHILYA VISHVAVIDYALAYA, INDORE

SCHOOL OF STATISTICS

SESSION 2021-23

Health Insurance Cost Prediction Using Regression Models

A project report Submitted for fulfilment for the degree of

Master of Science(statistics)

May, 2023

Submitted by:

Muskan Thakur

M.Sc. 4th semester

DAVV Indore

Under the supervision of

Dr. SNIGDHA BANERJEE

Head of a Department

School of statistics

DAVV Indore

FORWARD

I feel immense pleasure to forward the project entitled "**Health Insurance Cost Prediction Using Regression Models**", submitted by Miss Muskan Thakur of the School of Statistics, DAVV Indore.

Dr Snighdha Banerjee

Head of a Department

School of Statistics

DAVV, Indore

DECLARATION

I, the undersigned Miss Muskan Thakur declare that the work embodied in this project work hereby, titled "**Health Insurance Cost Prediction Using Regression Models**", forms my own contribution to the project work carried out under the guidance of Dr Snigdha Banerjee and has not been previously submitted to any other University for any of this Degree to this or any of this University.

Wherever reference has been made to previous works of this, it has been clearly indicated as such and included in the bibliography.

I, hereby further declare that all information of this project has been obtained and presented in accordance with academic rules and ethical conduct.

Date: **Muskan Thakur**

CERTIFICATE

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Master of Science in statistic at school of statistics-devi Ahilya Vishwavidyalaya, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: Muskan Thakur

Date:

Guided by:

Dr. SNIGDHA BANERJEE

Professor

School of statistics

D.A.V.V Indore

Acknowledgements

This project would not have been possible without the support of my family, professors and my friends.

Dr. SNIGDHA BANERJEE : My supervisor for this project has given so many valuable suggestions on implementations and correcting me on cases where I had deviated from the project. He had always given me a helping hand when I was confused or and pushed me in the right direction.

Signed: Muskan Thakur

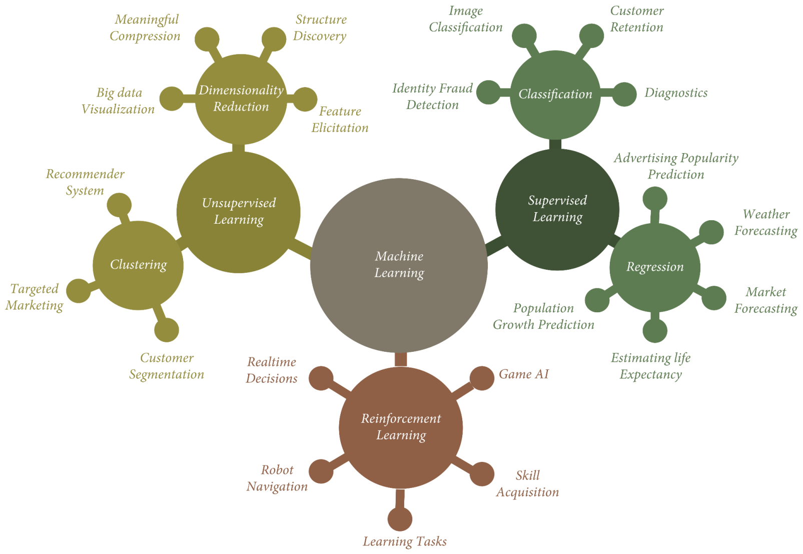
Table of Contents

ABSTRACT

India's government spends 1.5 percent of its annual GDP on public healthcare, which is significantly less than that of other countries. Global public health spending, on the other hand, has almost doubled in line with inflation in the last two decades, reaching US$ 8.5 trillion in 2019, or 9.8% of global GDP. Multinational multi-private sectors provide around 60% of comprehensive medical treatments and 70% of out-patient care, which charge patients astronomically high fees. Because of the rising expense of quality healthcare, increased life expectancy, and the epidemiological shift toward non-communicable diseases, health insurance is becoming an essential commodity for everyone. Insurance data has increased dramatically in the last decade, and carriers now have access to it. The health insurance system explores predictive modelling to boost its business operations and services. Computer algorithms and Machine Learning (ML) is used to study and analyse the past insurance data and predict new output values based on trends in customer behaviour, insurance policies, and data-driven business decisions, and support in formulating new schemes. Additionally, ML has found enormous and potential applications in the insurance industry. Thus, this paper develops a real-time insurance cost price prediction system named ML Health Insurance Prediction System (MLHIPS) using ML algorithms which will aid the insurance companies in the market for easy and rapid determination of values of premiums and thereby curb down health expenditure. The proposed model incorporates and demonstrates different models of regression such as Ridge Regression, Lasso Regression, Simple Linear Regression, Multiple Linear Regression to anticipate insurance costs and assess model outcomes.

1 INTRODUCTION

Public health is an integral part of the society, of the country and of the world we live in and is an important matter of concern. Human lives and public health may be endangered by natural calamities, global epidemic and pandemics, global crisis of medical aids, etc. which increases the vulnerability of public health. People can meet with unavoidable and unforeseen circumstances at any point of their lifetime. Individuals, families, companies and properties are uncovered and uninsured to diverse hazard forms, natural calamities, and the likelihood can shift. These perils include the possibility of mortality, health, and property disaster or resource depletion. People's lives revolve around two main elements: life and prosperity. However, keeping a safe and sound distance from unforeseen events is impossible. The Government of India spends 1.5% of GDP on public health, the lowest level globally [1]. In the last two decades, universal health care has almost doubled, surpassing US $ 8.5 trillion in 2019, or 9.8% of global GDP [2]. The multi-private sector having state-of-the-art facilities provides almost 60% of total hospitalizations and 70% outpatient services [3]. Thus, Health insurance is becoming an essential commodity for every individual because of the increasing cost of quality healthcare combined with higher life expectancy and widespread transition towards non-communicable diseases. To alleviate these types of problems, the world of funds has developed a variety of tools to protect people and organisations from such unseen catastrophic situations, through the utilisation of monetary capital to repay and compensate them. In this manner, insurance is an arrangement that diminishes or evacuates misfortune costs brought about by different dangers. Today, data has evolved dramatically since the recent past decade and insurance carriers have access to it. The health insurance system is exploring ways to use predictive modelling to boost their business operations and services. Computer algorithms and Machine Learning (ML) are used to study and analyse the historical insurance data and predict new output values based on trends in customer behaviour, insurance policies and data-driven business decisions, and support the formulation of new schemes. Besides, most insurance companies use conventional databases to store their data which is primarily structured data. Moreover, merely 10-15 percent of the total data available is processed for gaining insights. Thus, transformation of the data is necessary to gain valuable insights that may be very crucial for the growth of such companies. Therefore, analysing the structured data as unstructured data for making better decisions requires statistical and Machine Learning (ML) techniques. The main advantage of ML is that it can be effectively applied to a massive volume of structured, semi-structured, or unstructured datasets. The ML model can be used across multiple value chains to understand the weight-age of risk involved, claims made customer behaviour with greater predictive accuracy. ML applications in the health insurance sector include various tasks such as understanding risk tolerance and premium leakage leading to inaccurately pricing of premiums, loss deterrence, claims handling, expense management, subrogation, litigation, and fraud identification. When it comes to the value of health insurance in people's lives, it's vital for insurance firms to be as explicit as possible for measuring the sum secured by this approach and the protection charges which must be paid for it. The calculation of health insurance charges in the traditional the process are a hefty task for the insurance companies. The intervention of humans in this process may sometimes produce faulty or inaccurate results. Additionally, as the data increases manual calculations become lethargic and time consuming. Again, in such scenarios the implementation of ML models can be very beneficial for such companies. Therefore, ML may generalise the exertion or strategy to define such an approach. These models can perform self-learning to predict the cost of insurance using past insurance data of the companies. The model inputs are the main parameters that are utilised to calculate the instalments made. This enables the algorithm to precisely estimate the disbursement of insurance coverage. In this way, the correctness can be progressed with ML. The objective of the proposed model is to perform rapid estimation and prediction of insurance charges at a hospital incurred by a patient, using ML models upon the worldata.AI dataset. Thus, this paper develops a real-time insurance cost price prediction system named ML Health Insurance Prediction System (MLHIPS) using ML algorithms which will aid the insurance companies in the market for easy and rapid determination of values of premiums and thereby curb down health expenditure. The proposed model incorporates and demonstrates different regression models such as Multiple Linear Regression, Ridge Regression, Simple Linear Regression, Lasso Regression and Polynomial Regression to predict the insurance costs and compare the models based on their results.



1.1 Goals of Project

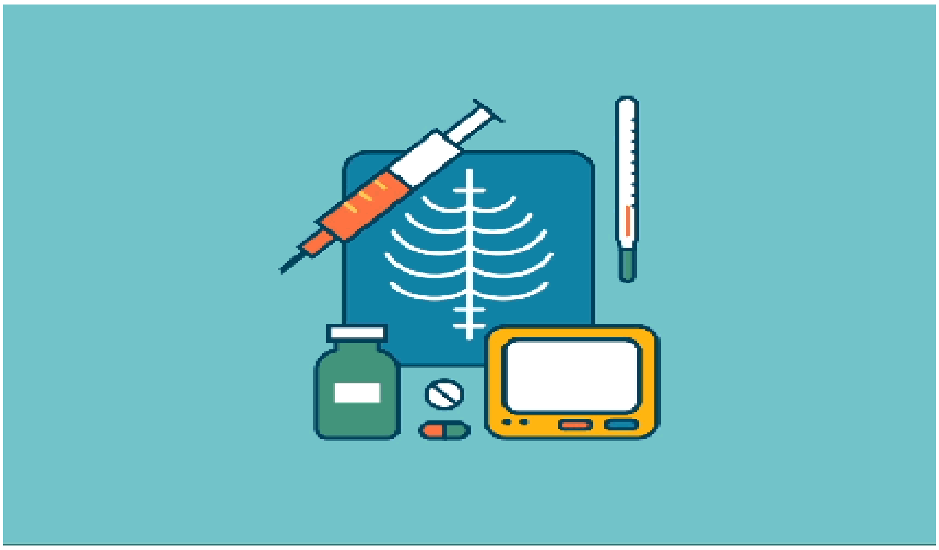
We are tasked with creating an automated system to estimate the annual medical charges for new customers, using information such as their age, sex, BMI, children, smoking habits and region of residence. Estimates from the system will be used to determine the annual insurance premium offered to the customer.

1.2 Motivation

To predict things has never been so easy. I used to wonder how Insurance amount is normally charged. So, in the meantime I came across this dataset and thought of working on it! Using this I wanted to know how few features determine our insurance amount!

1.3 Overview of Approach

The goal of this project is to tell in advance about the health insurance expenses, which could prove to be very beneficial for insurers and patients and this can be achieved by managing assets and selecting appropriate plans. To fulfill this goal, we have already discussed above various techniques of Machine Learning which can be implemented to the dataset, and the analysis of the dataset is also shown in this project. The attributes which play an important part in getting higher precision, are also told in this project. And because of this, a patient need not be tested on all attributes, but only on the ones which play a major role, and thus the expense of the patient can be minimized.



2 BACKGROUND

2.1 Literature Review

Several studies on estimating medical prices have been published in the health field in different contexts [4], [5], [6], [7], [8]. Machine learning has many probable assumptions, but its performance relies on picking a nearly precise algorithm for the specified problem domain and following the appropriate procedures to build, train, and deploy the model. Moran et al. [4] “utilized a comprehensive linear regression method which was used to predict the cost of an intensive care unit (ICU) using patient profile data, DRGs (diagnosis-related groups), the length of time spent in the hospital, and additional traits as features.” Sushmita et al. [5] “proposed a model based on the medical and past expense history of a person to predict his/her future medical costs. Quarterly projected expenses for the future 3,6,9 and 12 months were estimated with the use of the model. They have used random forest and linear regression analysis to predict the costs.” Lahiri et al. [6] “used a classification algorithm to predict whether an individual's medical expenses would increase in the next year, taking into account the medical expenses of the previous year.” Gregori et al. [7] have used the logit model and the OLS method to study the multivariate modelling of healthcare costs data. Bertsimas et al. [8] use data mining techniques, explicitly clustering algorithms and classification trees, and insurance claim data of nearly 500,000 members throughout a three-year period. Based on the data gathered from the medical expenses from the first two years, a justified third-year health-care cost projection is made.

2.2 Health Insurance Cost Prediction

Everyone's existence revolves around their health. Good health refers to a person’s capability to deal with the surroundings on a physical, emotional, mental, and social level. Everyone's existence is generally best unless some sort of health hassle arises that is uncertain and can not be expected before it happens. needs which include the desire of owning a house or a car or some other device of social popularity or different consumer durables of comfort can be postponed if the circle of relatives has inadequate savings and poor assets of income. However, this is not the case with the unexpected medical feature which wishes immediate money, useful resources and impacts the financial savings of the own family. economic stress on medical grounds can certainly smash the long-term monetary dreams of a family which may consist of education or marriage of children and retirement plans besides goals stated supra. One can also wonder about a solution to conquer this sort of crisis and the solution to that is none apart from health insurance which will assist in the protection of the good health of an individual and their own family without growing any possibility of a monetary crisis and disturbing monetary stability. Health insurance is a product of preferred coverage that covers fees associated with the medicine and surgical procedure of an insured which could be an individual, family, or a collection of people. It's an association in which a person, circle of relatives, or a group purchases medical health care coverage in advance with the aid of payment of a fee known as a top class. In other phrases, health insurance is an arrangement that enables to delay, defer, lessen or avoid charges related to the medical expenses of an insured. The insurer will either make certain cashless remedies of medical ailments or offer repayment of medical costs incurred under the policy in any of the network hospitals throughout the country.

2.3 Regression Models

The regression techniques used are the statistical methods that establish the association between a target or dependent variable and a set of independent or predictor variables. It assumes that both the target and the predictor variables are having numerical values and there exists some kind of correlation between the two. The models that we are implementing in our problem are discussed below,

1) **Simple Linear Regression:**

In simple linear regression[16], the target variable(Y) is dependent on a single independent variable(X) and the model establishes a linear relationship among these two variables.

The linear regression model tries to fit the regressor line between the independent(X) and dependent(y) variable.

The equation of the line is given by:
$Y = a + bX$

(1) where "a" and "b" are the models parameters called as regression coefficients, "a" is the value of the Y intercept that the line makes when X is equal to zero and "b" is the slope that signifies the change of Y with the change of X. More the value of "b"

means a small change in X causes a significant change in Y, and vice versa. The value of “a” and “b” can be found by Ordinary Least Square method. In linear regression models the values predicted may not be accurate always, there will always be some difference hence we add an error term to the original equation (1) that accounts for the difference and thus help in making better predictions.

$$Y = a + bX +$$

(2) Assumptions in Linear Regression

- The sample size of data should exceed the number of available parameters.
- Only over a restricted range of data the regression can be valid.
- Error term is normally distributed. This also means that the mean of the error has expected value of 0.

2) **Multiple Linear Regression:**

Similar to simple linear regression, multiple regression [17] is a statistical procedure that examines the degree of association between a set of independent variables and a dependent variable. There is just one independent and one dependent variable in basic linear regression, but there are numerous predictor variables in multiple linear regression and the value of dependent variable(Y) is now calculated depending on the values of the predictor variables. it is assumed that there is no dependency among the predictor variables. Suppose if the target value is dependent on “n” independent variables then the regression fits the regression line in a N dimensional space. The regressor line equation is now modified into

$$Y = a + b1X1+ b2\ X2+ b3X3+ \ldots.. + bn\ Xn \qquad (3)$$

where “a” is the Y-intercept value and < b1, b2, b3,…., bn > are the regression coefficients associated with the n independent variables and is the error term.

3) **Polynomial Regression:**

Polynomial Regression [18] is yet another a special case of linear regression. In Linear regression the model tries to fit a straight regression line between the dependent and independent variable. In scenarios where there doesn’t exist linear relationship between the target and predictor variable then instead of a straight line a curve is being fitted against the two variables. This is accomplished by fitting a polynomial equation of degree n on the non-linear data which establishes a curvilinear relationship among the dependent and independent variables. In polynomial regression the assumption that the independent variables must be independent of each other is not mandatory. The equation of the line thus reduces to:

$$Y = a + b\ x1X1+ b2X2+ b3X3+ \ldots. + bnXn + \qquad (4)$$

The following are some of the benefits of applying polynomial regression:

• Polynomial Regression offers the best estimate of the relationship between the dependent and independent variable.
• Higher degree polynomial generally provides a good• Polynomial Regression basically fits a wide range of curves of varying degree to the dataset.

Drawbacks of applying Polynomial Regression:

• These are too sensitive to the existence of outliers in the dataset, as outliers cause the model's variance to rise. When the model comes across an unknown data item, it Underperforms.

4) **Ridge Regression:**

Ridge regression [19] is a standard model tuning process used to analyse the data suffering from multicollinearity. This is a way to approximate the coefficients of the regression model when the independent variables are firmly related or there exists an association between them. Ridge Regression's main objective is to take the dataset and fit a new line into it in a way that does not overfit the model. For this purpose, ridge regression adds an insignificant amount of bias that determines the fitting of the line into the data. We obtain a substantial reduction in variance, which leads to an increase in the accuracy value by the addition of bias. The least Square determines the values of the parameters for the equation (1), which diminishes the sum of squared residuals. But in contrast the Ridge Regression regulates the value for parameters that results in minimization of the sum of squared residuals along with an additional term λ*b2. Ridge Regression performs L2 regularization. (5)
where y* = a + bX is the predicted value. In this ridge regression method, the coefficients are penalised by a value lambda, this acts as a control parameter, which determines how severe is the penalty and how much significance should be given to Xi. The higher the values of the lambda the bigger is the penalty and therefore the magnitude of coefficients is reduced. When the slope (b) of the line is steep then the target variable(Y) is very sensitive to relatively small change in the predictor variable variable(X). In ridge regression by the addition of the lambda value the sensitivity decreases. If lambda is zero then ridge regression reduces to linear regression and when lambda increases gradually the slope the line decreases asymptotically. To know which value of lambda is to choose we try different values of lambda and use cross validation to determine which one result in the lowest variance.

5) **Lasso Regression:**

Least Absolute Shrinkage and Selection Operator (LASSO) [20] is similar to ridge regression that penalizes the regression model. It performs L1 regularization. The Lasso regression process is usually used in machine learning for the selection of the significant subset of variables. The prediction accuracy of this model is higher when compared to other model interpretations. Like ridge regression lasso regression also results in a line with little amount of bias added to it which thereby decreases the variance of the model (6)

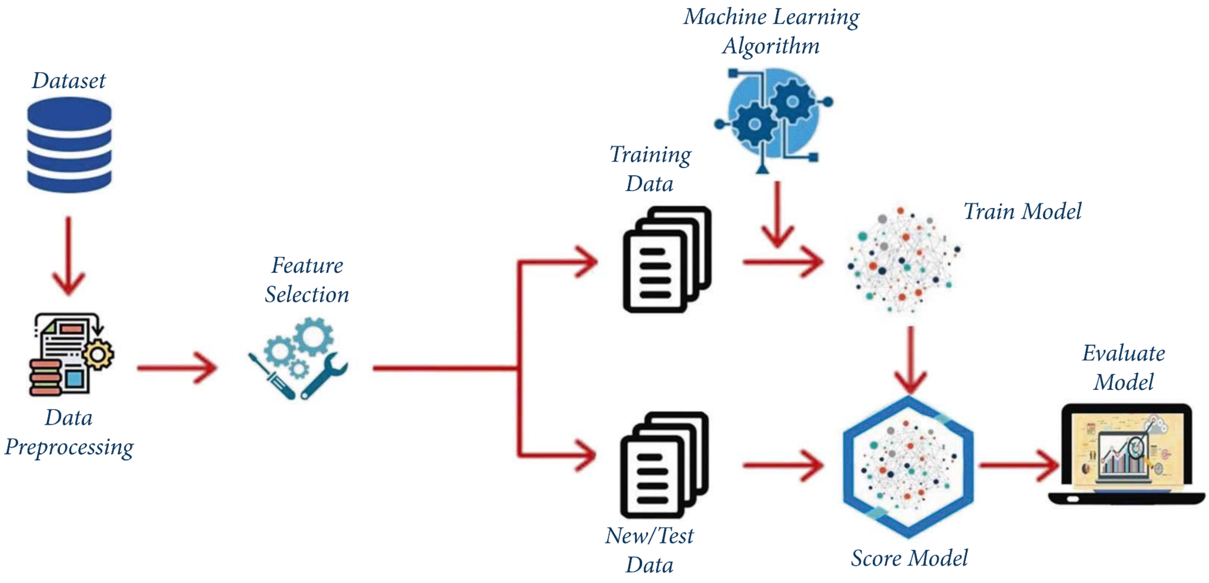
The major difference between lasso and ridge regression is, ridge regression decreases the slope asymptotically close to zero but lasso regression can reduce the slope all the way down to zero, thereby eliminating the useless parameters from the line equation that don’t have any significant role for predicting the value of the target variable.
Lasso regression usually works better under conditions where some predictors have high coefficients, and the rest have low coefficients. Ridge regression performs better when the result is a function of many predictors, all of which have coefficients of approximately the same size.

6) **Random Forest Regressor**

Random forest regressor creates a forest of decision trees. In a random forest, regressor output no longer relies upon one individual tree. every tree in the forest will give its output class and the class with the highest votes will be the final output. It is a bagging model which uses bootstrap sampling to train the model and is typically used to reduce the variance of models. Random forest algorithms can be used for both regression and classification and we also get the satisfactory result when we use a random forest regression model.

3 METHODOLOGY

We have performed machine learning techniques on medical insurance data. The medical insurance cost dataset is gained from worldData.AI, and we performed the data preprocessing. After preprocessing, we select the features by performing feature engineering. Then, the dataset is split into two parts, train and test datasets; about 70% of the total data are used for training, while the rest is for testing. The training dataset is used to create a model that predicts medical insurance costs for the year, while the test dataset is used to evaluate the regression models. For regression exploring the dataset, then categorical values are converted to numerical values. The steps of our working methodology are shown in Figure .



Pandas

Pandas is an open source, Python licensed library that offers high-performance, easy-to-use data structures, and data analysis tools to the Python programming language. The Data Frame is the core data structure. Data frame allows tabular data to be stored and manipulated in observation rows and variable columns. A broad variety of stored data types is available, such as CSVs, TSV's (Tab separated values), JSONs (Hypertext Mark-Up Language), and more.

Pandas can read different types. A Data Frame consists of both a row and a column index, a two-dimensional set of values. A series is a special collection of index values.

In our project, we have converted dataset CSV files to data frames:

1. age
2. sex
3. bmi
4. children
5. smoker
6. region
7. charges

Numpy

NumPy is an array-processing application for general purposes. It stands for 'Numerical Python'. It is a library of multidimensional array objects, and a set of array processing routines. NumPy has functions built in for linear algebra and the generation of random numbers.

Matplotlib

Is the art of displaying data through charts, icons, presentations and more. It is most common to translate complex data for a non-technical audience into comprehensible insights. Matplotlib is one of the most powerful Data Visualization Python packages used. This is a cross - platform framework designed to make Two dimensional graphs from records in arrays. This also provides an object-oriented API which helps, for example, to embed plots into implementations using Python GUI toolkits such as PyQt.

Seaborn

Seaborn is an enhancement to matplotlib and not a substitution for it. The reason for this is that it is placed on top of matplotlib and you will often explicitly invoke matplotlib functions to draw simpler plots already available through the namespace pyplot. Matplotlib is completely scalable but it can be difficult to know what settings to change to achieve an appealing plot. Seaborn comes with a number of custom themes to track the matplotlib look and a high level user interface. It is closely integrated with the PyData stack, including support of SciPy and stats models, data structures for NumPy and Pandas, and statistical routines.

sklearn.preprocessing

The sklearn.preprocessing package provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

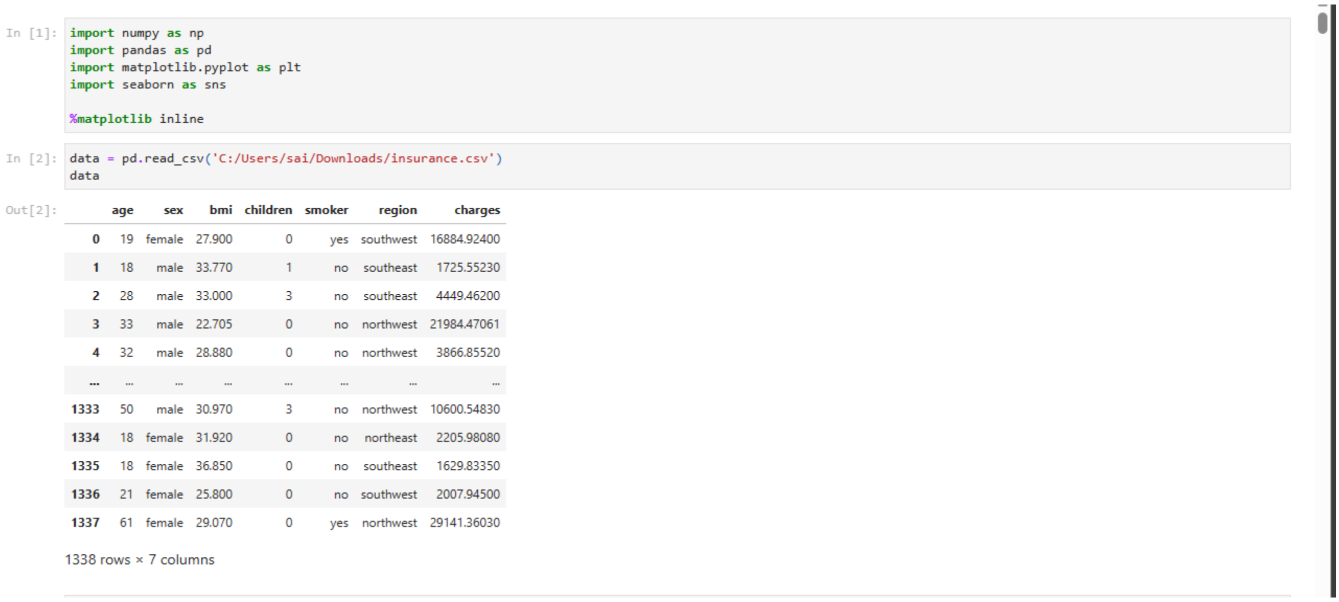
In general, learning algorithms benefit from standardization of the data set. If some outliers are present in the set, robust scalers or transformers are more appropriate. The behaviors of the different scalers, transformers, and normalizers on a dataset containing marginal outliers is highlighted in Compare the effect of different scalers on data with outliers.

sklearn.ensemble.RandomForestRegressor

A random forest is a meta estimator that fits a number of classifying decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is controlled with the max_samples parameter if bootstrap=True (default), otherwise the whole dataset is used to build each tree.

3.1 Dataset Collection

In this study we have used the worlddata.ai dataset of Medical Insurance cost and trained our model on attributes like age, sex, BMI, children. smoker, region which contains 1338 training instances and 267 testing instances.



```
In [1]: import numpy as np
        import pandas as pd
        import matplotlib.pyplot as plt
        import seaborn as sns

        %matplotlib inline
```

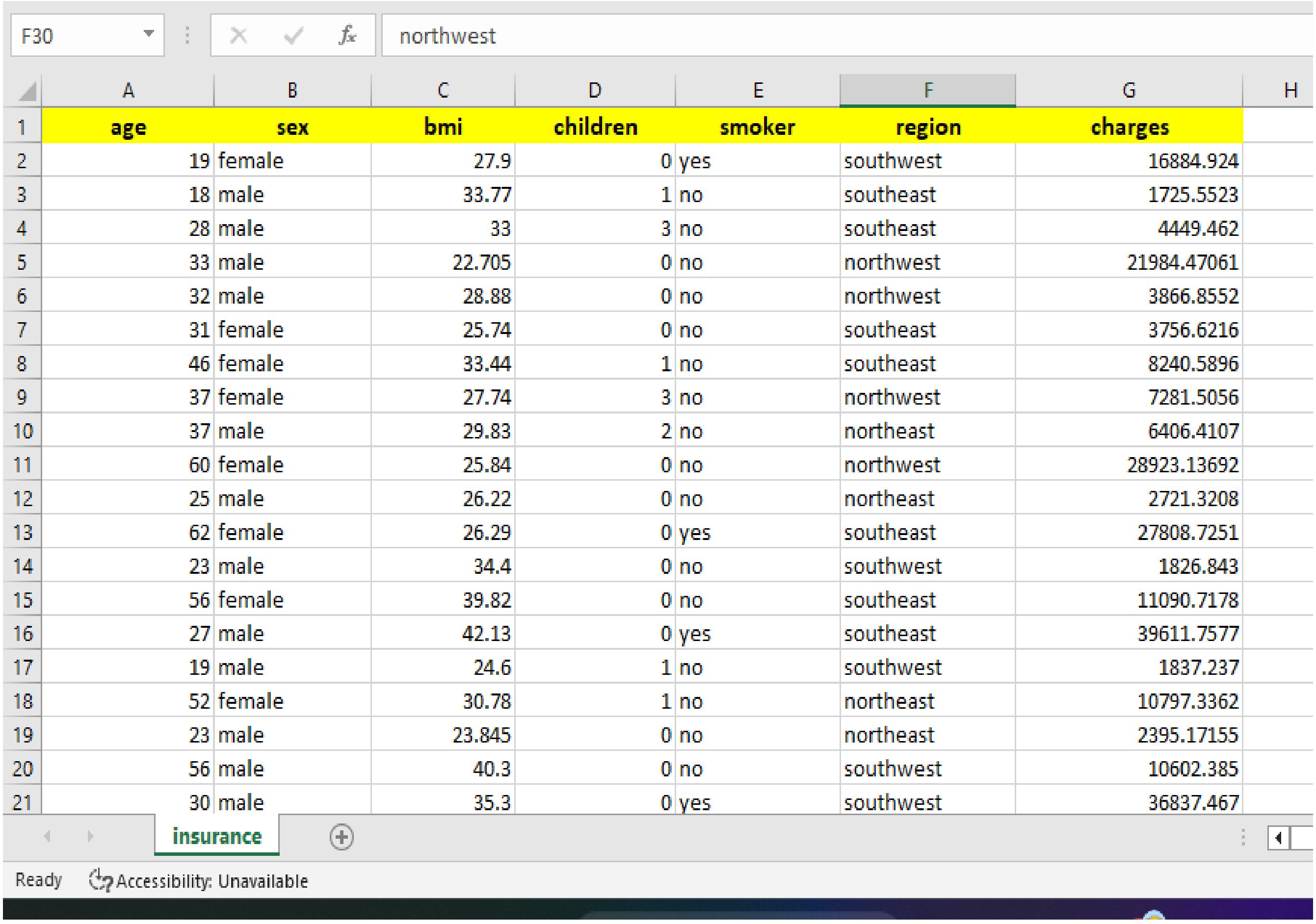
```
In [2]: data = pd.read_csv('C:/Users/sai/Downloads/insurance.csv')
        data
```

```
Out[2]:
         age     sex     bmi  children  smoker     region      charges
   0      19  female  27.900         0     yes  southwest  16884.92400
   1      18    male  33.770         1      no  southeast   1725.55230
   2      28    male  33.000         3      no  southeast   4449.46200
   3      33    male  22.705         0      no  northwest  21984.47061
   4      32    male  28.880         0      no  northwest   3866.85520
 ...     ...     ...     ...       ...     ...        ...          ...
1333      50    male  30.970         3      no  northwest  10600.54830
1334      18  female  31.920         0      no  northeast   2205.98080
1335      18  female  36.850         0      no  southeast   1629.83350
1336      21  female  25.800         0      no  southwest   2007.94500
1337      61  female  29.070         0     yes  northwest  29141.36030

1338 rows × 7 columns
```

You can see all the attributes in the following table:

Name	Description
Age	Customer's Age
BMI	Body mass index of the customer
Number of kids	Number of kids of the customer
Gender	Male / Female
Smoker	Whether the customer is smoker or not
Region	Where the customer lives: southwest, southeast, northeast, northwest
Charges (target variable)	Medical fee the customer has to pay

3.2 Understanding Dataset



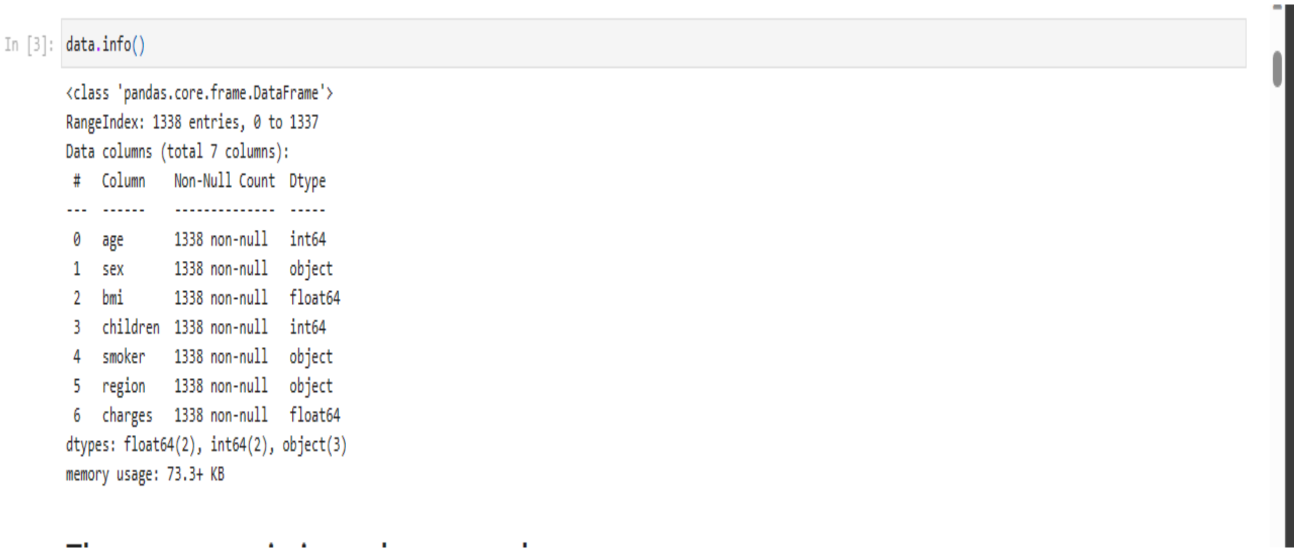
age	sex	bmi	children	smoker	region	charges
19	female	27.9	0	yes	southwest	16884.924
18	male	33.77	1	no	southeast	1725.5523
28	male	33	3	no	southeast	4449.462
33	male	22.705	0	no	northwest	21984.47061
32	male	28.88	0	no	northwest	3866.8552
31	female	25.74	0	no	southeast	3756.6216
46	female	33.44	1	no	southeast	8240.5896
37	female	27.74	3	no	northwest	7281.5056
37	male	29.83	2	no	northeast	6406.4107
60	female	25.84	0	no	northwest	28923.13692
25	male	26.22	0	no	northeast	2721.3208
62	female	26.29	0	yes	southeast	27808.7251
23	male	34.4	0	no	southwest	1826.843
56	female	39.82	0	no	southeast	11090.7178
27	male	42.13	0	yes	southeast	39611.7577
19	male	24.6	1	no	southwest	1837.237
52	female	30.78	1	no	northeast	10797.3362
23	male	23.845	0	no	northeast	2395.17155
56	male	40.3	0	no	southwest	10602.385
30	male	35.3	0	yes	southwest	36837.467

The dataset contains seven variables, as shown in the table above. While calculating the cost of the Charges of a customer which is our target variable the values of the rest six of the variables are taken into consideration. In this phase, the data is reviewed, properly reconstructed, and properly applied to machine learning algorithms. The dataset was first checked for missing values. The dataset was found containing missing values in the bmi and charges columns. The missing values were imputed by the mean values of the respective attribute Values. As regression models accept only numerical data, the categorical columns in our case the sex, smoker and region columns containing categorical columns were converted into numerical values using label encoding. Then the updated dataset was partitioned into training and testing dataset. And the model was trained using the training dataset.

3.3 Pre-processing Data set

To work with missing values

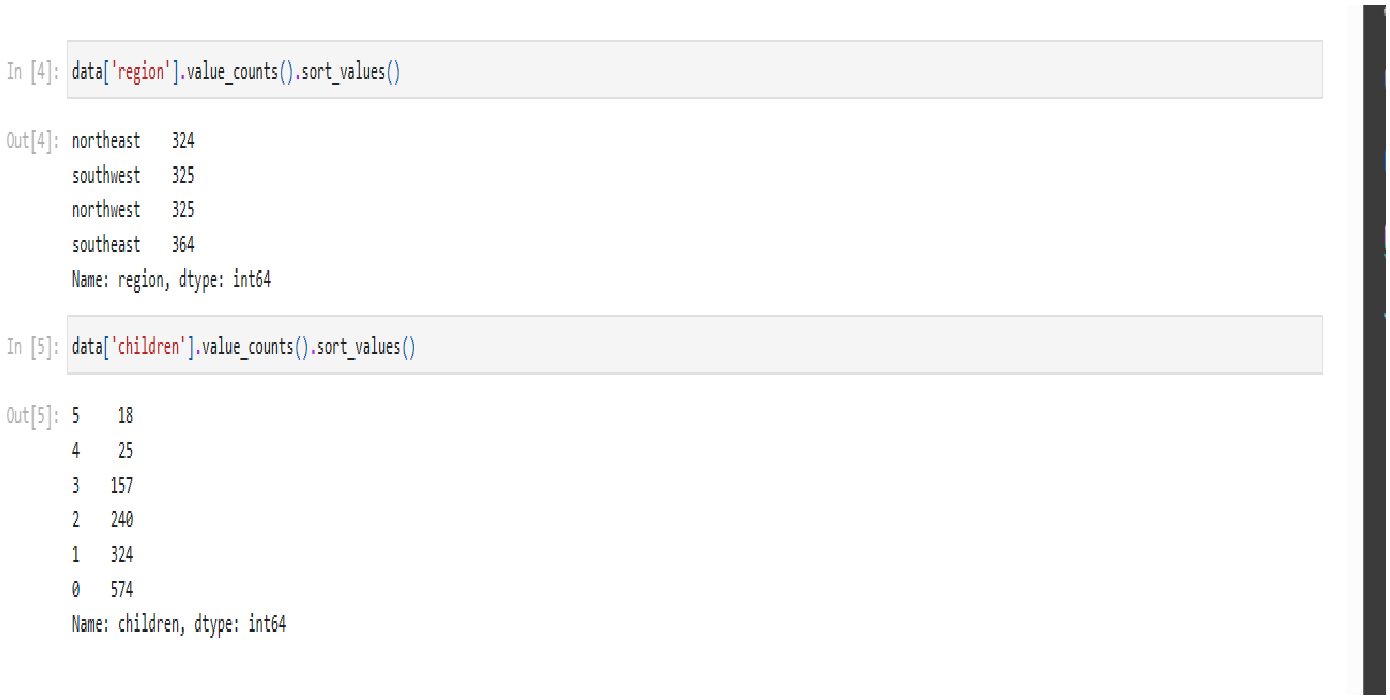
To replace nulls with non-null values, a technique known as imputation has been used. Let's calculate the total number of nulls in each column of our dataset. The first step is to check which cells in our Data Frame are null by using .isnull() and then to count the number of nulls in each column we use an aggregate function for summing .isnull.sum()



```
In [3]: data.info()
```

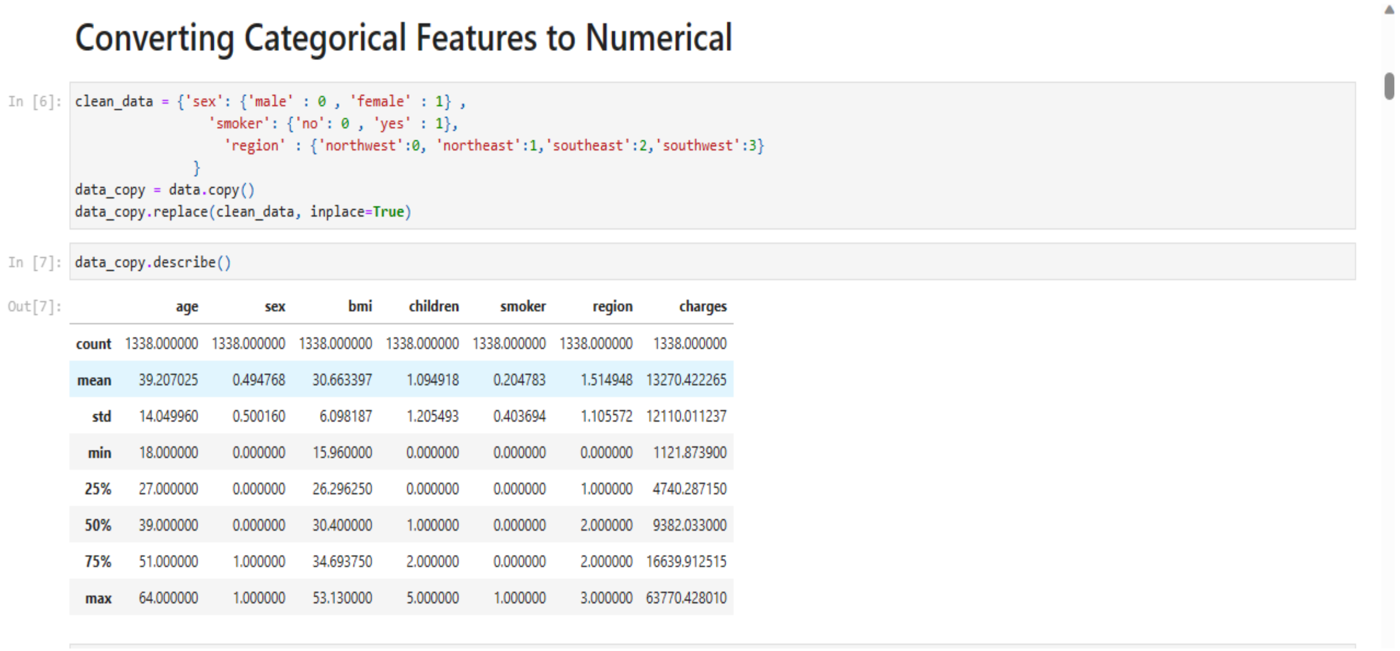
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1338 entries, 0 to 1337
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       1338 non-null   int64
 1   sex       1338 non-null   object
 2   bmi       1338 non-null   float64
 3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

1. **There are no missing values as such, so that we will calculate region, children value and sort them.**



```
In [4]: data['region'].value_counts().sort_values()

Out[4]: northeast    324
        southwest    325
        northwest    325
        southeast    364
        Name: region, dtype: int64

In [5]: data['children'].value_counts().sort_values()

Out[5]: 5     18
        4     25
        3    157
        2    240
        1    324
        0    574
        Name: children, dtype: int64
```

2. After that we convert features to numerical value.



Converting Categorical Features to Numerical

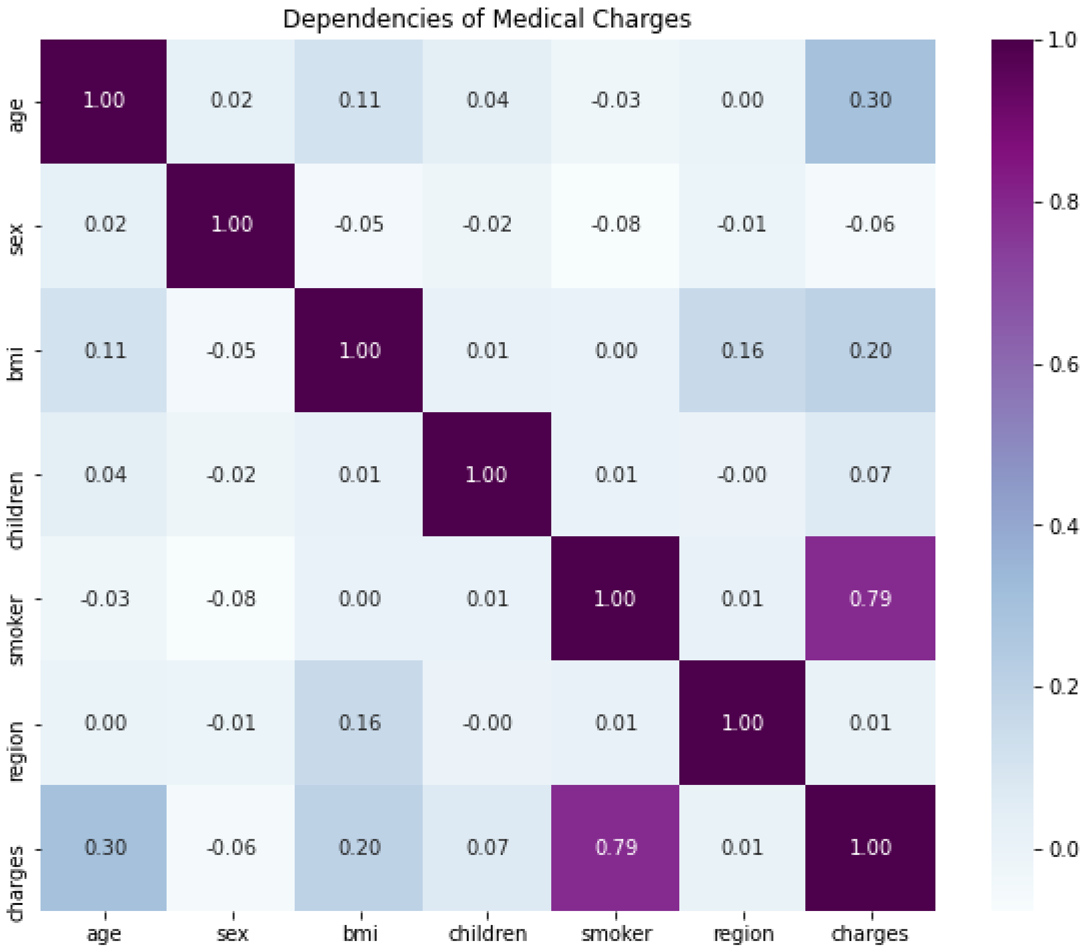
```
In [6]: clean_data = {'sex': {'male' : 0 , 'female' : 1} ,
                         'smoker': {'no': 0 , 'yes' : 1},
                           'region' : {'northwest':0, 'northeast':1,'southeast':2,'southwest':3}
                       }
        data_copy = data.copy()
        data_copy.replace(clean_data, inplace=True)

In [7]: data_copy.describe()
```

Out[7]:

	age	sex	bmi	children	smoker	region	charges
count	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000	1338.000000
mean	39.207025	0.494768	30.663397	1.094918	0.204783	1.514948	13270.422265
std	14.049960	0.500160	6.098187	1.205493	0.403694	1.105572	12110.011237
min	18.000000	0.000000	15.960000	0.000000	0.000000	0.000000	1121.873900
25%	27.000000	0.000000	26.296250	0.000000	0.000000	1.000000	4740.287150
50%	39.000000	0.000000	30.400000	1.000000	0.000000	2.000000	9382.033000
75%	51.000000	1.000000	34.693750	2.000000	0.000000	2.000000	16639.912515
max	64.000000	1.000000	53.130000	5.000000	1.000000	3.000000	63770.428010

3. Correlation Matrix

When it comes to machine learning, feature engineering is the process of extracting features from raw data while applying domain expertise in order to improve the performance of ML algorithms. In the medical insurance cost dataset, attributes such as smoker, BMI, and age are the most important factors that determine charges. Also, we see that sex, children, and region do not affect the charges. We might drop these 3 columns as they have less correlation by plotting the heat map graph to see the dependency of dependent value on independent features. The heat map makes it easy to identify which features are most related to the other features or the target variable. Outcomes are shown in Figure

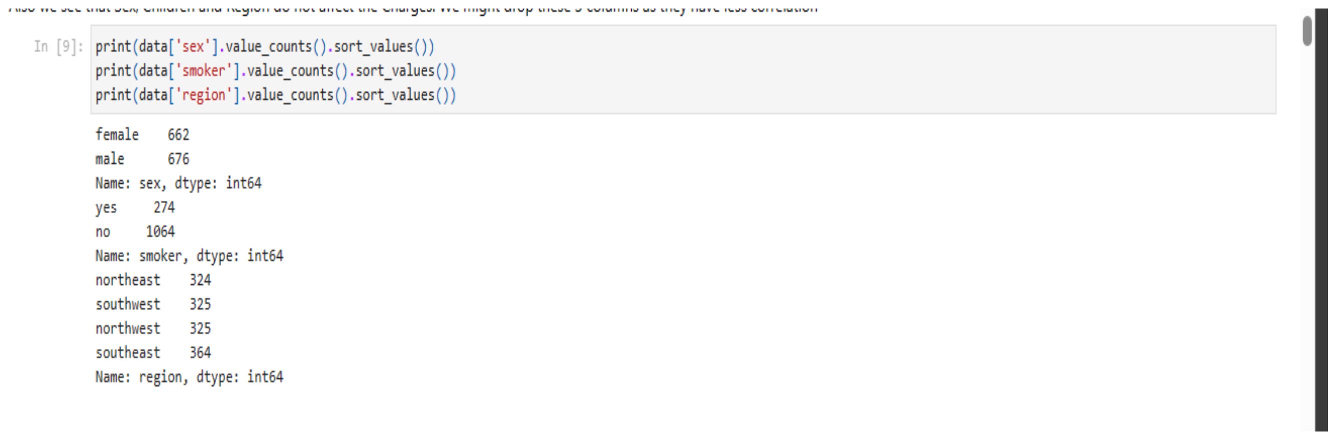


	age	sex	bmi	children	smoker	region	charges
age	1.00	0.02	0.11	0.04	-0.03	0.00	0.30
sex	0.02	1.00	-0.05	-0.02	-0.08	-0.01	-0.06
bmi	0.11	-0.05	1.00	0.01	0.00	0.16	0.20
children	0.04	-0.02	0.01	1.00	0.01	-0.00	0.07
smoker	-0.03	-0.08	0.00	0.01	1.00	0.01	0.79
region	0.00	-0.01	0.16	-0.00	0.01	1.00	0.01
charges	0.30	-0.06	0.20	0.07	0.79	0.01	1.00

Correlation plot of features

Smoker, BMI and Age are most important factor that determnines - Charges

Also we see that Sex, Children and Region do not affect the Charges. We might drop these 3 columns as they have less correlation



Also we see that Sex, Children and Region do not affect the charges. We might drop these 3 columns as they have less correlation

In [9]:
```
print(data['sex'].value_counts().sort_values())
print(data['smoker'].value_counts().sort_values())
print(data['region'].value_counts().sort_values())
```

```
female    662
male      676
Name: sex, dtype: int64
yes     274
no     1064
Name: smoker, dtype: int64
northeast    324
southwest    325
northwest    325
southeast    364
Name: region, dtype: int64
```

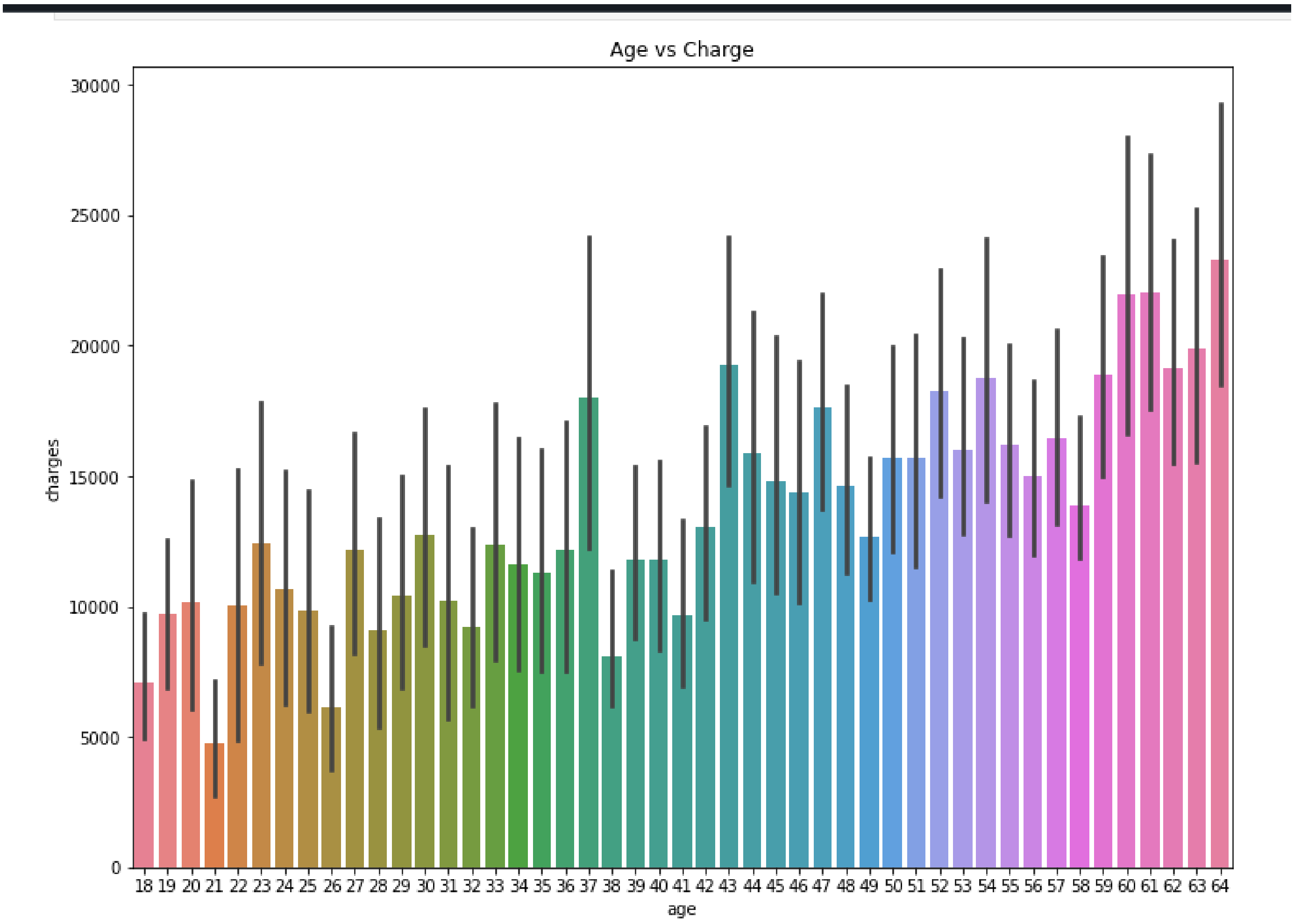
Now we are confirmed that there are no other values in above pre-processed column, We can proceed with EDA

3.4 Exploratory Data Analysis

Distribution of features:

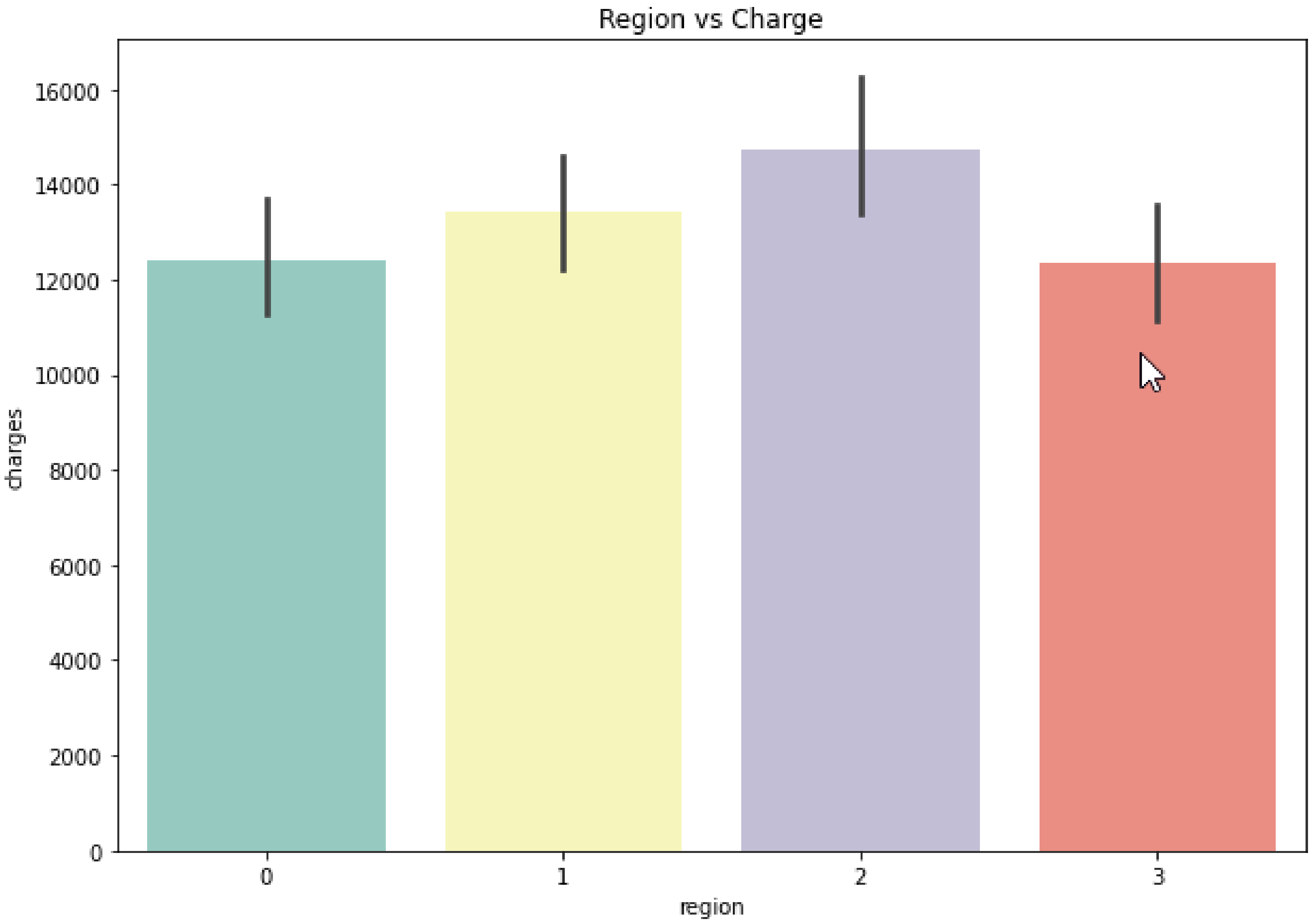
Age vs. Charges

We can see in Figure that with the growing age, the insurance charges are going to be increased. For example, when the age touches 64, the insurance charge is 23000, as shown in Figure . Age is shown on the *x-axis*, and charges are given on the *y-axis*.



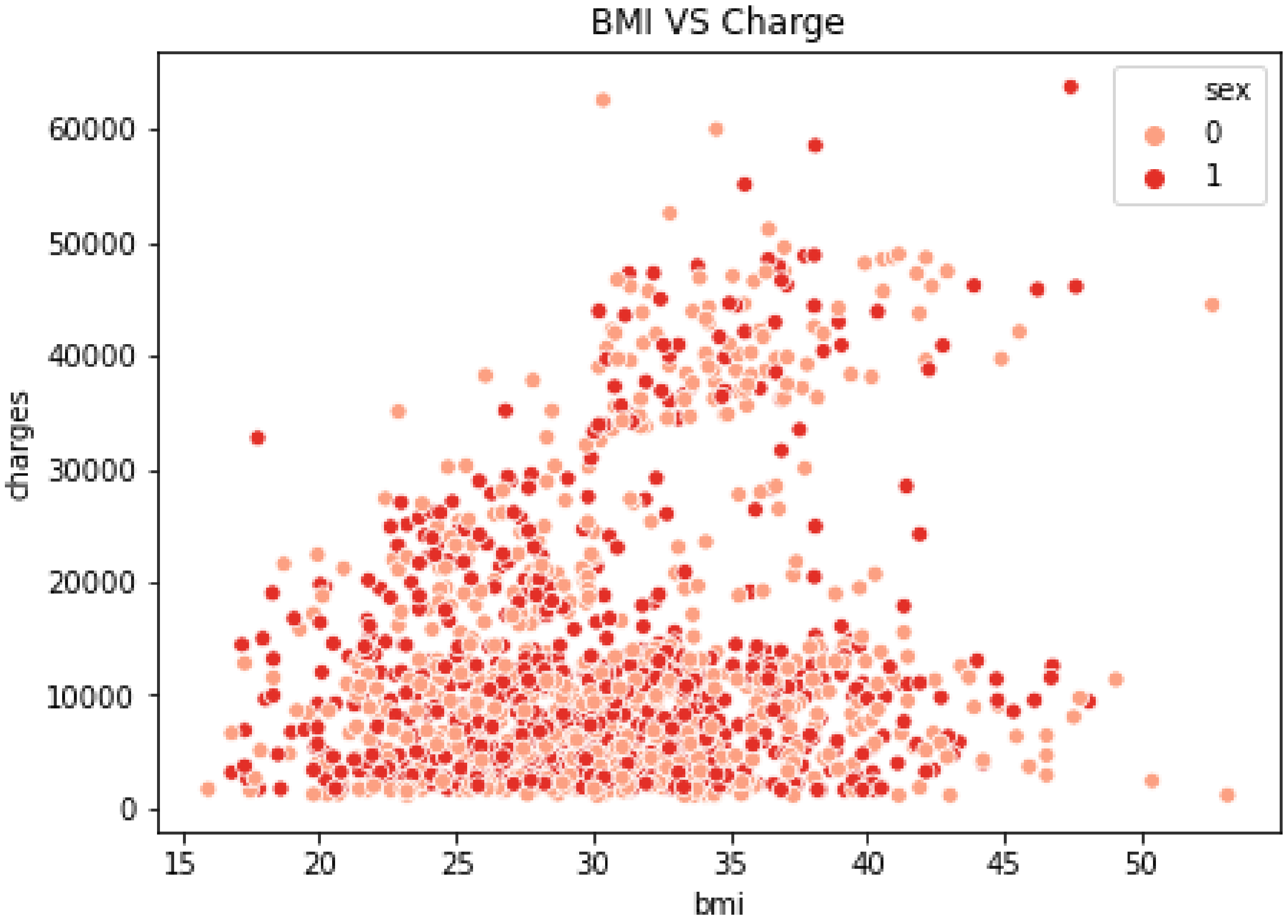
Region vs. Charges

Insurance charges vary concerning certain regions as shown in Figure The health insurance charges in the southeast are greater than in other regions. The region is displayed on the *x-axis*, and charges are shown on the *y-axis*.



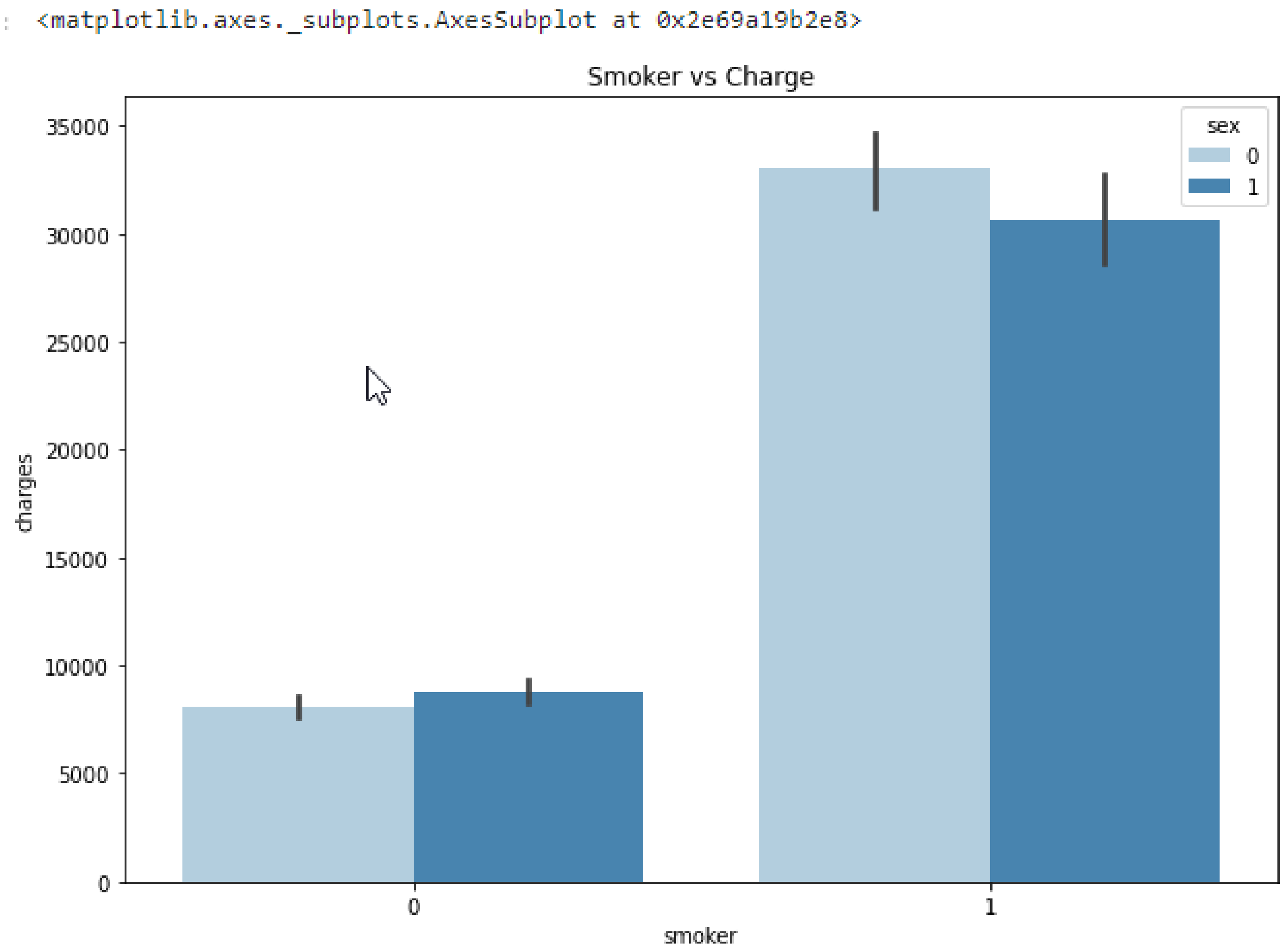
BMI vs. Charges

In Figure , the zero value is used to represent the females and one value is used for the males. The BMI values of sex or gender types (male and female) are given in the *x-axis*, and the charges are presented in the *y-axis*. It can be clearly seen that when the values of BMI are varied, the insurance charges will vary accordingly as shown in Figure .



Smoker vs. Charges

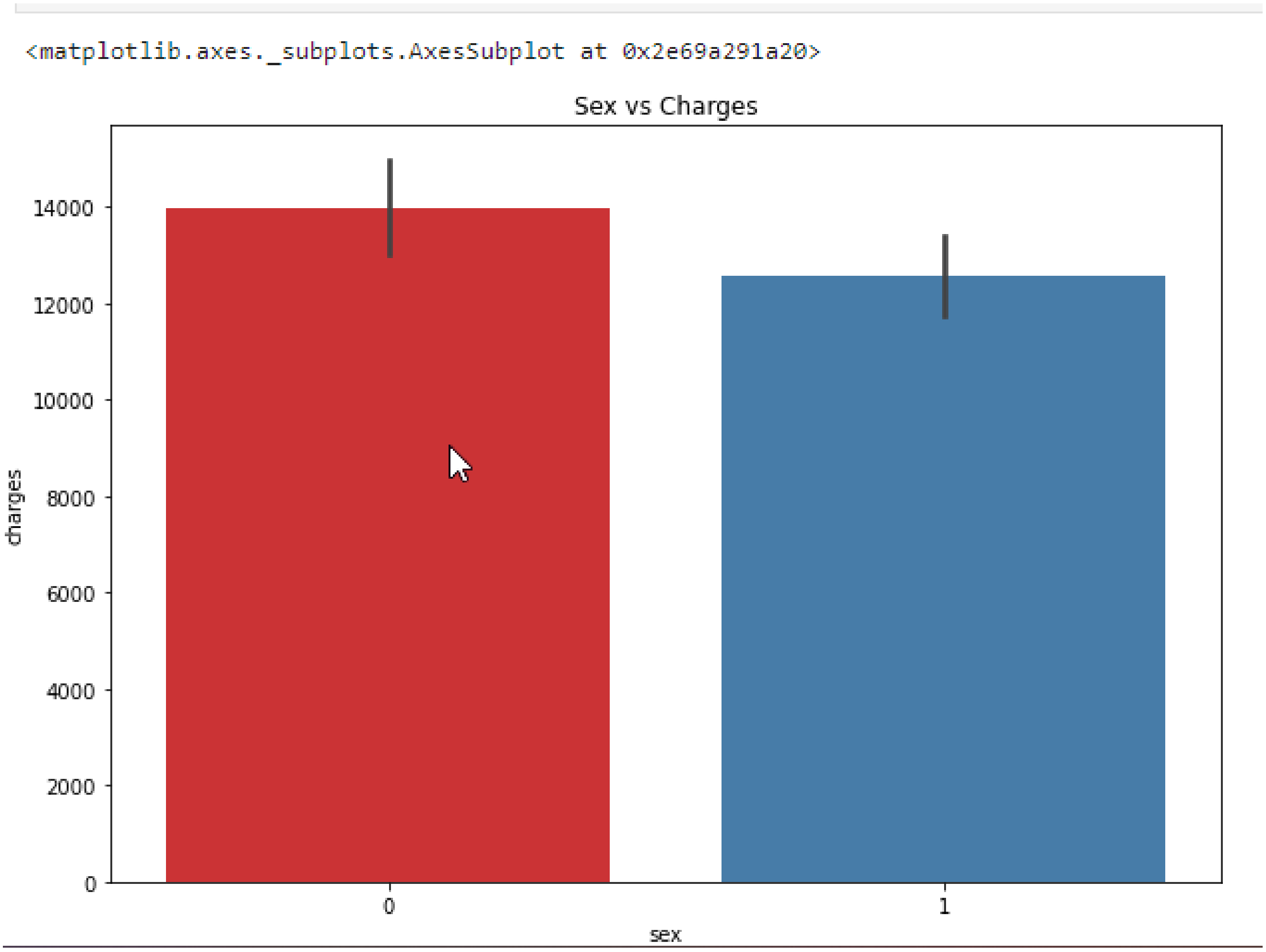
Figure illustrates that as a normal smoker, the medical insurance cost varies slightly. However, men are more addicted and passionate to smoking as compared to women so the health insurance cost for females is greater as compared to the males. We can see in Figure that with the increase of smoking habits, the insurance charges are going to be decreased for men and increased for women. Smokers’ values are shown on the *x-axis*, and charges are shown on the *y-axis*.



```
: <matplotlib.axes._subplots.AxesSubplot at 0x2e69a19b2e8>
```

Sex vs. Charges

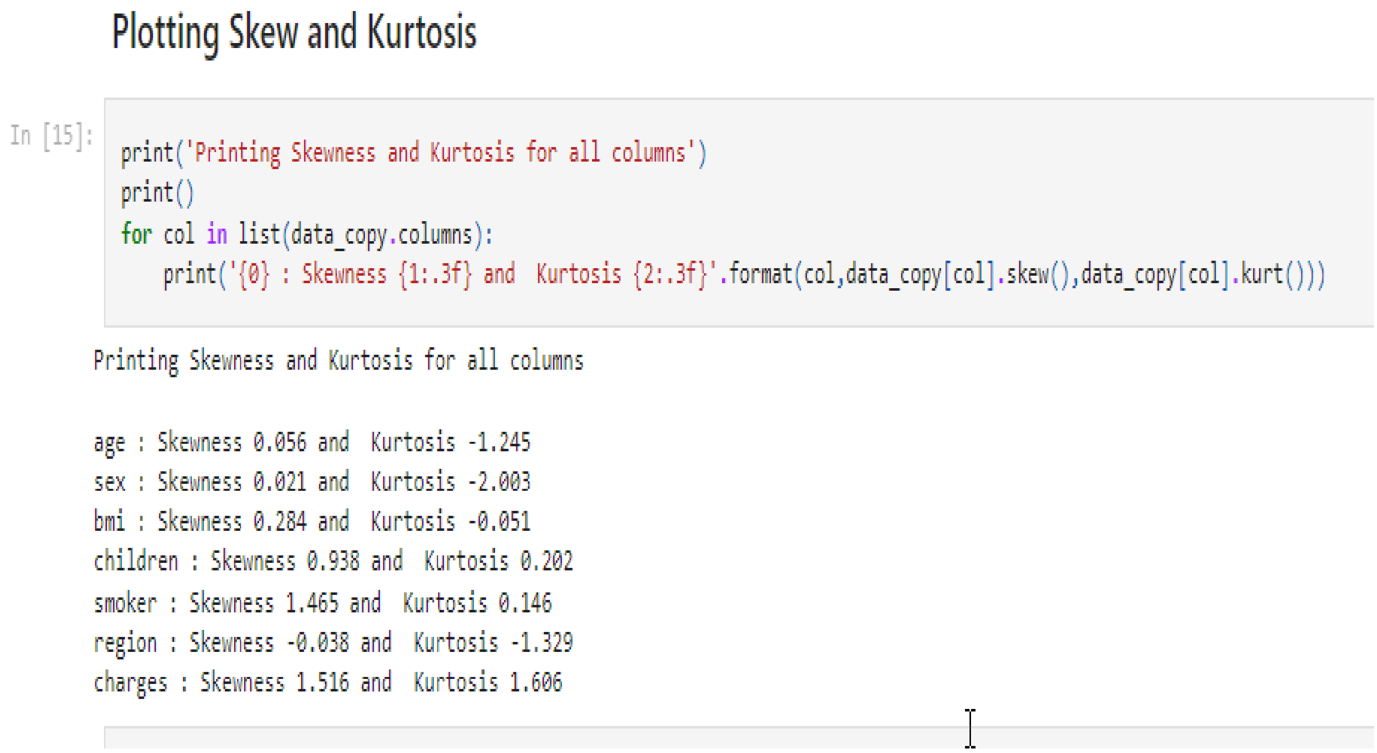
The medical insurance charges for the female gender are always greater than for the male as shown in Figure . It gives the sex types on the *x-axis* and the charges on the *y-axis*. The figure illustrates that the insurance charges for the female are 14000, and for the male, the charges are around 13000.



```
<matplotlib.axes._subplots.AxesSubplot at 0x2e69a291a20>
```

Skew and Kurtosis

Skewness is a metric that quantifies symmetry in a given scenario, or more specifically, the lack of it. If a distribution or data set appears the same on all sides of the graph to the left and right of the centre point, it is said to be symmetric. Kurtosis is a measure of how heavy-tailed or light-tailed the data are when compared to the normal distribution, according to the normal distribution. Heavy tails or outliers are more probable in data sets with a high kurtosis than data sets with a low kurtosis. When there is a low kurtosis in a data collection, it is more likely that there will be no outliers . The most extreme instance would be if there is a uniform distribution. Table below displays the values for the skew and kurtosis of the attributes of a medical dataset.

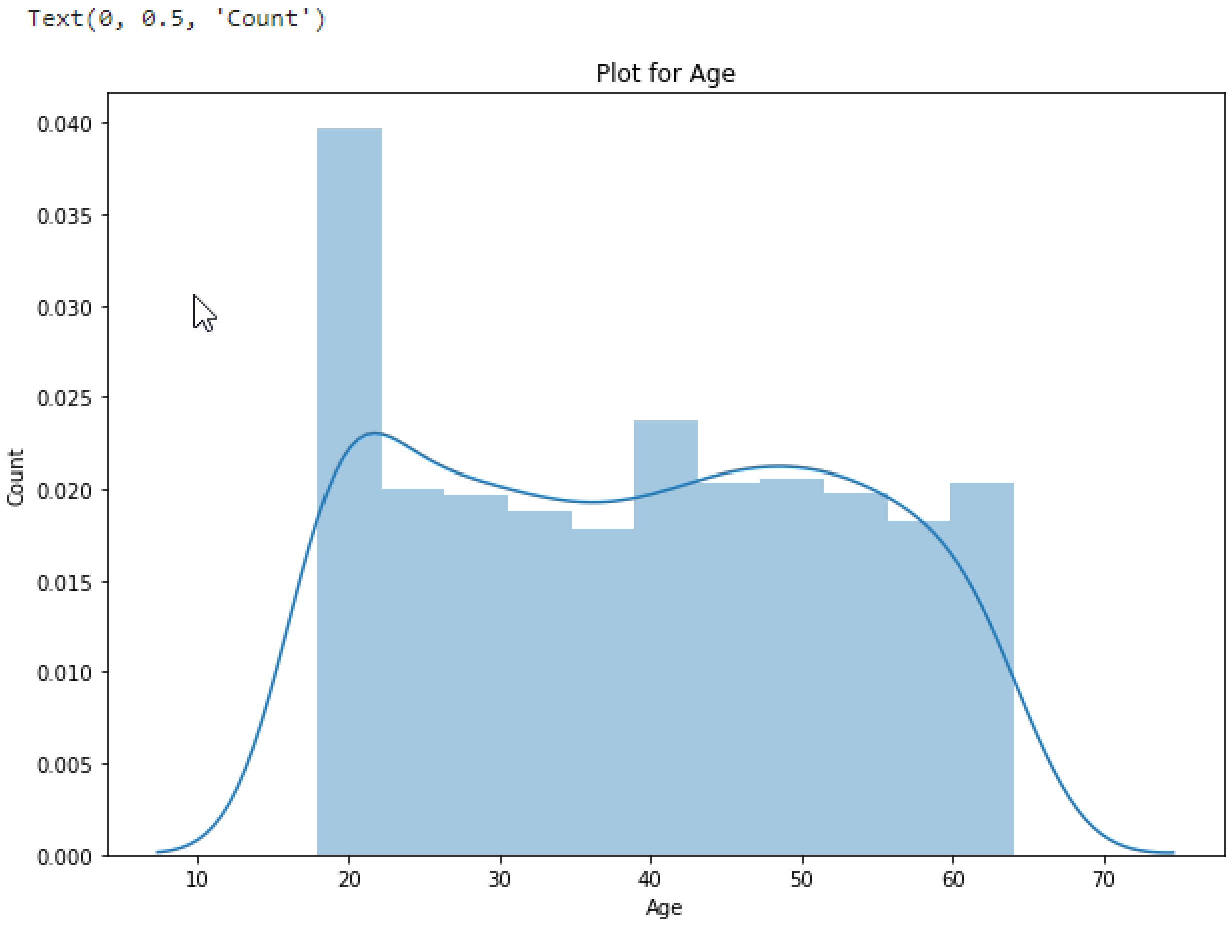


Plotting Skew and Kurtosis

```
print('Printing Skewness and Kurtosis for all columns')
print()
for col in list(data_copy.columns):
    print('{0} : Skewness {1:.3f} and  Kurtosis {2:.3f}'.format(col,data_copy[col].skew(),data_copy[col].kurt()))
```

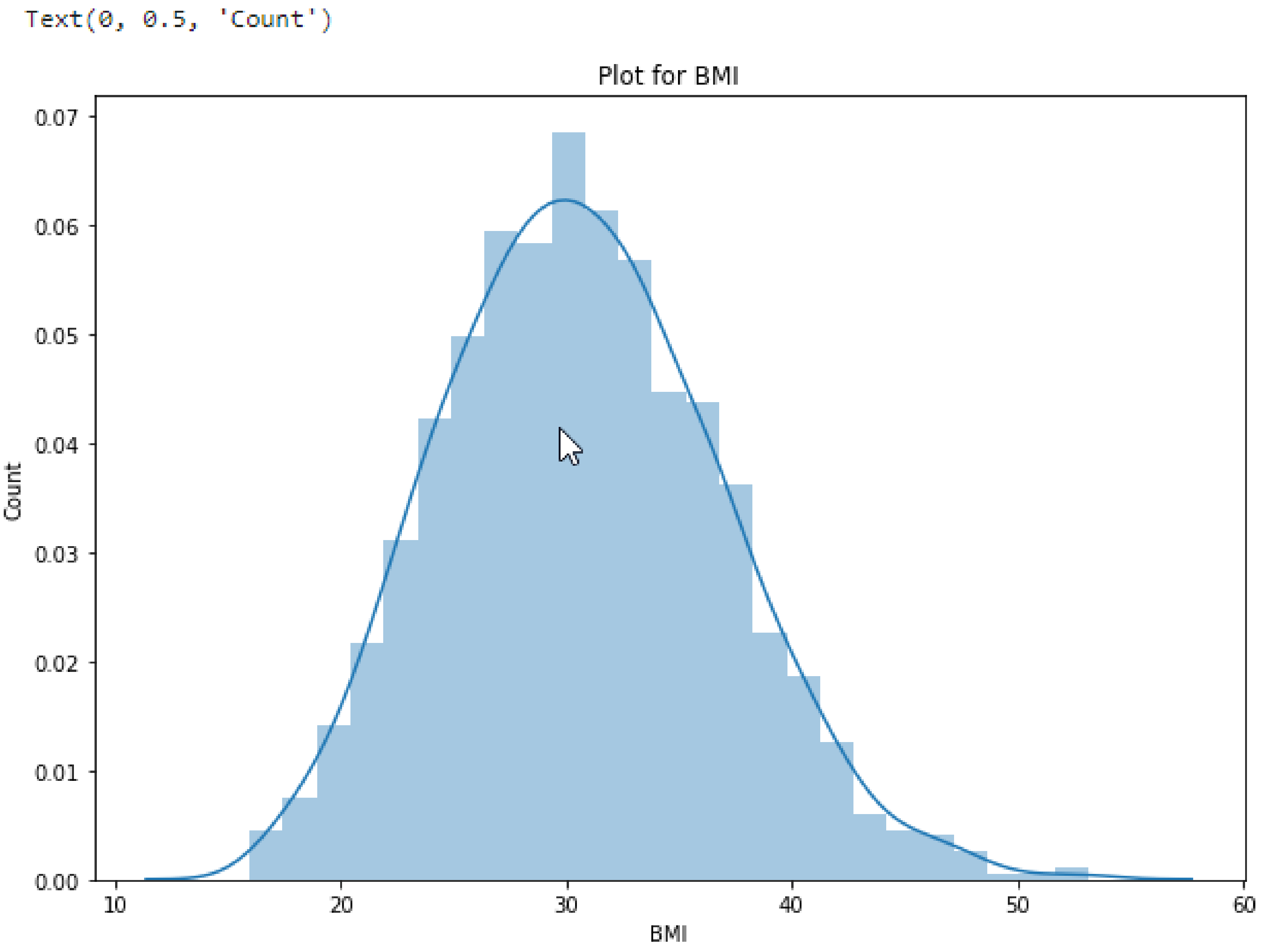
```
Printing Skewness and Kurtosis for all columns

age : Skewness 0.056 and  Kurtosis -1.245
sex : Skewness 0.021 and  Kurtosis -2.003
bmi : Skewness 0.284 and  Kurtosis -0.051
children : Skewness 0.938 and  Kurtosis 0.202
smoker : Skewness 1.465 and  Kurtosis 0.146
region : Skewness -0.038 and  Kurtosis -1.329
charges : Skewness 1.516 and  Kurtosis 1.606
```

The skew value of the age plot is 0.056, and the kurtosis value is −1.245 as shown in Figure below.



```
Text(0, 0.5, 'Count')
```

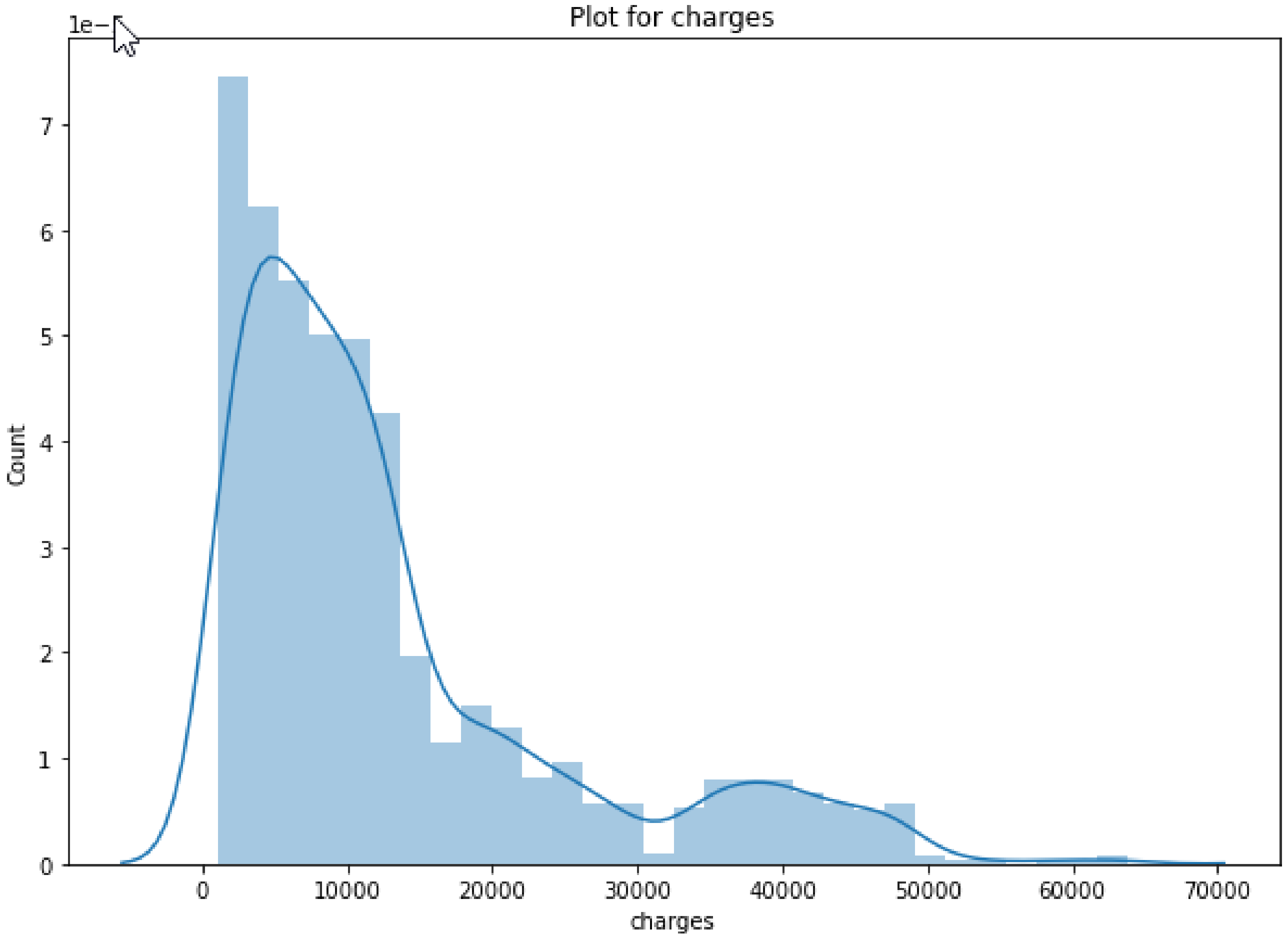
According to BMI, 0.284 and −0.051 are the skew and kurtosis values of BMI, respectively, as shown in Figure below.



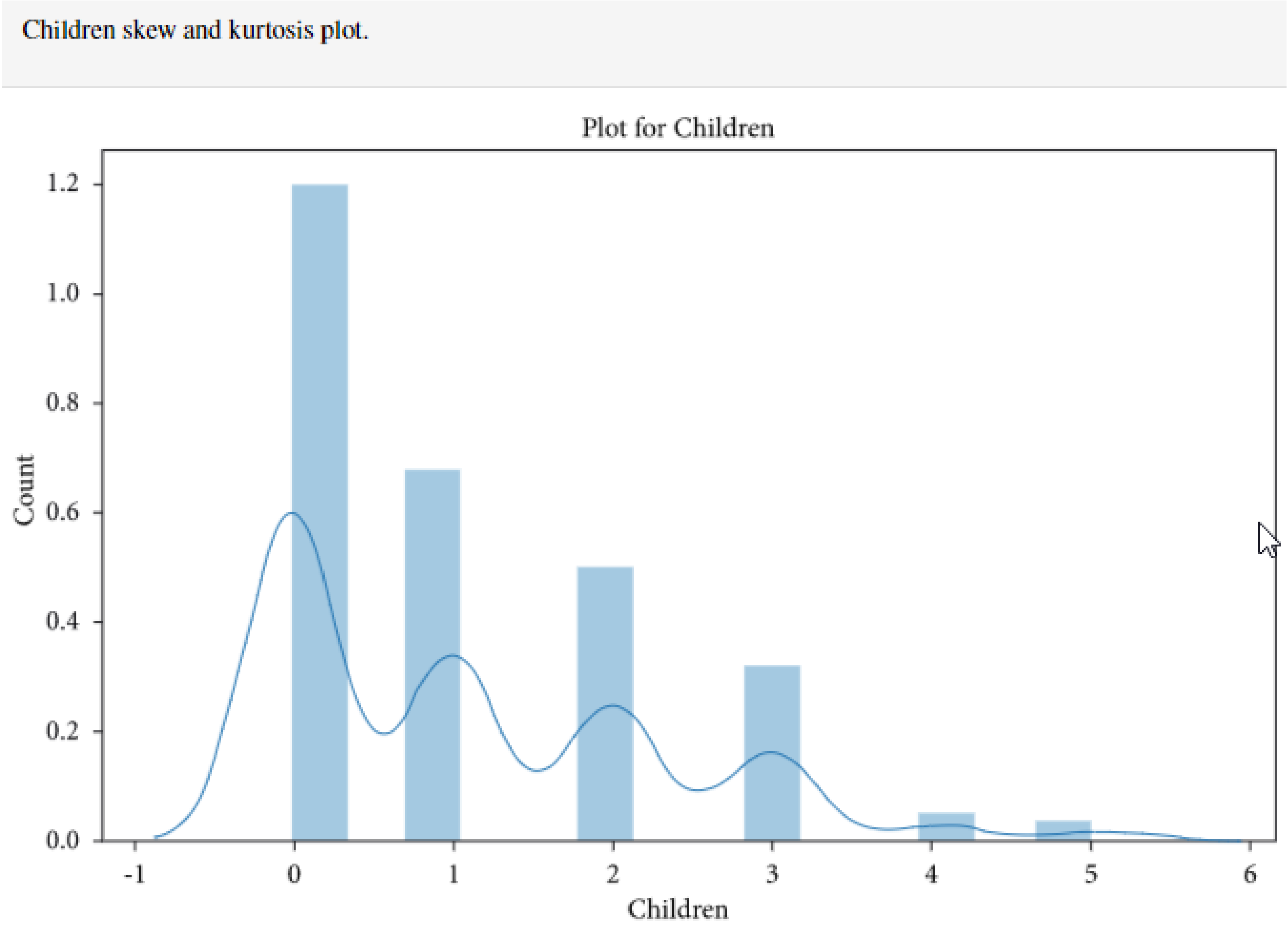
```
Text(0, 0.5, 'Count')
```

There might be few outliers in Charges but then we cannot say that the value is an outlier as there might be cases in which Charge for medical was very less actually!

```
Text(0, 0.5, 'Count')
```



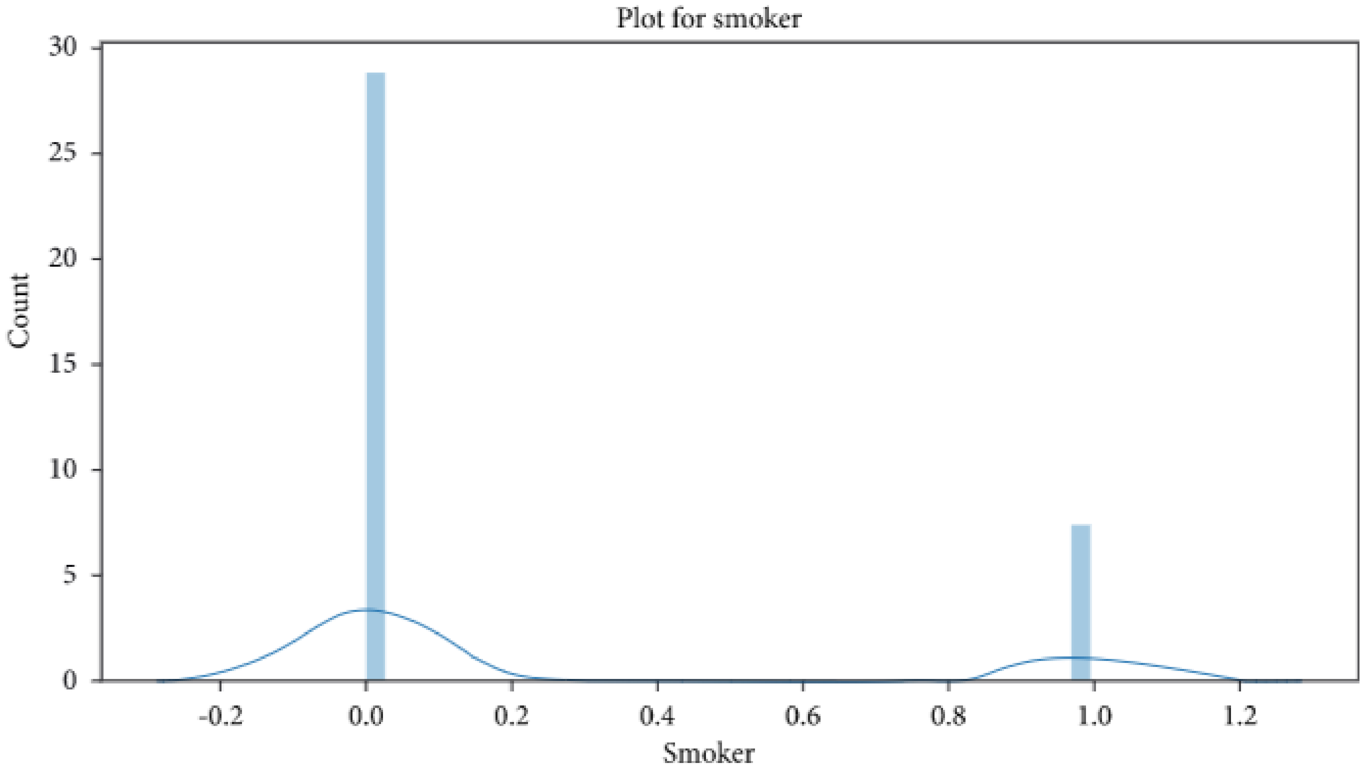
For children, 0.938 and 0.2020 are the skewness and kurtosis values of children, as shown in Figure below.



Children skew and kurtosis plot.

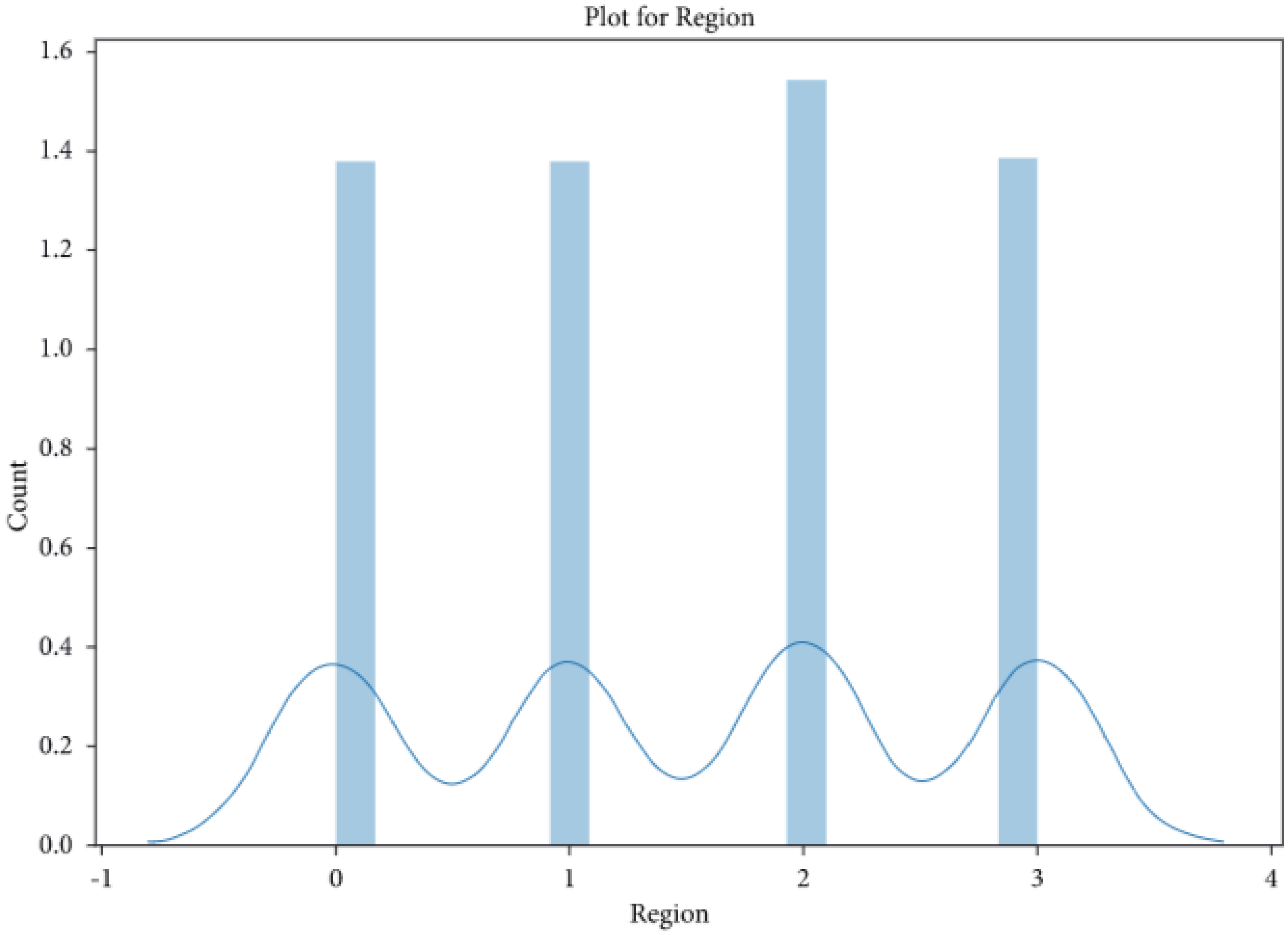
In case of smokers, 1.465 and 0.146 are the skewness and kurtosis values of smokers as shown in Figure below.

Smoker skew and kurtosis value plot.



Considering region, −0.038 and −1.329 are the skew and kurtosis values of region, respectively, as shown in Figure below.

Region skewness and kurtosis plot.



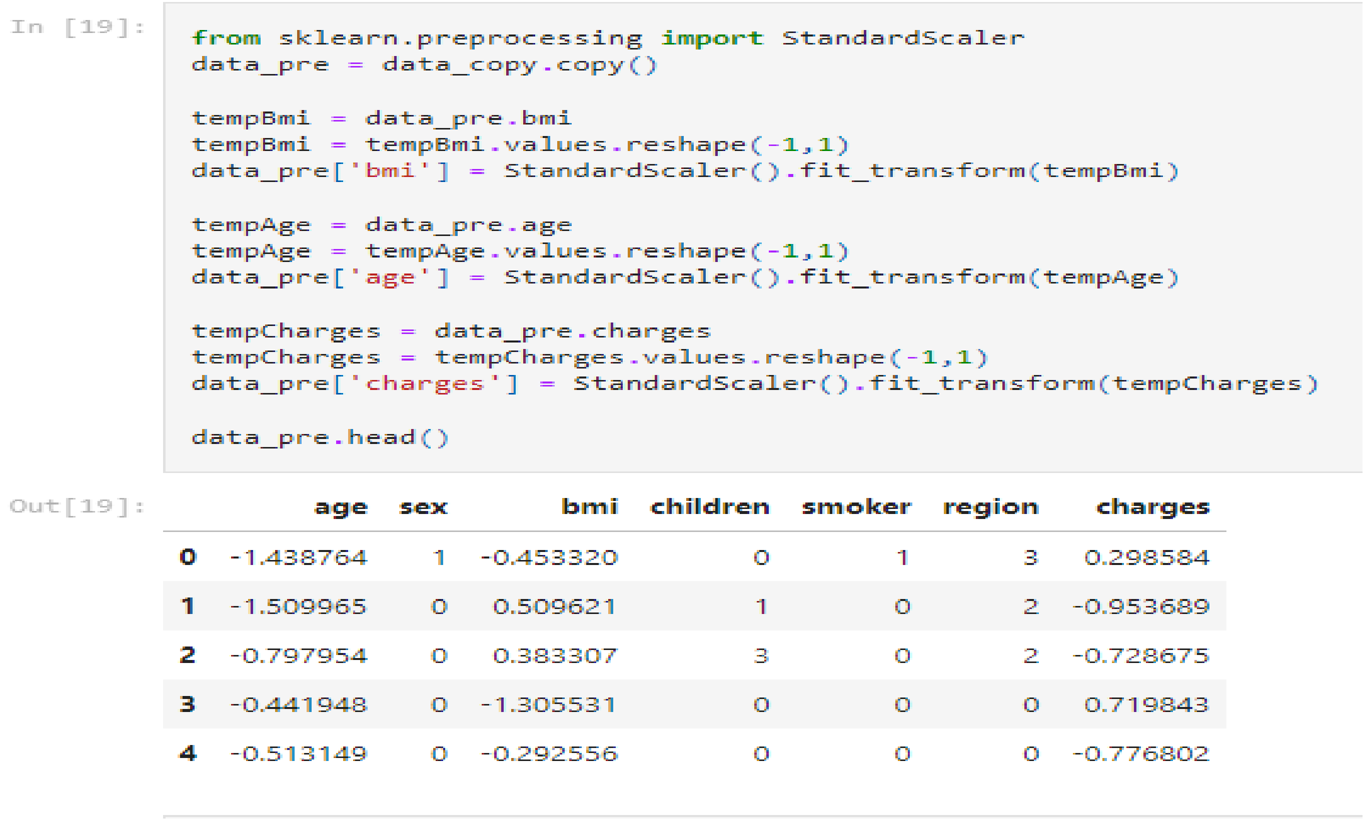
Preparing data - We can scale BMI and Charges Column before proceeding with Prediction

Step 1 clean: To get correct results, the information must be cleaned and missing values need to be filled.

Step 2 remodel: on this, we use smoothing, aggregation, and normalization tasks to make information more comprehensible through converting the layout of the data.

Step 3 Integration: before processing, we need to integrate the data because it is probably obtained from diverse sources, not always from a single one.

Step 4 reduction: The acquired data is very complicated and needs to be formatted for accomplishing preferred outcomes. The data is then classified and divided into test and training data which can be then attempted on different algorithms to get accuracy score results.



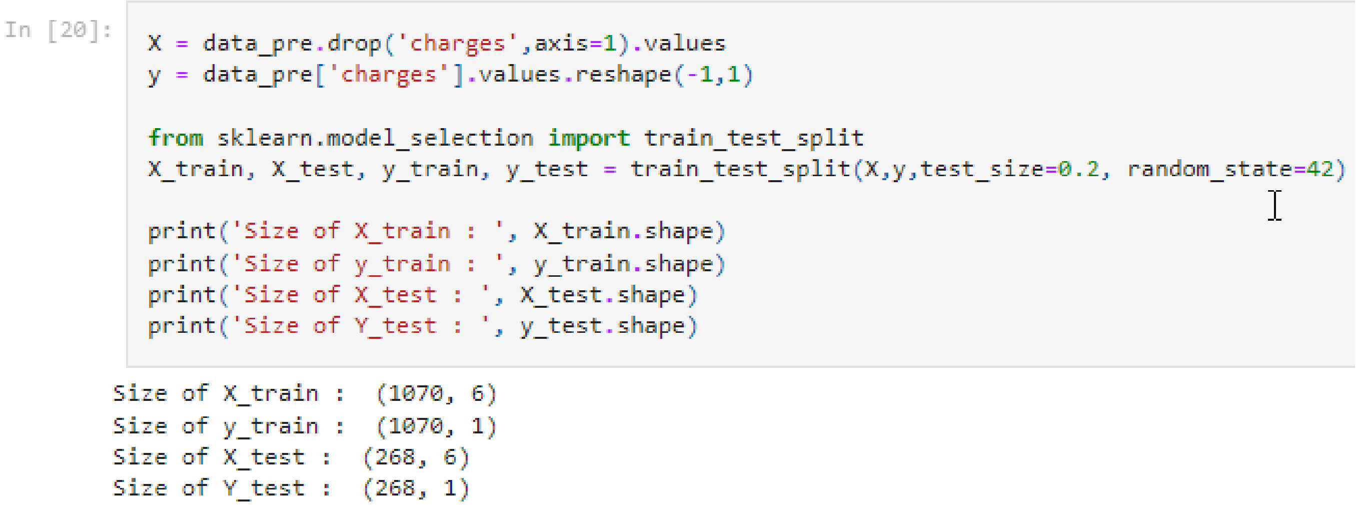
In [19]:

```
from sklearn.preprocessing import StandardScaler
data_pre = data_copy.copy()

tempBmi = data_pre.bmi
tempBmi = tempBmi.values.reshape(-1,1)
data_pre['bmi'] = StandardScaler().fit_transform(tempBmi)

tempAge = data_pre.age
tempAge = tempAge.values.reshape(-1,1)
data_pre['age'] = StandardScaler().fit_transform(tempAge)

tempCharges = data_pre.charges
tempCharges = tempCharges.values.reshape(-1,1)
data_pre['charges'] = StandardScaler().fit_transform(tempCharges)

data_pre.head()
```

Out[19]:

	age	sex	bmi	children	smoker	region	charges
0	-1.438764	1	-0.453320	0	1	3	0.298584
1	-1.509965	0	0.509621	1	0	2	-0.953689
2	-0.797954	0	0.383307	3	0	2	-0.728675
3	-0.441948	0	-1.305531	0	0	0	0.719843
4	-0.513149	0	-0.292556	0	0	0	-0.776802

Prepare data for train and test:

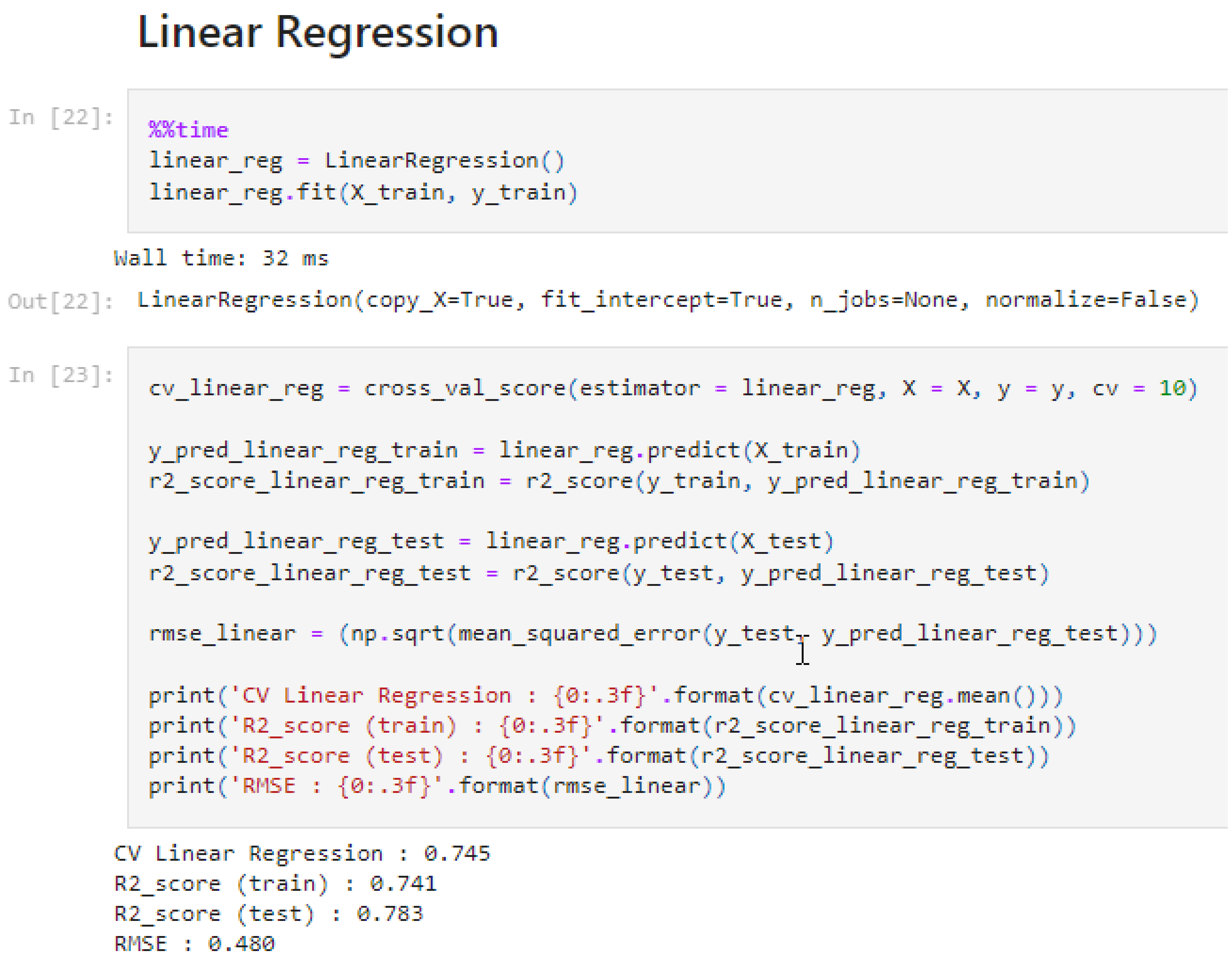


In [20]:

```
X = data_pre.drop('charges',axis=1).values
y = data_pre['charges'].values.reshape(-1,1)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=42)

print('Size of X_train : ', X_train.shape)
print('Size of y_train : ', y_train.shape)
print('Size of X_test : ', X_test.shape)
print('Size of Y_test : ', y_test.shape)
```

```
Size of X_train :  (1070, 6)
Size of y_train :  (1070, 1)
Size of X_test :  (268, 6)
Size of Y_test :  (268, 1)
```

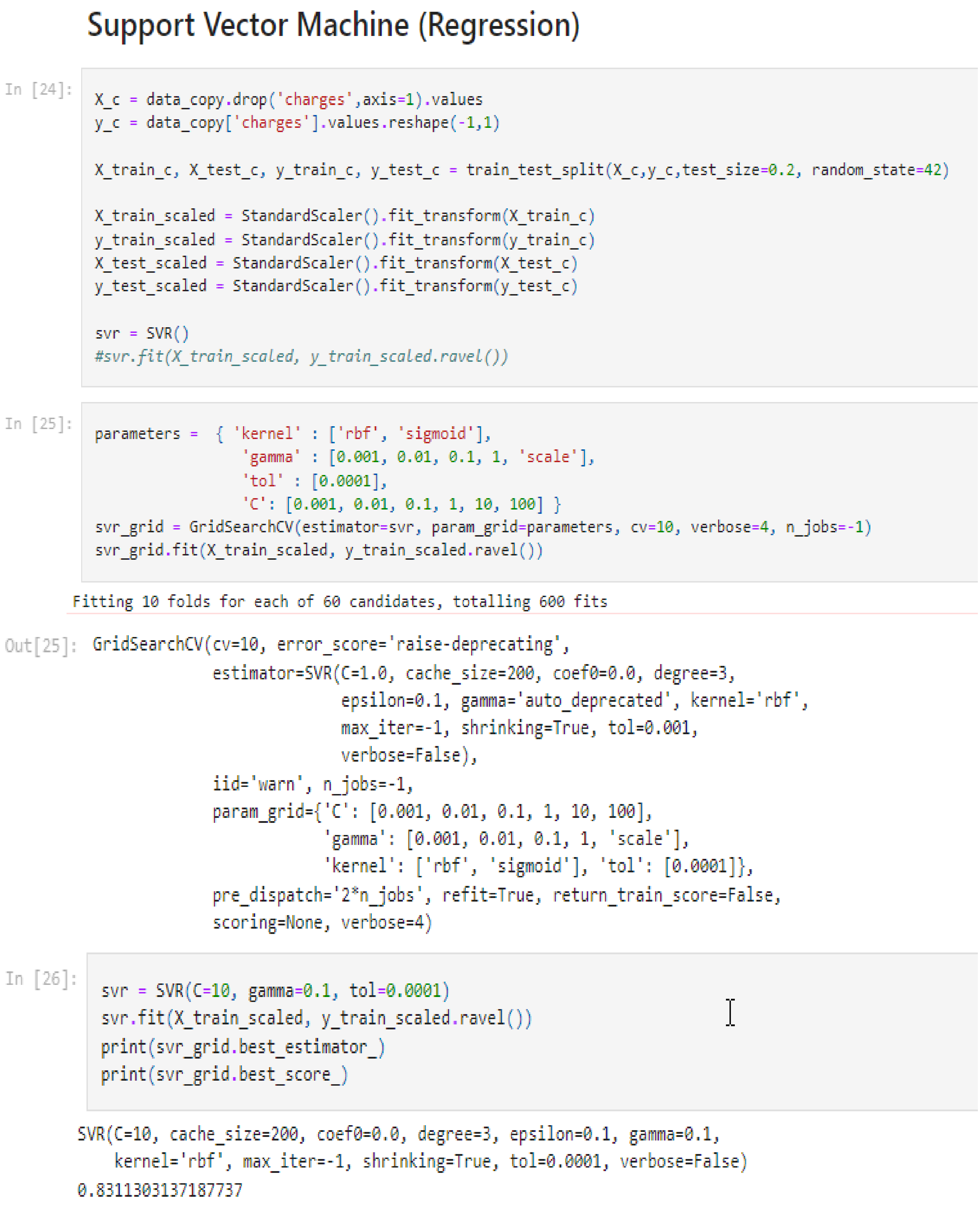
4. IMPLEMENTATION



Linear Regression

```
In [22]:  %%time
          linear_reg = LinearRegression()
          linear_reg.fit(X_train, y_train)

          Wall time: 32 ms

Out[22]:  LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
In [23]:  cv_linear_reg = cross_val_score(estimator = linear_reg, X = X, y = y, cv = 10)

          y_pred_linear_reg_train = linear_reg.predict(X_train)
          r2_score_linear_reg_train = r2_score(y_train, y_pred_linear_reg_train)

          y_pred_linear_reg_test = linear_reg.predict(X_test)
          r2_score_linear_reg_test = r2_score(y_test, y_pred_linear_reg_test)

          rmse_linear = (np.sqrt(mean_squared_error(y_test, y_pred_linear_reg_test)))

          print('CV Linear Regression : {0:.3f}'.format(cv_linear_reg.mean()))
          print('R2_score (train) : {0:.3f}'.format(r2_score_linear_reg_train))
          print('R2_score (test) : {0:.3f}'.format(r2_score_linear_reg_test))
          print('RMSE : {0:.3f}'.format(rmse_linear))

          CV Linear Regression : 0.745
          R2_score (train) : 0.741
          R2_score (test) : 0.783
          RMSE : 0.480
```



Support Vector Machine (Regression)

In [24]:

```
X_c = data_copy.drop('charges',axis=1).values
y_c = data_copy['charges'].values.reshape(-1,1)

X_train_c, X_test_c, y_train_c, y_test_c = train_test_split(X_c,y_c,test_size=0.2, random_state=42)

X_train_scaled = StandardScaler().fit_transform(X_train_c)
y_train_scaled = StandardScaler().fit_transform(y_train_c)
X_test_scaled = StandardScaler().fit_transform(X_test_c)
y_test_scaled = StandardScaler().fit_transform(y_test_c)

svr = SVR()
#svr.fit(X_train_scaled, y_train_scaled.ravel())
```

In [25]:

```
parameters =  { 'kernel' : ['rbf', 'sigmoid'],
                 'gamma' : [0.001, 0.01, 0.1, 1, 'scale'],
                 'tol' : [0.0001],
                 'C': [0.001, 0.01, 0.1, 1, 10, 100] }
svr_grid = GridSearchCV(estimator=svr, param_grid=parameters, cv=10, verbose=4, n_jobs=-1)
svr_grid.fit(X_train_scaled, y_train_scaled.ravel())
```

```
Fitting 10 folds for each of 60 candidates, totalling 600 fits
```

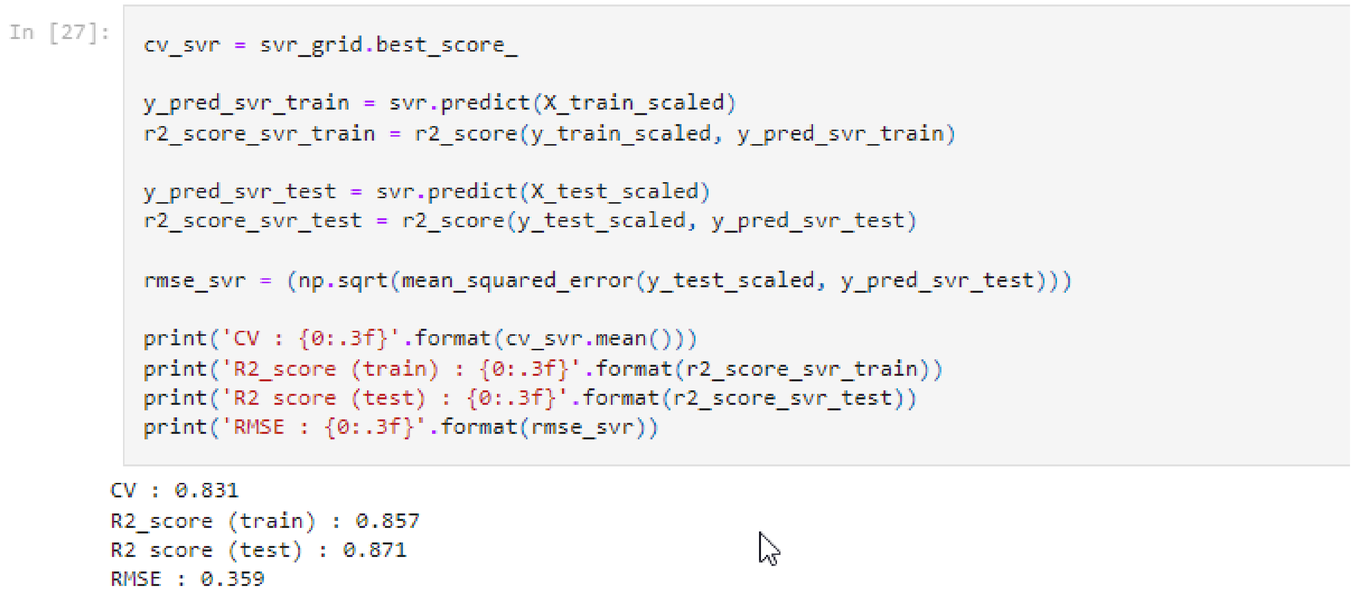
Out[25]:

```
GridSearchCV(cv=10, error_score='raise-deprecating',
             estimator=SVR(C=1.0, cache_size=200, coef0=0.0, degree=3,
                           epsilon=0.1, gamma='auto_deprecated', kernel='rbf',
                           max_iter=-1, shrinking=True, tol=0.001,
                           verbose=False),
             iid='warn', n_jobs=-1,
             param_grid={'C': [0.001, 0.01, 0.1, 1, 10, 100],
                         'gamma': [0.001, 0.01, 0.1, 1, 'scale'],
                         'kernel': ['rbf', 'sigmoid'], 'tol': [0.0001]},
             pre_dispatch='2*n_jobs', refit=True, return_train_score=False,
             scoring=None, verbose=4)
```

In [26]:

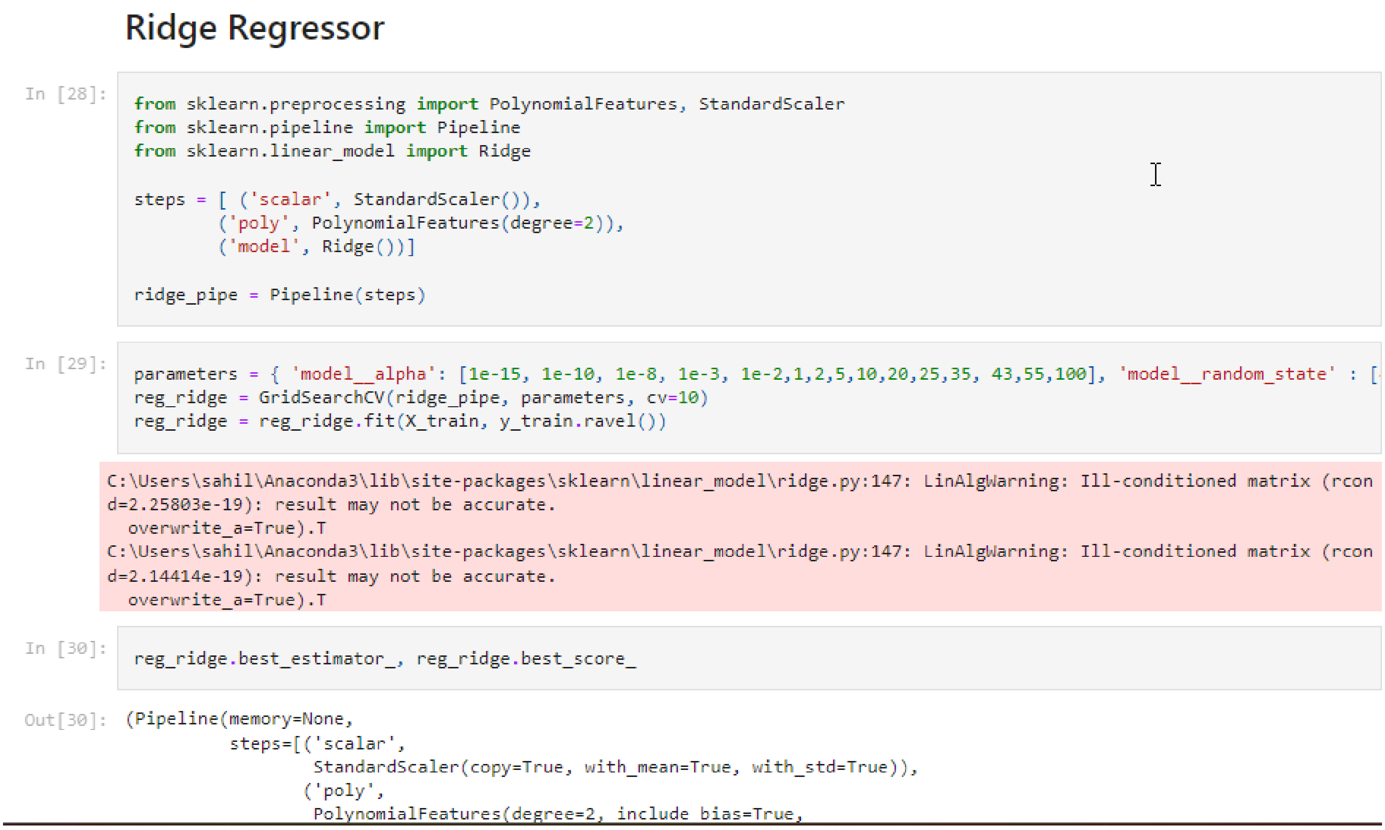
```
svr = SVR(C=10, gamma=0.1, tol=0.0001)
svr.fit(X_train_scaled, y_train_scaled.ravel())
print(svr_grid.best_estimator_)
print(svr_grid.best_score_)
```

```
SVR(C=10, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma=0.1,
    kernel='rbf', max_iter=-1, shrinking=True, tol=0.0001, verbose=False)
0.8311303137187737
```



```
In [27]:  cv_svr = svr_grid.best_score_

          y_pred_svr_train = svr.predict(X_train_scaled)
          r2_score_svr_train = r2_score(y_train_scaled, y_pred_svr_train)

          y_pred_svr_test = svr.predict(X_test_scaled)
          r2_score_svr_test = r2_score(y_test_scaled, y_pred_svr_test)

          rmse_svr = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_svr_test)))

          print('CV : {0:.3f}'.format(cv_svr.mean()))
          print('R2_score (train) : {0:.3f}'.format(r2_score_svr_train))
          print('R2 score (test) : {0:.3f}'.format(r2_score_svr_test))
          print('RMSE : {0:.3f}'.format(rmse_svr))
```

```
CV : 0.831
R2_score (train) : 0.857
R2 score (test) : 0.871
RMSE : 0.359
```



Ridge Regressor

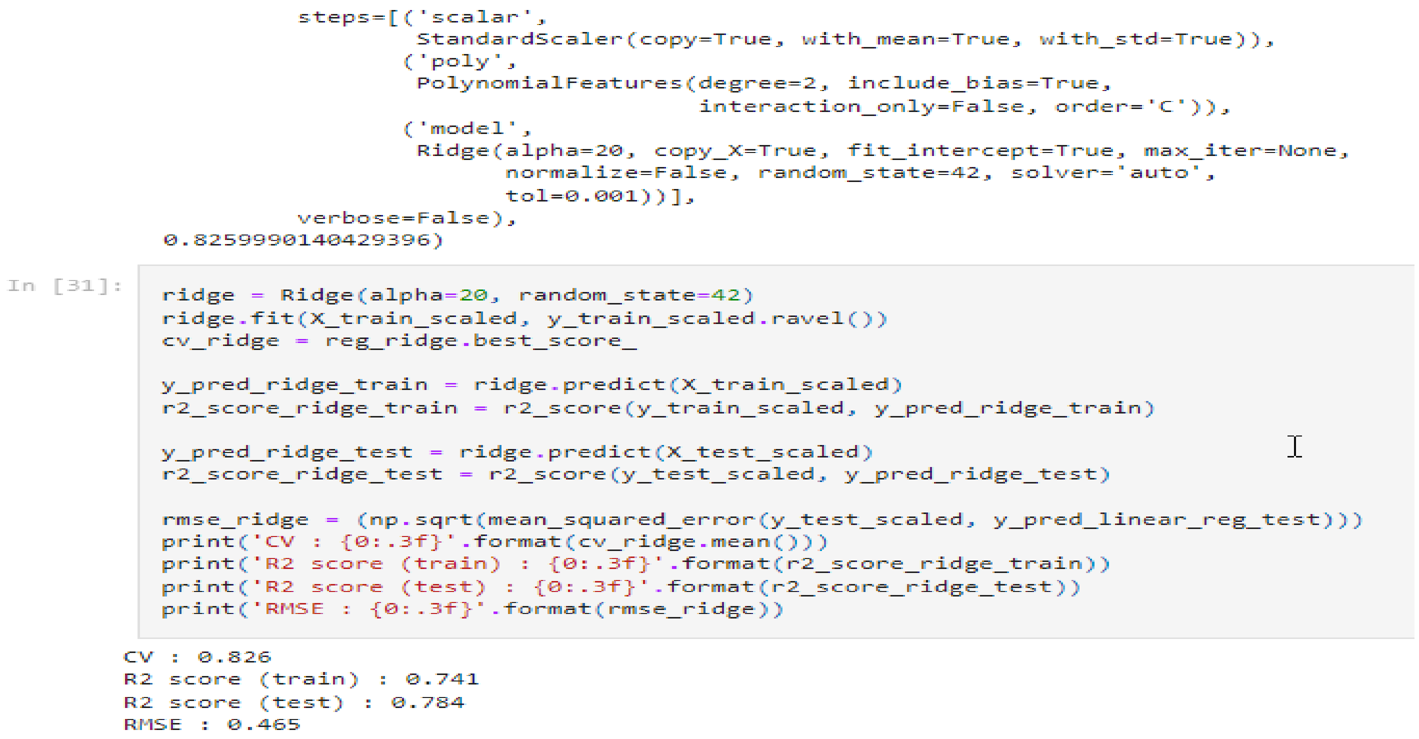
```
In [28]:  from sklearn.preprocessing import PolynomialFeatures, StandardScaler
          from sklearn.pipeline import Pipeline
          from sklearn.linear_model import Ridge

          steps = [ ('scalar', StandardScaler()),
                  ('poly', PolynomialFeatures(degree=2)),
                  ('model', Ridge())]

          ridge_pipe = Pipeline(steps)
```

```
In [29]:  parameters = { 'model__alpha': [1e-15, 1e-10, 1e-8, 1e-3, 1e-2,1,2,5,10,20,25,35, 43,55,100], 'model__random_state' : [
          reg_ridge = GridSearchCV(ridge_pipe, parameters, cv=10)
          reg_ridge = reg_ridge.fit(X_train, y_train.ravel())
```

```
C:\Users\sahil\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-conditioned matrix (rcon
d=2.25803e-19): result may not be accurate.
  overwrite_a=True).T
C:\Users\sahil\Anaconda3\lib\site-packages\sklearn\linear_model\ridge.py:147: LinAlgWarning: Ill-conditioned matrix (rcon
d=2.14414e-19): result may not be accurate.
  overwrite_a=True).T
```

```
In [30]:  reg_ridge.best_estimator_, reg_ridge.best_score_
```

```
Out[30]: (Pipeline(memory=None,
                   steps=[('scalar',
                           StandardScaler(copy=True, with_mean=True, with_std=True)),
                         ('poly',
                           PolynomialFeatures(degree=2, include_bias=True,
```



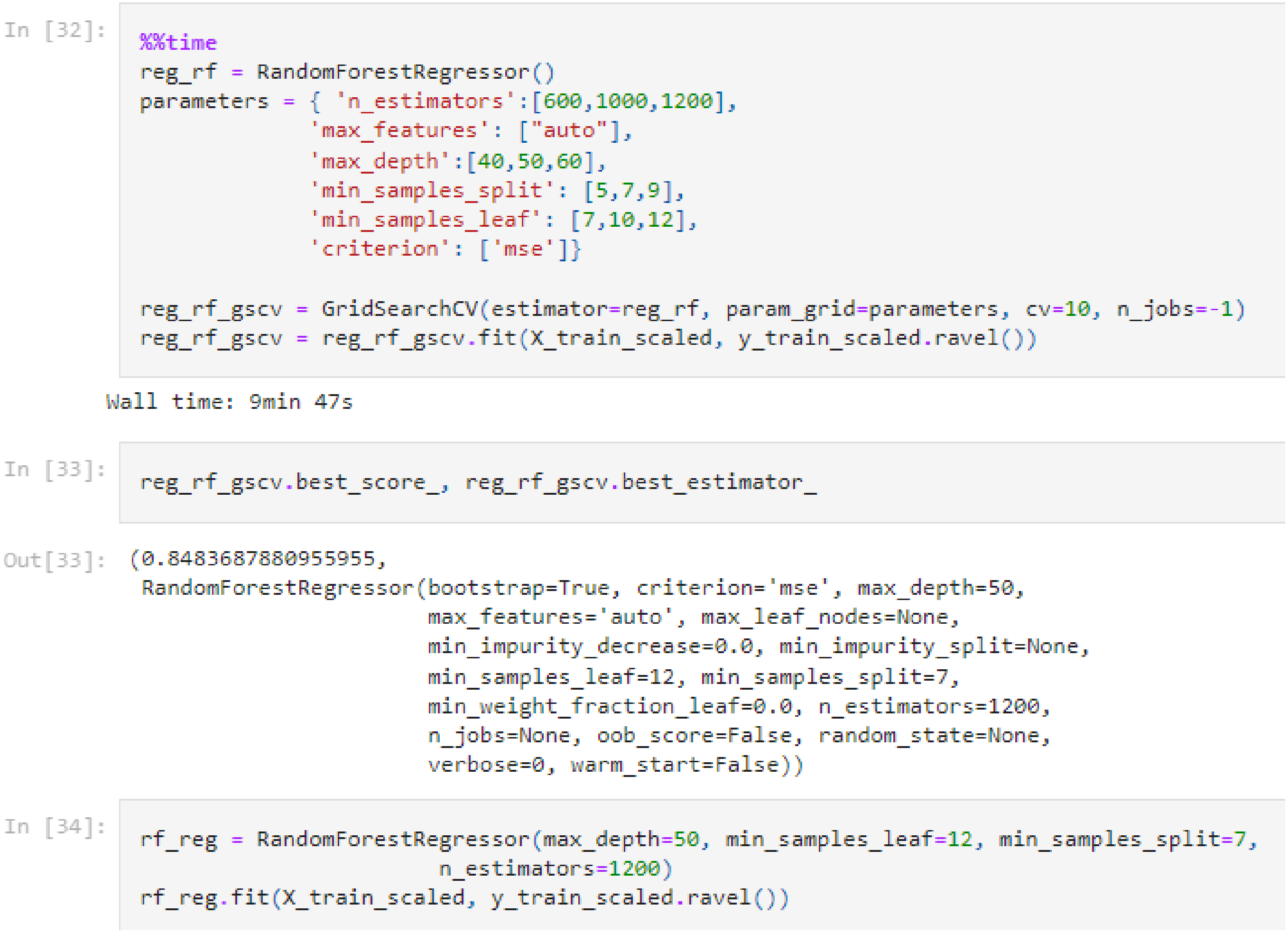
```
              steps=[('scalar',
                      StandardScaler(copy=True, with_mean=True, with_std=True)),
                     ('poly',
                      PolynomialFeatures(degree=2, include_bias=True,
                                         interaction_only=False, order='C')),
                     ('model',
                      Ridge(alpha=20, copy_X=True, fit_intercept=True, max_iter=None,
                            normalize=False, random_state=42, solver='auto',
                            tol=0.001))],
              verbose=False),
 0.8259990140429396)
```

In [31]:

```
ridge = Ridge(alpha=20, random_state=42)
ridge.fit(X_train_scaled, y_train_scaled.ravel())
cv_ridge = reg_ridge.best_score_

y_pred_ridge_train = ridge.predict(X_train_scaled)
r2_score_ridge_train = r2_score(y_train_scaled, y_pred_ridge_train)

y_pred_ridge_test = ridge.predict(X_test_scaled)
r2_score_ridge_test = r2_score(y_test_scaled, y_pred_ridge_test)

rmse_ridge = (np.sqrt(mean_squared_error(y_test_scaled, y_pred_linear_reg_test)))
print('CV : {0:.3f}'.format(cv_ridge.mean()))
print('R2 score (train) : {0:.3f}'.format(r2_score_ridge_train))
print('R2 score (test) : {0:.3f}'.format(r2_score_ridge_test))
print('RMSE : {0:.3f}'.format(rmse_ridge))
```

```
CV : 0.826
R2 score (train) : 0.741
R2 score (test) : 0.784
RMSE : 0.465
```

RandomForest Regressor



In [32]:

```
%%time
reg_rf = RandomForestRegressor()
parameters = { 'n_estimators':[600,1000,1200],
             'max_features': ["auto"],
             'max_depth':[40,50,60],
             'min_samples_split': [5,7,9],
             'min_samples_leaf': [7,10,12],
             'criterion': ['mse']}

reg_rf_gscv = GridSearchCV(estimator=reg_rf, param_grid=parameters, cv=10, n_jobs=-1)
reg_rf_gscv = reg_rf_gscv.fit(X_train_scaled, y_train_scaled.ravel())
```

```
Wall time: 9min 47s
```

In [33]:

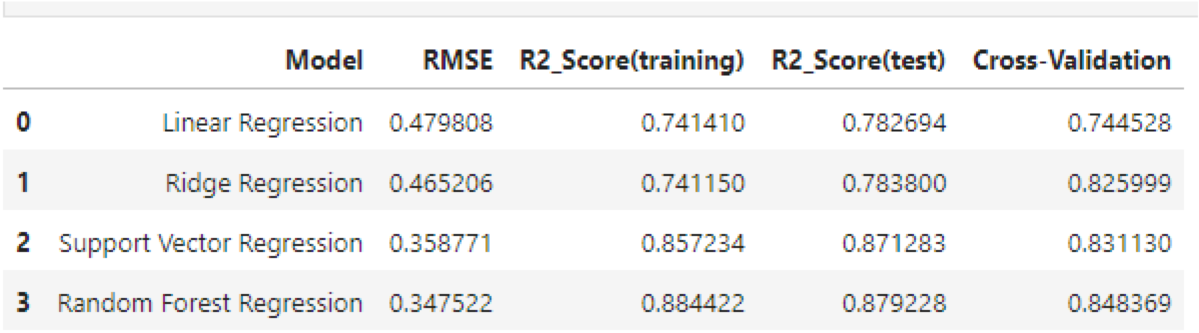
```
reg_rf_gscv.best_score_, reg_rf_gscv.best_estimator_
```

Out[33]:

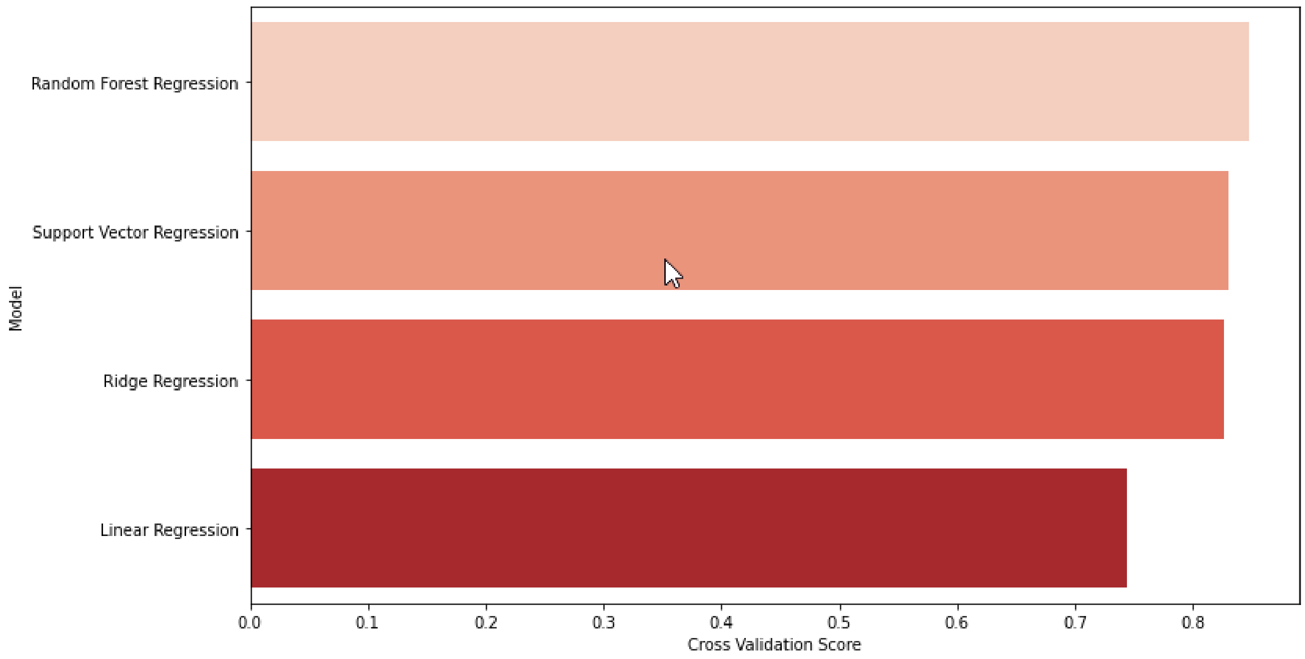
```
(0.8483687880955955,
 RandomForestRegressor(bootstrap=True, criterion='mse', max_depth=50,
                       max_features='auto', max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=12, min_samples_split=7,
                       min_weight_fraction_leaf=0.0, n_estimators=1200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False))
```

In [34]:

```
rf_reg = RandomForestRegressor(max_depth=50, min_samples_leaf=12, min_samples_split=7,
                               n_estimators=1200)
rf_reg.fit(X_train_scaled, y_train_scaled.ravel())
```

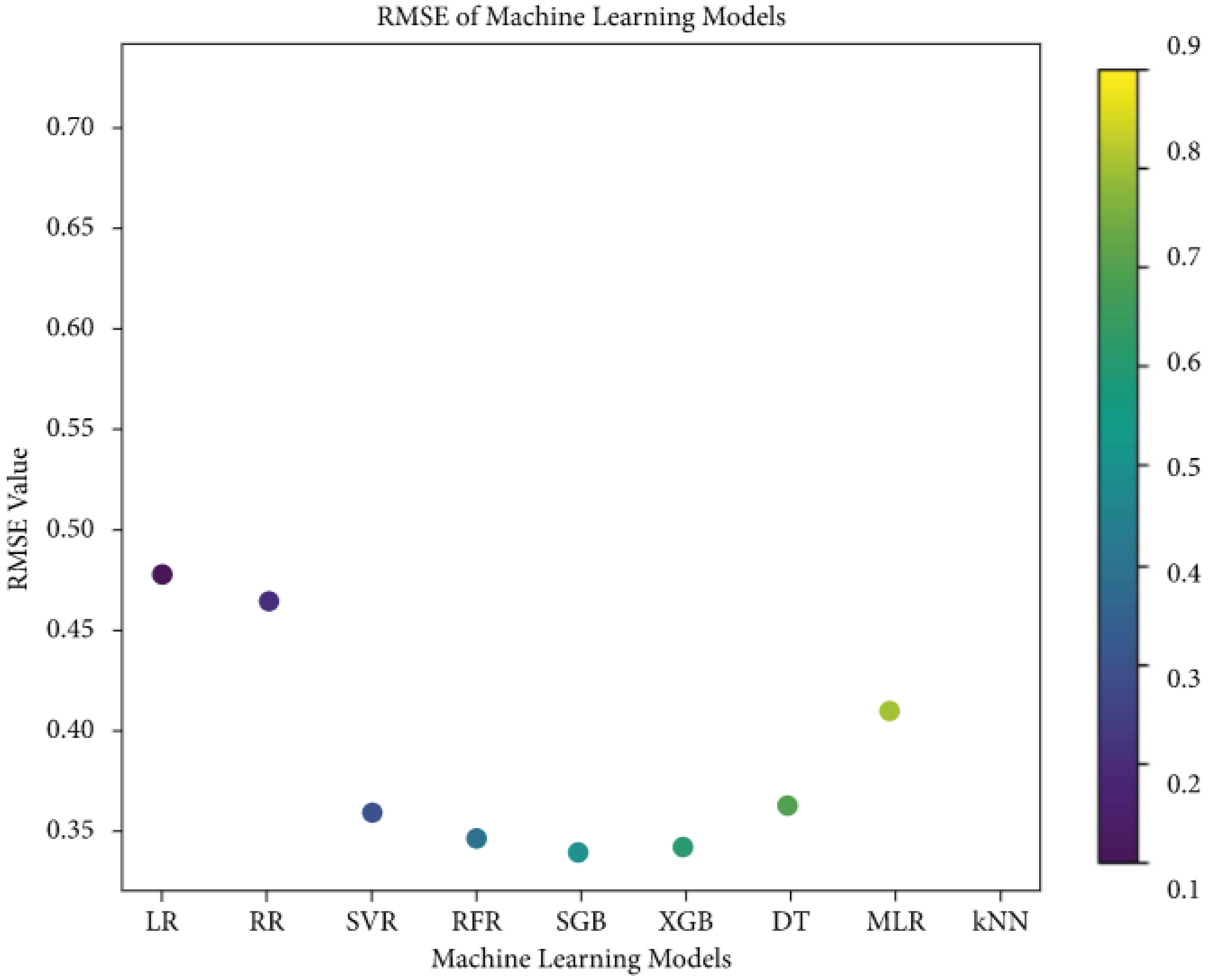


	Model	RMSE	R2_Score(training)	R2_Score(test)	Cross-Validation
0	Linear Regression	0.479808	0.741410	0.782694	0.744528
1	Ridge Regression	0.465206	0.741150	0.783800	0.825999
2	Support Vector Regression	0.358771	0.857234	0.871283	0.831130
3	Random Forest Regression	0.347522	0.884422	0.879228	0.848369



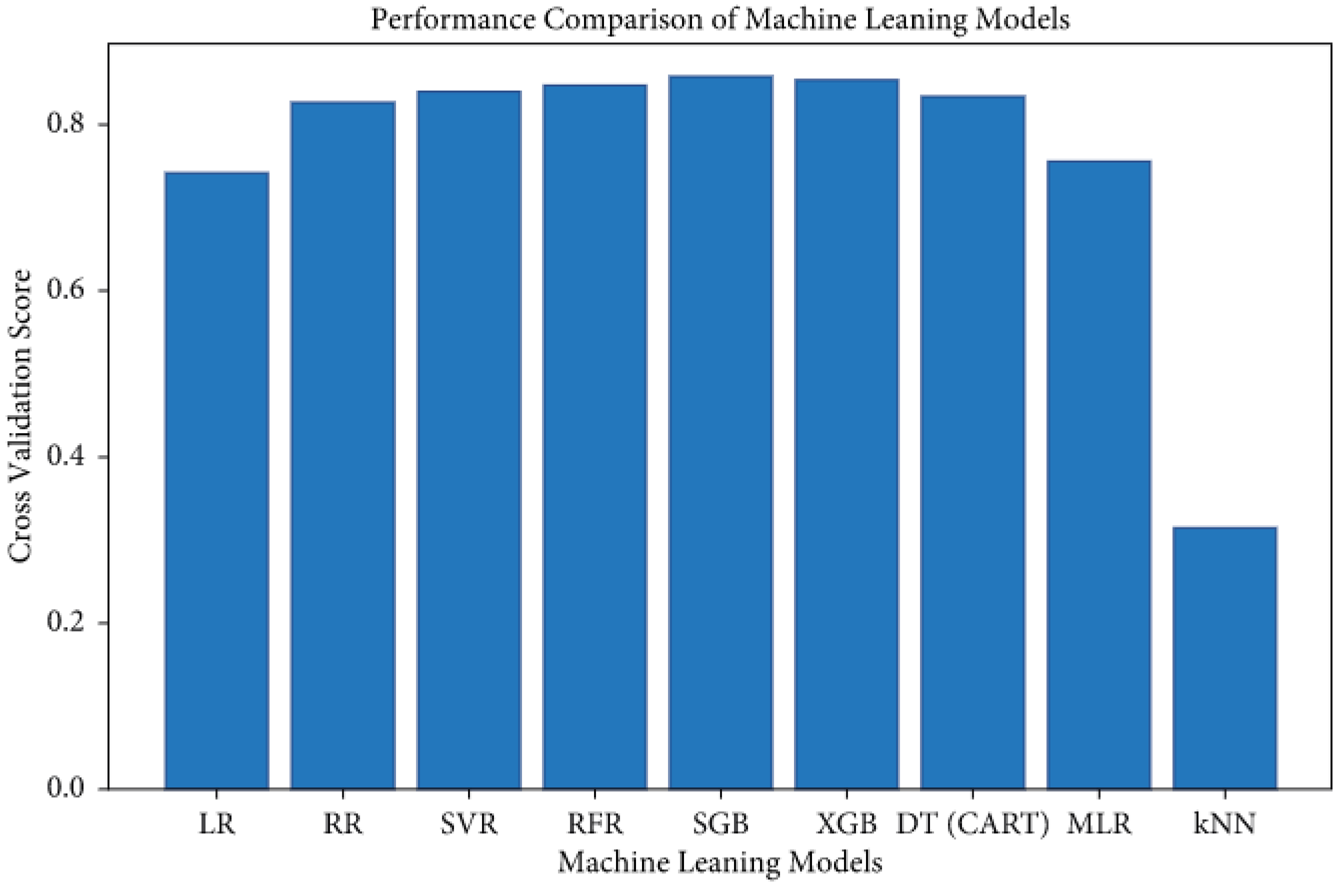
Performance of ML Algorithms

The performance of all the algorithms in terms of RMSE (root mean squared error), training and test scores, and cross-validations is shown in Table 5. In Figure 15, the RMSE value of all machine learning (ML) algorithms is visualized for better understanding. By comparing the RMSE value of these ML models, in comparison to the other ML models, k-Nearest Neighbors provides a high RMSE value of 0.726835.



By comparing the performance of all these machine learning algorithms, we conclude that Stochastic Gradient Boosting, XGBoost, and Random Forest Regression performed better as compared to the other ML algorithms and these models achieved almost 86%, 85%, and 85% accuracy, respectively, as shown in Figure

Graph for all models to compare their performance.



The objective of this study was to predict the price of medical insurance. We used various regression techniques like Linear Regression, Ridge Regression, Lasso Regression, Random forest, and Elastic Net. For this experiment, we recommend that one should use a machine with at least 16GB RAM and generation of Intel Processor shall be at-least 9th technology or above. Dataset was split into 2 sets, one set for training and one for testing purposes. We used python programming for noting the results of carried-out techniques on test and training datasets.

5. RESULTS

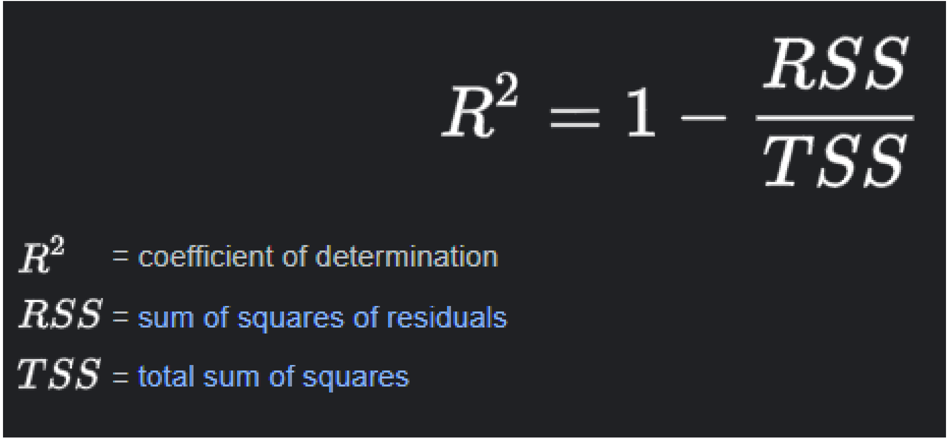
Machine learning (ML) is one aspect of computational intelligence that can solve different problems in a wide range of applications and systems when it comes to leveraging historical data. Predicting medical insurance costs is still a problem in the healthcare industry that needs to be investigated and improved. In this paper, by using a set of ML algorithms, a computational intelligence approach is applied to predict healthcare insurance costs. The medical insurance dataset was obtained from the github repository and was utilised for training and testing the Linear Regression, Ridge Regressor, Support Vector Regression, XGBoost, Stochastic Gradient Boosting, Decision Tree, Random Forest Regressor, k-Nearest Neighbors, and Multiple Linear Regression ML algorithms. The regression of this dataset followed the steps of preprocessing, feature engineering, data splitting, regression, and evaluation. The resultant outcome revealed that Stochastic Gradient Boosting (SGB) achieved a high accuracy of 86% with an RMSE of 0.340.

The regression model's performance is evaluated on the basis of the following metrics

- R2_Score
- Root Mean Square Error (RMSE)

R2_Score:

R-Squared is a good measure to evaluate the model fitness. The R-squared value lies between 0 to 1 (0% to 100%). Large value represents a better fit.



$$R^2 = 1 - \frac{RSS}{TSS}$$

R^2 = coefficient of determination

RSS = sum of squares of residuals

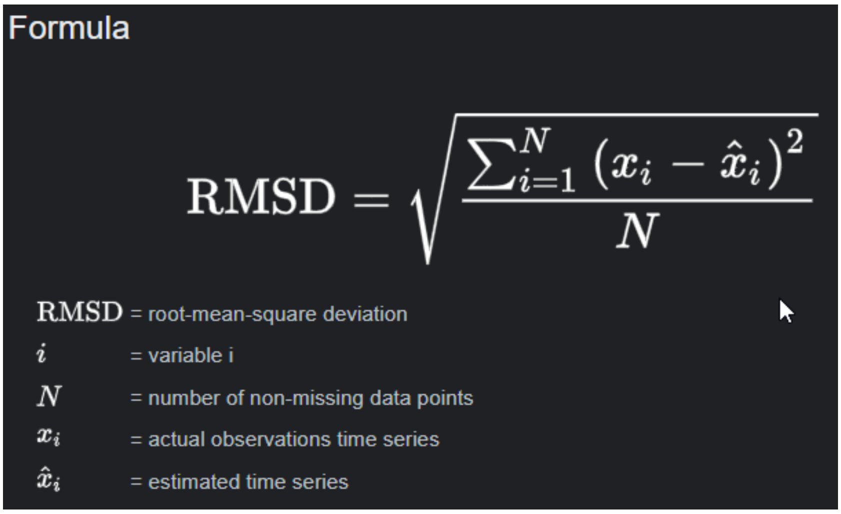
TSS = total sum of squares

(7)

where SSE (Squared sum of error): sum of the squared residuals, which is squared differences of each observation from the predicted value. and, SST (Sum of Squared Total): squared differences of each observation from the overall mean., where is the average of the observed values.

RMSE:

The Root Mean Square error is a common method of calculating a model’s prediction error which represents how close the observed data points are to the model's predicted values, shows the model's absolute fit to the data points. A better match is indicated by lower RMSE values



Formula

$$\text{RMSD} = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}}$$

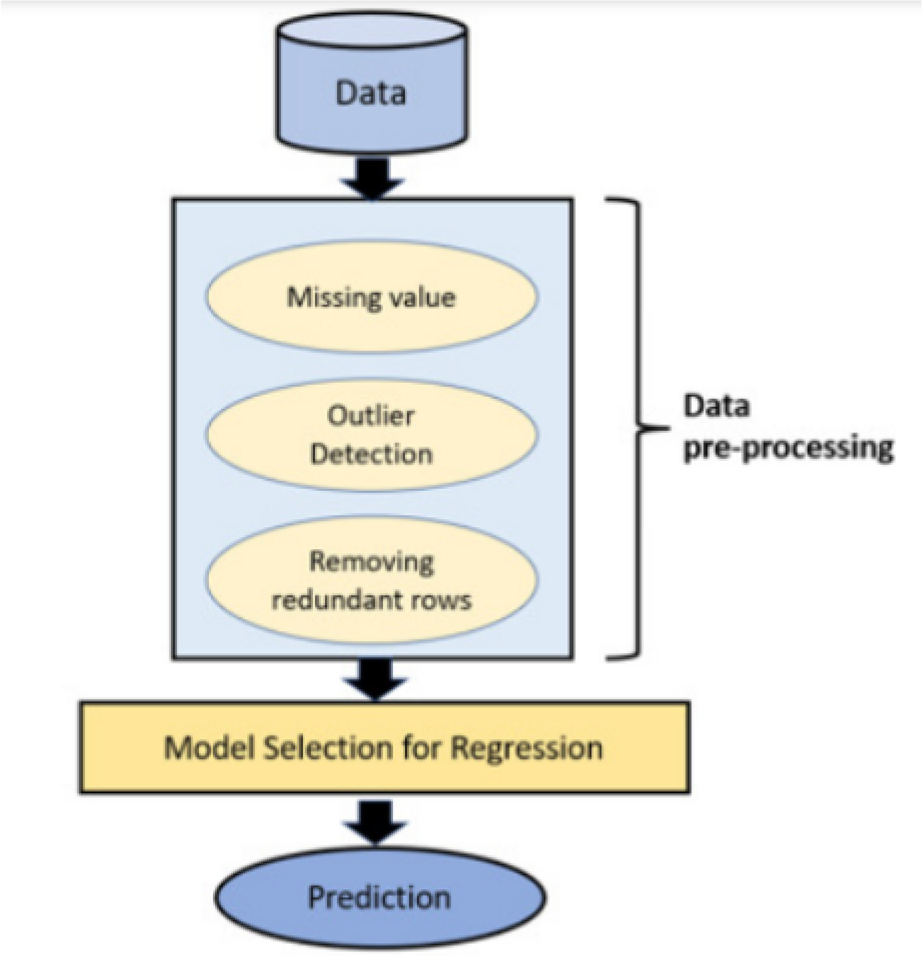
RMSD = root-mean-square deviation
i = variable i
N = number of non-missing data points
x_i = actual observations time series
$\hat{x}_i$ = estimated time series

The outcomes of the above discussed models after testing the models on the test dataset were noted. Table.2 displays t RMSE and Accuracy metrics of the models discussed so far.

OBSERVATIONS

From the preceding calculations, it can be seen that Polynomial Regression outperforms other models for the proposed MLHIPS giving an accuracy of 80.97%. The Polynomial Regression in contrast to other models fits a curve to the dataset which increases the variance of the model thereby reducing the residual error.
With an RMSE of 5100.53 and R2 value of 0.80, the model achieves superior results when using Polynomial Regression . If there exists a non linear relationship between the target and set of predictor variables then polynomial regression eventually produces better results. Besides, Polynomial Regression Ridge and Lasso Reg ression of MLHIPS have achieved an accuracy value of 75.82% and 75.86%, respectively. The 6 independent parameters have a considerable correlation amongst them as a result of which lasso regression and ridge regression produces similar results. The Multiple linear regression model of MLHIPS has achieved an accuracy of 75.86%. whereas the simple linear regression had an accuracy of 62.86% which was the lowest among all. In the above scenario the dataset is partitioned

into 80-20 ratio for training and testi In the dataset of 70:30 ratio, the accuracy of the polynomial



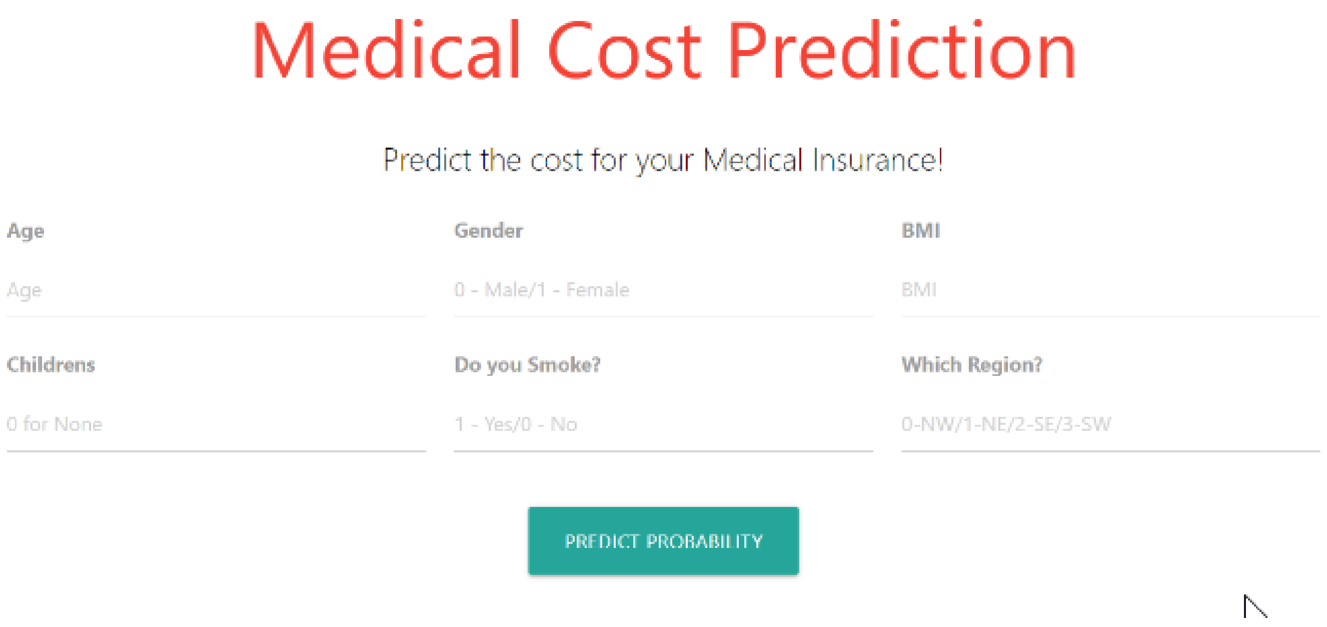
Flowchart of MLHIPS Model

regression decreases from previous value of 80.97% to 80.54% which is negligible, and similarly the rest of the models have also produced a drop in the accuracy values. The multiple linear regression, ridge regression and lasso regression accuracy values are reduced by approximately 2 units from their past values of 75.86 % to 73.32% for multiple linear regression, 75.82 % to 73.56% for ridge from 75.86% to 74.12% for lasso regression Further it was noted that on increasing the degree of the polynomial regression from n=2 to n=3 the accuracy has increased from previous value of 80.97% to 83.62%. But later on, further increasing the degree to 4 and higher values there was a drop in accuracy from 83.62% to 68.06% for degree n=4 and 51.98% for degree n=5. Thus, with degree n=3 the polynomial regression gives us a good accuracy in predicting the charges. A working of the MLHIPS model

CONCLUSION : Model gave 86% accuracy for Medical Insurance Amount Prediction using Random Forest Regressor.

6. FUTURE WORK

In this project we discussed some of the traditional regression models for our proposed problem statement, moving forward some of the other techniques like Support Vector Machine (SVM), XGBoost, Decision Tree (CART), Random Forest Classifier and Stochastic Gradient Boosting needs to be addressed as the future work. Several optimization techniques such as the Genetic Algorithm or the Gradient Descent Algorithm may be applied on top of model evaluation. We can also apply some feature selection techniques to our dataset before we train our model to gain a good accuracy value as some of the features may be omitted while predicting the charges. Besides a model to perform well a good balanced dataset with a greater number of observations is required which will reduce the variability of the model so in the future if we get more data than the model can be trained well.will make an application using flask that will look like this:



Medical Cost Prediction

Predict the cost for your Medical Insurance!

7. BIBLIOGRAPHY

[1] “National Health Accounts,” National Health Systems Resource Centre. [Online].Available:https://nhsrcindia.org/national-health-accounts records
[2] “Global Expenditure on Health”, WHO annual report 2021, [Online].Available:https://www.who.int/newsroom/events/detail/2021/12/15/default-calendar/global-spending-on-health-2021
[3] “Health Insurance of India’s missing middle”, Niti Ayog India, Oct 2021, [Online]. Available: https://www.niti.gov.in
[4] J. L. Moran, P. J. Solomon, A. R. Peisach, and J. Martin, “New models for old questions: generalized linear models for cost prediction,” Journal of evaluation in clinical practice, vol. 13, no. 3, pp. 381–389, 2007.
[5] S. Sushmita, S. Newman, J. Marquardt, P. Ram, V. Prasad, M. D. Cock, A. Teredesai et al., “Population cost prediction on public healthcare datasets,” in Proceedings of the 5th International Conference on Digital Health 2015. ACM, 2015, pp. 87–94. Association for Computing Machinery, New York, NY, USA, 87–94.
[6] Lahiri B, Agarwal N. “Predicting healthcare expenditure increase for an individual from Medicare data”. Proceedings of the ACM SIGKDD Workshop on Health Informatics, 2014.
[7] Gregori, M. Petrinco, S. Bo, A. Desideri, F. Merletti, and E. Pagano, “Regression models for analyzing costs and their determinants in health care: an introductory review,” International Journal for Quality in Health Care, vol. 23, no. 3, pp. 331–341, 2011.
[8] Bertsimas, M. V. Bjarnad´ottir, M. A. Kane, J. C. Kryder, R. Pandey, S. Vempala, and G. Wang, “Algorithmic prediction of health-care costs,” Operations Research, vol. 56, no. 6, pp. 1382–1392, 2008.
[9] Stucki, O. “Predicting the customer churn with machine learning methods: case: private insurance customer data” Master’s dissertation, LUT University, Lappeenranta, Finland, 2019.
[10] Sterne, J. A., White, I. R., Carlin, J. B., Spratt, M., Royston, P., Kenward, Carpenter, J. R. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls. Bmj, 338L.
[11] H. Demirtas, “Flexible Imputation of Missing Data”, J. Stat. Soft., vol. 85, no. 4, pp. 1–5, Jul. 2018. Available: DOI: 10.18637/jss.v085.b04 .
[12] H. Goldstein, W. Browne and J. Rasbash, “Multilevel modelling of medical data,” Statistics in Medicine, John Wiley and Sons, vol. 21, no. 21, pp. 3291–3315, 2002.
[13] T. Han, A. Siddique, K. Khayat, J. Huang and A. Kumar, “An ensemble machine learning approach for prediction and optimization of modulus of elasticity of recycled aggregate concrete,” Construction and Building Materials, vol. 244, pp. 118–271, 2020.

[14] X. Zhu, C. Ying, J. Wang, J. Li, X. Lai et al., “Ensemble of ML-kNN for classification algorithm recommendation,” Knowledge-Based Systems, vol. 106, pp. 933, 2021.

[15] G. Reddy, S. Bhattacharya, S. Ramakrishnan, C. L. Chowdhary, S. Hakak et al., “An ensemble-based machine learning model for diabetic retinopathy classification,” in 2020 Int. Conf. on Emergig Trends in Information Technology and Engineering, IC-ETITE, VIT Vellore, IEEE, pp. 1–6, 2020.

[16] Douglas C Montgomery, Elizabeth A Peck and G Geoffrey Vining, “Introduction to linear regression analysis”, John Wiley & Sons, vol. 821, 2012.

[17] Tian Jinyu, Zhao Xin et al., “Apply multiple linear regression model to predict the audit opinion,” in 2009 ISECS International Colloquium on Computing, Communication, Control, and Management, IEEE, pp.1–6, 2009.

[18] Ostertagova et al., ”Modelling using Polynomial Regression”, “Procedia Engineering”, vol. 48, pp. 500-506, 2012.

[19] Donald W. Marquardt, Ronald D. Snee et al., ”Ridge Regression in Practice”, ”The American Statistician”, vol. 29, pp – 3-20, 2012.

[20] V. Roth, “The generalised LASSO”,” IEEE Transactions on Neural Networks”, vol. 15, pp – 16 28, 2004.

[21] Medical Cost Prediction Dataset, [Online].Available: https://www.kaggle.com/hely333/eda-regression/dat

[22] sahil chachra