HEALTH CARE CAPSTONE PROJECT

AMARLATA KUMARI (PGP-DSBA March'22)

DATE:-19th March 2023

TABLE OF CONTENTS

List of tables

List of Images

Health Care

1. INTRODUCTION

Health care sector in India

India’s healthcare ecosystem is among the fastest growing sectors having benefited from government policies to make the country a global hub for health and wellness. This sector is growing at a brisk pace due to its strengthening coverage, services and increasing expenditure by public as well as private players. India is cost competitive compared to its peers in Asia and Western countries. The cost of surgery in India is about one-tenth of that in the US or Western Europe. The hospital industry in India is forecast to increase to Rs. 8.6 trillion (US$ 132.84 billion) by FY22 from Rs. 4 trillion (US$ 61.79 billion) in FY17 at a CAGR of 16–17%. The Government of India is planning to increase public health spending to 2.5% of the country's GDP by 2025.

Health care market size in India 2022-2027

In terms of revenue, the healthcare market was valued at INR 21.14 Trn in FY 2021. It is estimated to reach INR 110.21 Trn by FY 2027, expanding at a compounding annual growth rate (CAGR) of ~30.70% during the FY 2022 – FY 2027 forecast period.

The market for hospitals, outpatient care services, medical tourism, and diagnostic services witnessed a decline in 2020. However, it started recovering from the second quarter of 2021. The growing prevalence of chronic, lifestyle-related diseases, and inflow of funds from public as well as private investors propelled growth of the market. The exposure to advanced technologies such as telehealth and telemedicine in the healthcare ecosystem also bolstered the growth of the healthcare market. The government aims to increase healthcare spending to 3% of the Gross Domestic Product (GDP) by the end of 2022.

The market is segmented into outpatient care centers, hospitals, pharmaceuticals, medical equipment and supplies, diagnostic services, digital healthcare, research and development, medical insurance, and medical tourism. The pharmaceuticals segment was the second-largest segment in FY 2021 with a market share of 16.50%. It is estimated to grow to a market share of 19.91% in FY 2027. The digital healthcare segment also demonstrated a notable market share of 13.00% in FY 2021 and is anticipated to increase by 4.44% by FY 2027.

Healthcare segments were severely affected during the first wave of the pandemic. Meanwhile, digital healthcare, pharmaceuticals, and medical equipment and supplies witnessed remarkable growth during the COVID-19 outbreak. However, in 2021, with the spread of COVID-19 under control to a certain extent, the dispersion in the growth rate of healthcare market segments started reducing. Despite this, the India healthcare industry is anticipated to witness significant growth in the coming few years. In India, the medical insurance market is expected to expand consistently because people are apprehensive about suffering from complicated ailments and bearing the expenditure of the treatment.

Business Problem:-

India’s Insurance industry is one of the premium sectors experiencing upward growth. This upward growth of the insurance industry can be attributed to growing incomes and increasing awareness in the industry. India is the fifth largest life insurance market in the world's emerging insurance markets, growing at a rate of 32-34% each year.

The insurance industry of India has 57 insurance companies - 24 are in the life insurance business, while 34 are non-life insurers.

The future looks promising for the life insurance industry with several changes in the regulatory framework which will lead to further changes in the way the industry conducts its business and engages with its customers. Life insurance industry in the country is expected to increase by 14-15% annually for the next three to five years.

The market share of private sector companies in the general and health insurance market increased from 47.97% in FY19 to 48.03% in FY20. In the life insurance segment, private players held a market share of 33.78% in premium underwritten services in FY20.

Healthcare inflation According to the Economic Times, inflation in healthcare is growing at a rate of 12 to 18%! This includes overall costs such as cost of medicines, hospital admission charges, cost of various treatments, medical advancements and so on. Due to the rise in these expenses, insurer too needs to increase the sum insured every year i.e. coverage to be able to cover for these costs when we make a claim. This is primarily why there is consequently an increase in our health insurance premium too when we renew for the new policy year.

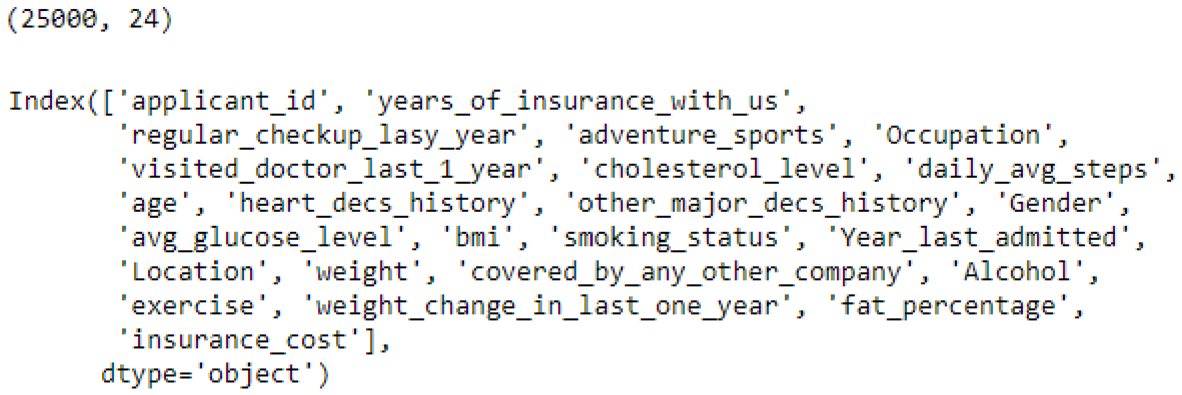
2. EDA and Business Implication

Data Report:-

- Understanding how data was collected in terms of time, frequency and methodology
- This section aims at giving how the data is collected.
- Data collection is the process of gathering, measuring, and analyzing accurate data from a variety of relevant sources to find answers to research problems, answer questions, evaluate outcomes, and forecast trends and probabilities.
- This is a capstone project driven by the great learning hence the dataset that we use in our analysis is provided from the learning platform.

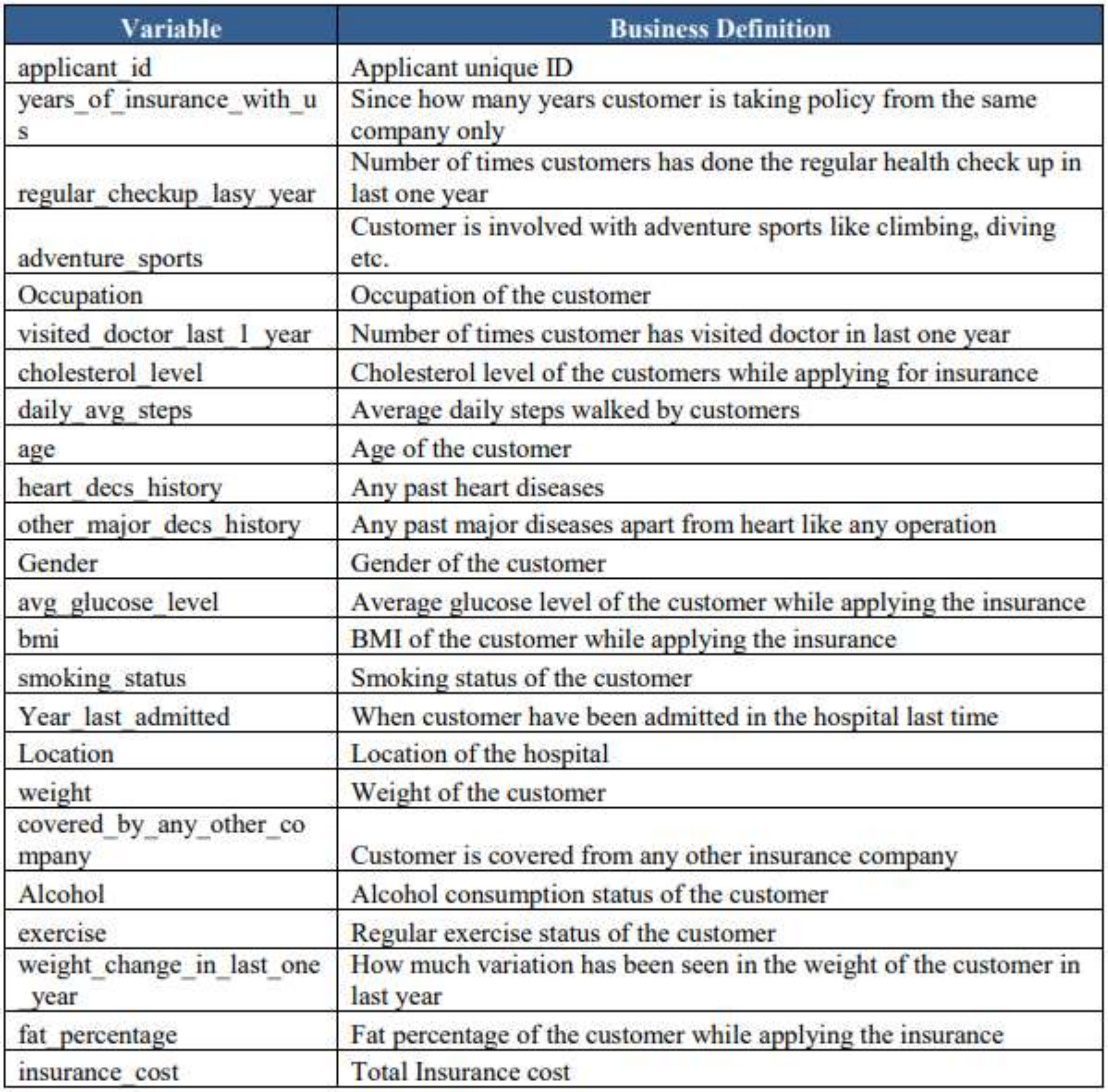
Visual inspection of data (rows, columns, descriptive details)

- Dataset consists of 25000 rows and 24 columns.



```
(25000, 24)

Index(['applicant_id', 'years_of_insurance_with_us',
       'regular_checkup_lasy_year', 'adventure_sports', 'Occupation',
       'visited_doctor_last_1_year', 'cholesterol_level', 'daily_avg_steps',
       'age', 'heart_decs_history', 'other_major_decs_history', 'Gender',
       'avg_glucose_level', 'bmi', 'smoking_status', 'Year_last_admitted',
       'Location', 'weight', 'covered_by_any_other_company', 'Alcohol',
       'exercise', 'weight_change_in_last_one_year', 'fat_percentage',
       'insurance_cost'],
      dtype='object')
```

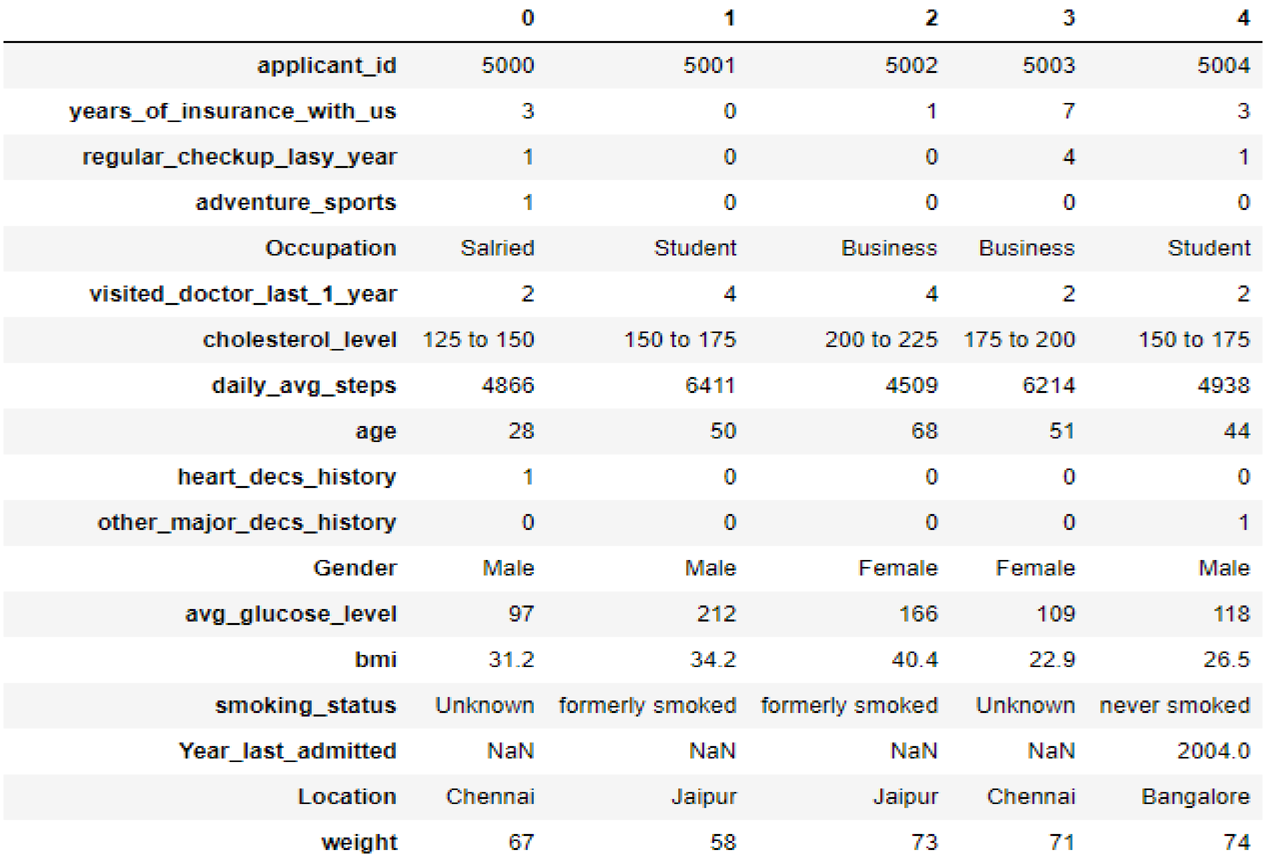
- Let’s discuss the data dictionary for detail analysis and understanding the variables of the dataset.



Variable	Business Definition
applicant_id	Applicant unique ID
years_of_insurance_with_us	Since how many years customer is taking policy from the same company only
regular_checkup_lasy_year	Number of times customers has done the regular health check up in last one year
adventure_sports	Customer is involved with adventure sports like climbing, diving etc.
Occupation	Occupation of the customer
visited_doctor_last_1_year	Number of times customer has visited doctor in last one year
cholesterol_level	Cholesterol level of the customers while applying for insurance
daily_avg_steps	Average daily steps walked by customers
age	Age of the customer
heart_decs_history	Any past heart diseases
other_major_decs_history	Any past major diseases apart from heart like any operation
Gender	Gender of the customer
avg_glucose_level	Average glucose level of the customer while applying the insurance
bmi	BMI of the customer while applying the insurance
smoking_status	Smoking status of the customer
Year_last_admitted	When customer have been admitted in the hospital last time
Location	Location of the hospital
weight	Weight of the customer
covered_by_any_other_company	Customer is covered from any other insurance company
Alcohol	Alcohol consumption status of the customer
exercise	Regular exercise status of the customer
weight_change_in_last_one_year	How much variation has been seen in the weight of the customer in last year
fat_percentage	Fat percentage of the customer while applying the insurance
insurance_cost	Total Insurance cost

Image1.Data_Dictionary

The below table shows the sample of the dataset

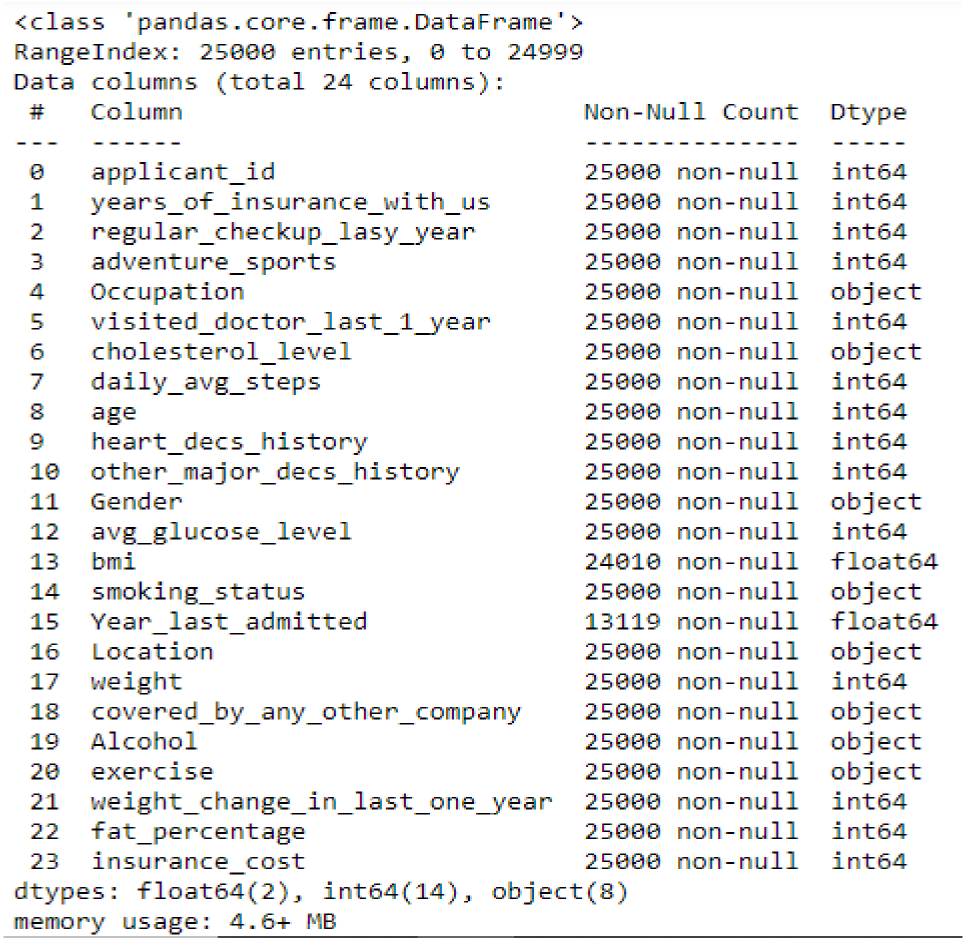


	0	1	2	3	4
applicant_id	5000	5001	5002	5003	5004
years_of_insurance_with_us	3	0	1	7	3
regular_checkup_lasy_year	1	0	0	4	1
adventure_sports	1	0	0	0	0
Occupation	Salried	Student	Business	Business	Student
visited_doctor_last_1_year	2	4	4	2	2
cholesterol_level	125 to 150	150 to 175	200 to 225	175 to 200	150 to 175
daily_avg_steps	4866	6411	4509	6214	4938
age	28	50	68	51	44
heart_decs_history	1	0	0	0	0
other_major_decs_history	0	0	0	0	1
Gender	Male	Male	Female	Female	Male
avg_glucose_level	97	212	166	109	118
bmi	31.2	34.2	40.4	22.9	26.5
smoking_status	Unknown	formerly smoked	formerly smoked	Unknown	never smoked
Year_last_admitted	NaN	NaN	NaN	NaN	2004.0
Location	Chennai	Jaipur	Jaipur	Chennai	Bangalore
weight	67	58	73	71	74

Table1.Sample_Dataset

The above table represent the top 5 rows of the dataset

Data Info:-



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 24 columns):
 #   Column                         Non-Null Count  Dtype  
---  ------                         --------------  -----  
 0   applicant_id                   25000 non-null  int64  
 1   years_of_insurance_with_us     25000 non-null  int64  
 2   regular_checkup_lasy_year      25000 non-null  int64  
 3   adventure_sports               25000 non-null  int64  
 4   Occupation                     25000 non-null  object 
 5   visited_doctor_last_1_year     25000 non-null  int64  
 6   cholesterol_level              25000 non-null  object 
 7   daily_avg_steps                25000 non-null  int64  
 8   age                            25000 non-null  int64  
 9   heart_decs_history             25000 non-null  int64  
 10  other_major_decs_history       25000 non-null  int64  
 11  Gender                         25000 non-null  object 
 12  avg_glucose_level              25000 non-null  int64  
 13  bmi                            24010 non-null  float64
 14  smoking_status                 25000 non-null  object 
 15  Year_last_admitted             13119 non-null  float64
 16  Location                       25000 non-null  object 
 17  weight                         25000 non-null  int64  
 18  covered_by_any_other_company   25000 non-null  object 
 19  Alcohol                        25000 non-null  object 
 20  exercise                       25000 non-null  object 
 21  weight_change_in_last_one_year 25000 non-null  int64  
 22  fat_percentage                 25000 non-null  int64  
 23  insurance_cost                 25000 non-null  int64  
dtypes: float64(2), int64(14), object(8)
memory usage: 4.6+ MB
```

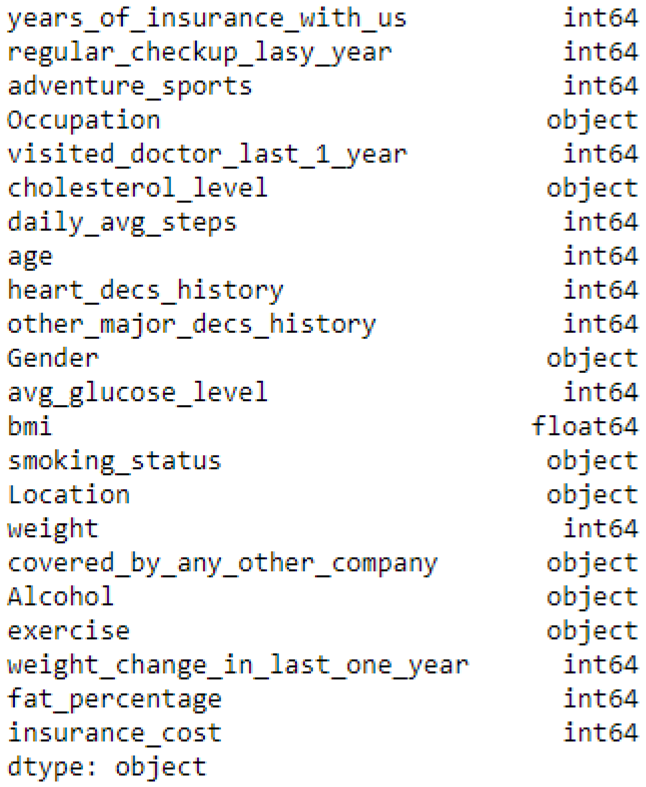
The dataset contains 25000 rows and 24 columns out of which:

- Columns are of float data type.
- 8 Columns are of Object data type.
- 14 Columns are of int data type.

Variable Identification and Typecasting

- This is one of the most important steps, Why?
- Because pandas is not very good when it comes to recognizing the datatype of the important variables. So in this section, we will be analyzing the datatypes of each variables and converting them to respective types.

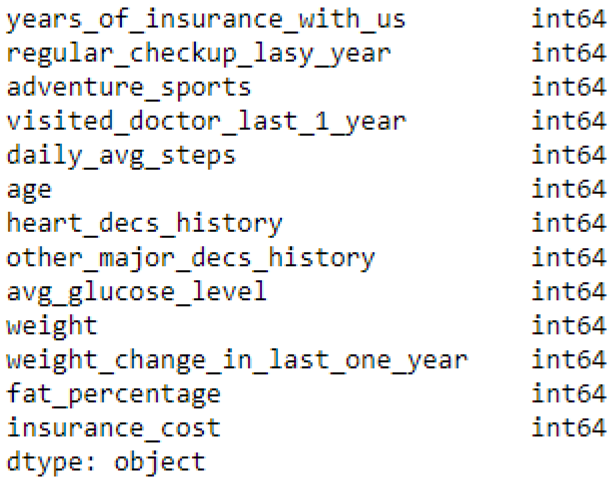
Let’s check the data types of the data



```
years_of_insurance_with_us          int64
regular_checkup_lasy_year           int64
adventure_sports                    int64
Occupation                         object
visited_doctor_last_1_year          int64
cholesterol_level                  object
daily_avg_steps                     int64
age                                 int64
heart_decs_history                  int64
other_major_decs_history            int64
Gender                             object
avg_glucose_level                   int64
bmi                               float64
smoking_status                     object
Location                           object
weight                              int64
covered_by_any_other_company       object
Alcohol                            object
exercise                           object
weight_change_in_last_one_year      int64
fat_percentage                      int64
insurance_cost                      int64
dtype: object
```

There are a lot of variables visible at one, so let's narrow this down by looking at one datatype at once. We will start with int

Integer Data Type



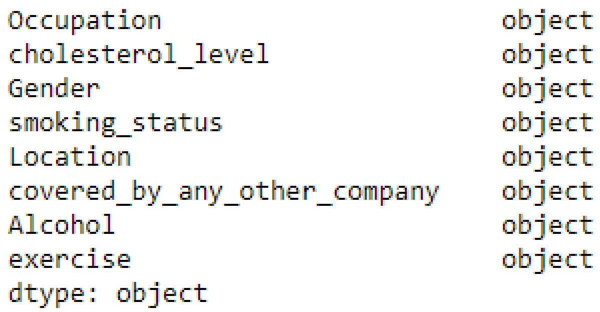
```
years_of_insurance_with_us        int64
regular_checkup_lasy_year         int64
adventure_sports                  int64
visited_doctor_last_1_year        int64
daily_avg_steps                   int64
age                               int64
heart_decs_history                int64
other_major_decs_history          int64
avg_glucose_level                 int64
weight                            int64
weight_change_in_last_one_year    int64
fat_percentage                    int64
insurance_cost                    int64
dtype: object
```

All the variables data types are correctly defined so we will not perform type casting on above variable.

Observation:-

- Years_of_insurance_with_us are a unique number of years customers are with the company, therefore it should be converted to category.
- Regular_chekup_lasy_year again represents number of years, therefore it should be convereted to category.
- Adventure_sports have binary value whether customer has interest in any adventure sports like trekking, cycling etc or not, it should be convereted to category.
- Visited_doctor_last_one_year, Heart_decs_history, other_major_decs_history and weight_change_in_last_one_year also have binary values, therefore should be converted to category.
- Daily_avg_steps, age, Avg_glucose_level, weight, fat_percentage and insurance_cost are numbers and hence we are okay with them as integer.

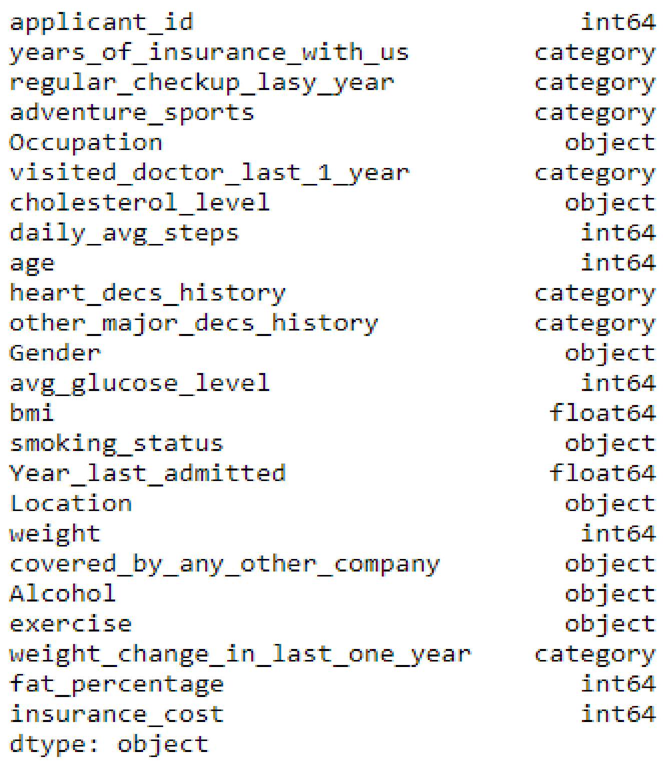
Object data types:-



```
Occupation                  object
cholesterol_level           object
Gender                      object
smoking_status              object
Location                    object
covered_by_any_other_company    object
Alcohol                     object
exercise                    object
dtype: object
```

All the variables data types are correctly defined so we will not perform type casting on above variable.

The below result shows the data types of all the variables



```
applicant_id                         int64
years_of_insurance_with_us        category
regular_checkup_lasy_year         category
adventure_sports                  category
Occupation                          object
visited_doctor_last_1_year        category
cholesterol_level                   object
daily_avg_steps                      int64
age                                  int64
heart_decs_history                category
other_major_decs_history          category
Gender                              object
avg_glucose_level                    int64
bmi                                float64
smoking_status                      object
Year_last_admitted                 float64
Location                            object
weight                               int64
covered_by_any_other_company        object
Alcohol                             object
exercise                            object
weight_change_in_last_one_year    category
fat_percentage                       int64
insurance_cost                       int64
dtype: object
```

EDA (Exploratory data analysis) :-

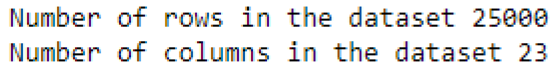
The EDA aims at cleaning the data to make it ready for the analysis. It targets to give the Univariate analysis by finding the distribution of the continuous and categorical data, And Bivariate Analysis by finding the relationship and correlation between different variables in the dataset.

Exploratory data analysis can help detect obvious errors, identify outliers in datasets, understand relationships, unearth important factors, find patterns within data, and provide new insights.

Removal of unwanted variables

We will drop applicant_id column as it’s a mere identifier and doesn’t contribute much to our analysis.

Shape of the dataset after dropping a column:-



```
Number of rows in the dataset 25000
Number of columns in the dataset 23
```

Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)

The purpose of Univariate Analysis is to find out which variables have clear separation for the target variable – separation of mean & median of continuation variables and their skewness affecting the target variable.

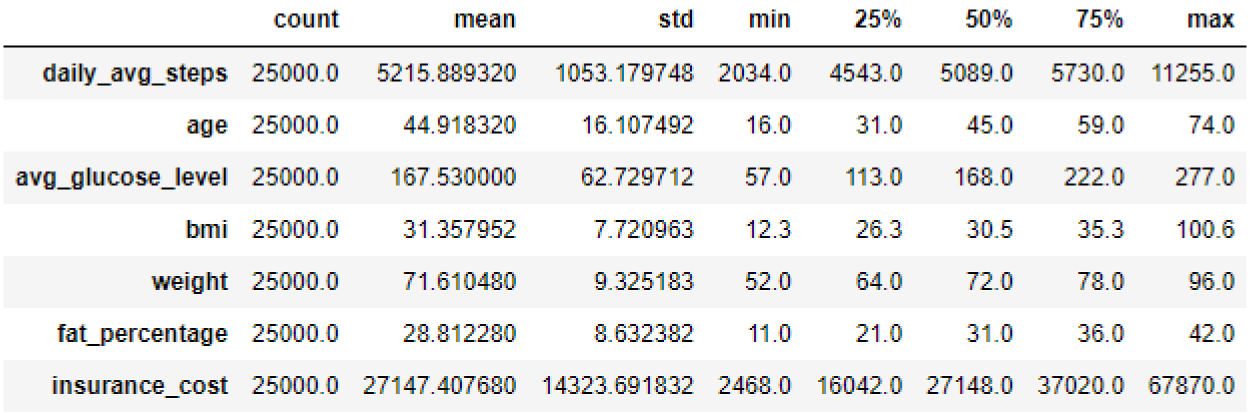
When dealing with numerical variables, we have to check their properties like:

- **Mean**
- **Median**
- **Kurtosis/skewness**
- **distribution/range**

Univariate Analysis of continuous Variables

First, we will do the univariate analysis of continuous variables. We will first use the describe function to get the descriptive statistics of continuous variables.

Data description:-



	count	mean	std	min	25%	50%	75%	max
daily_avg_steps	25000.0	5215.889320	1053.179748	2034.0	4543.0	5089.0	5730.0	11255.0
age	25000.0	44.918320	16.107492	16.0	31.0	45.0	59.0	74.0
avg_glucose_level	25000.0	167.530000	62.729712	57.0	113.0	168.0	222.0	277.0
bmi	25000.0	31.357952	7.720963	12.3	26.3	30.5	35.3	100.6
weight	25000.0	71.610480	9.325183	52.0	64.0	72.0	78.0	96.0
fat_percentage	25000.0	28.812280	8.632382	11.0	21.0	31.0	36.0	42.0
insurance_cost	25000.0	27147.407680	14323.691832	2468.0	16042.0	27148.0	37020.0	67870.0

Table 2.Dataset Summary

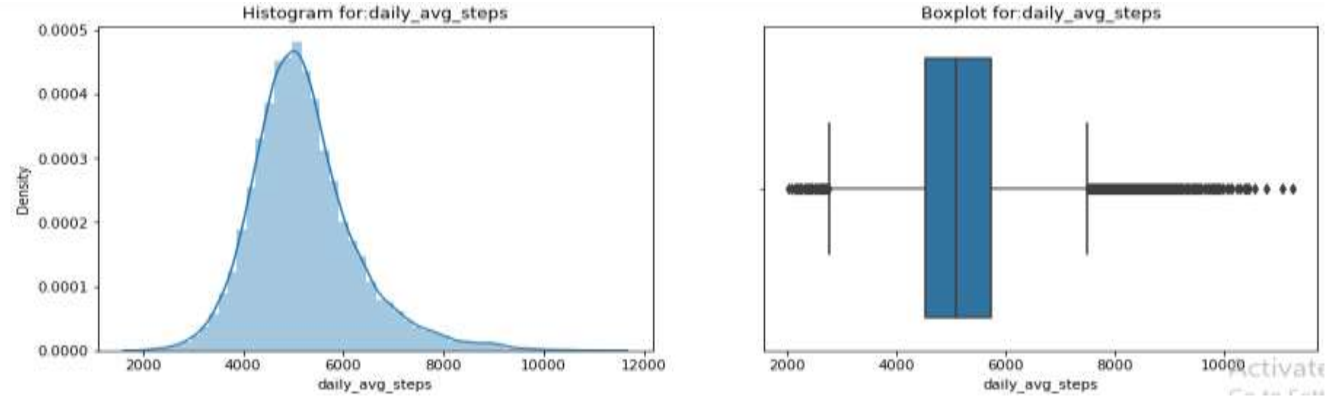
The description of the data is used to transform the data into information. The five number summary, which forms the basis for a boxplot, is a good example of summarizing data. The above table is statistical summary of the dataset.

Key Observations:

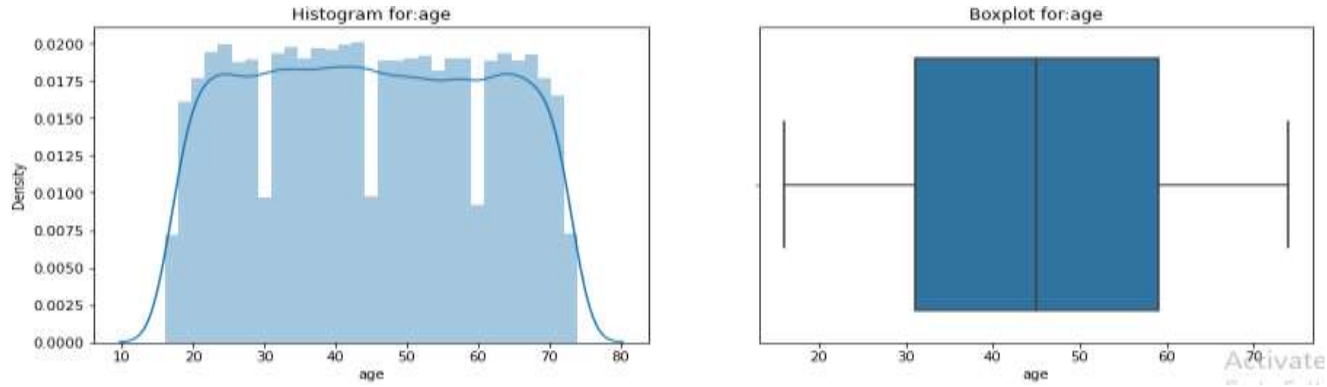
- We can see that the minimum age among the applicants is 16. This tells that there are young people also who take health insurance.
- Dataset contain the health and habits of the customer, insurance cost of an individual varies from 2468.00 to 67870.00, age is from 16 to 74 years, daily_avg_steps varies from 2034 to 11255, avg_glucose_level varies from 57 to 277 , bmi varies from 12.3 to 100.6
- Year_last_admitted and bmi have missing value.
- Since the mean and median values are very far apart the variables seem to be skewed
- By looking at the dataset, it appears that there are outliers in the variables. The same is visible from the distribution of 5 values (min, 25 percentile, 50 percentile, 75 percentile and maximum)

Now we will plot histograms for continuous columns to see the frequency distribution of values of columns.

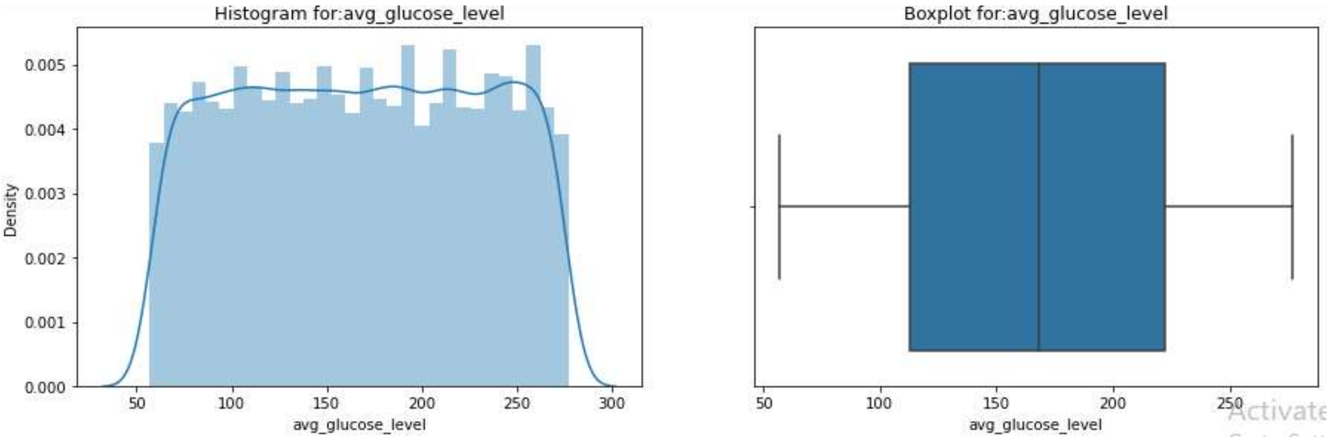
Histogram and Boxplot for daily_avg_steps



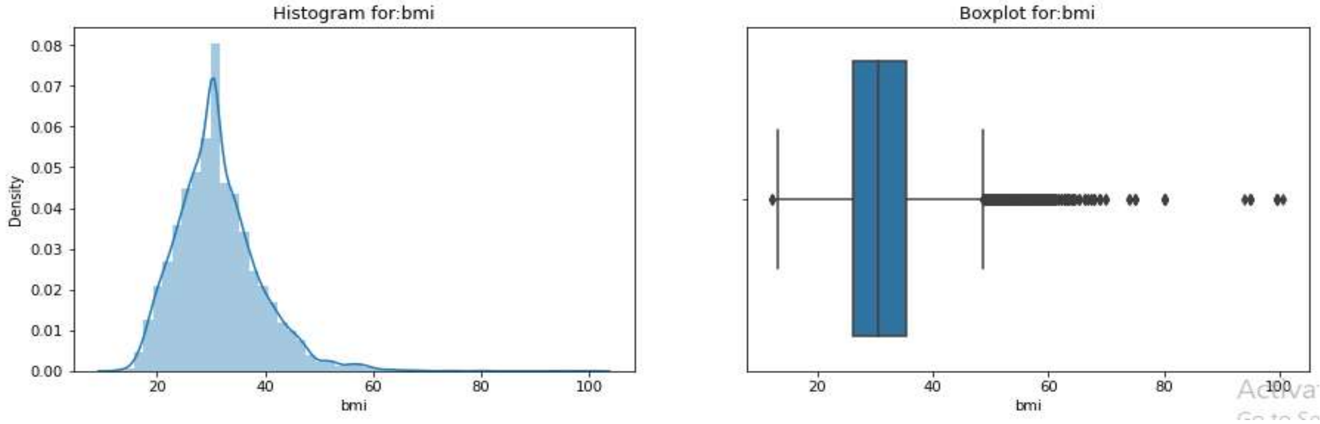
Histogram and Boxplot for age



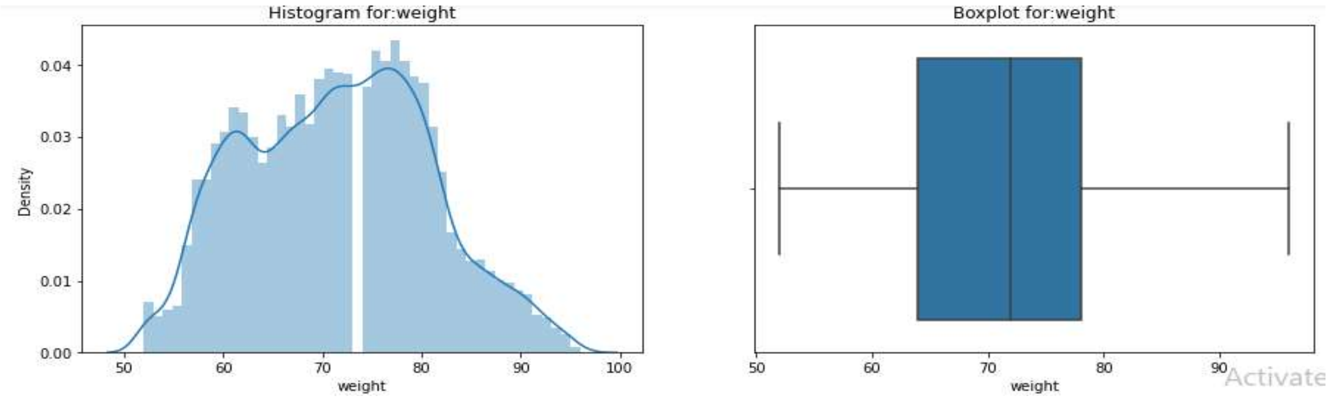
Histogram and Boxplot for avg_glucose_level



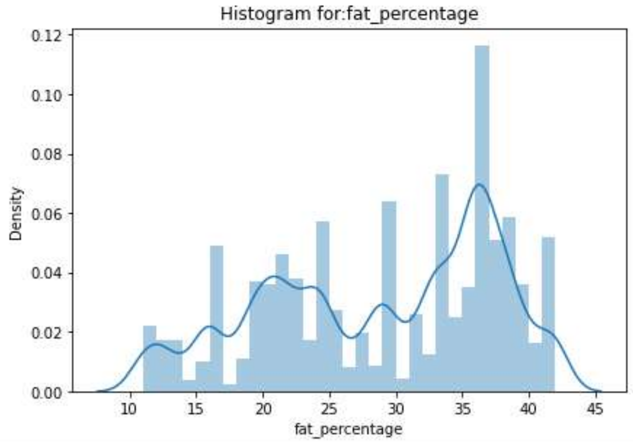
Histogram and Boxplot for bmi

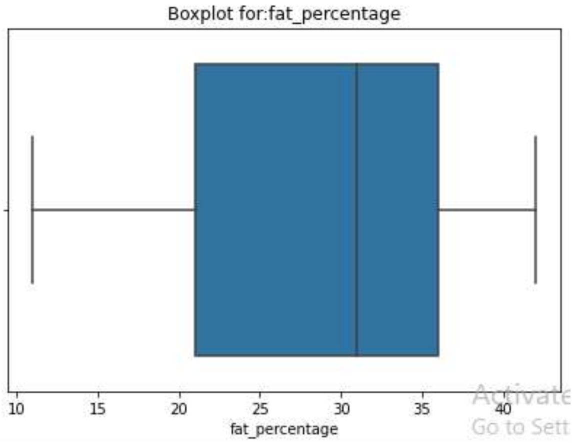


Histogram and Boxplot for weight

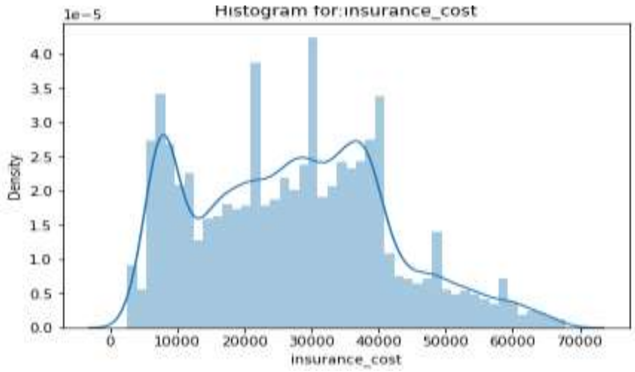


Histogram and Boxplot for fat_percentage





Histogram and Boxplot for insurance_cost



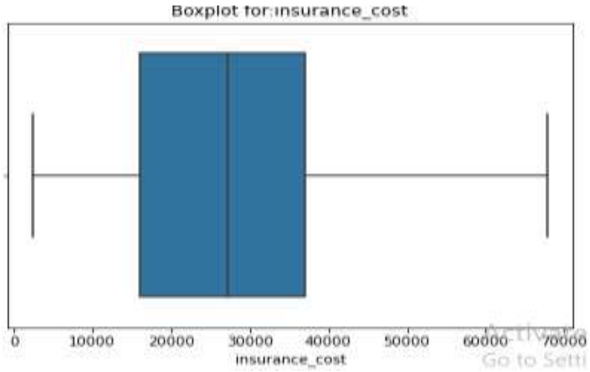
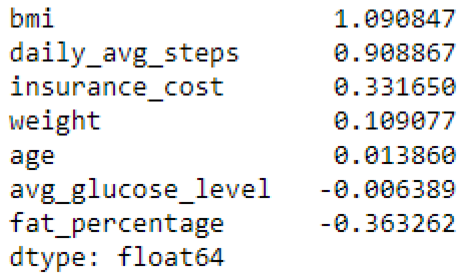


Image2.Histogram and Boxplot of continuous variable

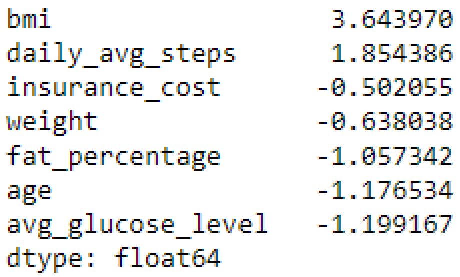
Let’s Check the skewness and kurtosis of the data

Skewness:-



```
bmi                 1.090847
daily_avg_steps     0.908867
insurance_cost      0.331650
weight              0.109077
age                 0.013860
avg_glucose_level  -0.006389
fat_percentage     -0.363262
dtype: float64
```

Kurtosis:-



```
bmi                   3.643970
daily_avg_steps       1.854386
insurance_cost       -0.502055
weight               -0.638038
fat_percentage       -1.057342
age                  -1.176534
avg_glucose_level    -1.199167
dtype: float64
```

Observations:-

- **Average daily steps**

  Most of the health insurance is taken by the customers whose avg daily steps ranges between 5000 and 6000.

  Skewness is 0.90 means that avg daily steps is positively or slightly skewed

  Kurtosis 1.85 means most of the data points are present in high proximity of the mean.

  Outliers are present in the data.

- **Age:-**

  Most of the customers Age ranges between 31 and 74

  Median of the age is 45.

  Skewness is 0.01 means that the data are nearly symmetrical.

  Kurtosis -1.17 means most of the data points are present in high proximity of the mean.

- **Average glucose level:-**

  Most of the customers Avg_glucose_level ranges between 113 and 222.

  Median of glucose level is 168.

  Skewness is 0.006 means that the data are nearly symmetrical.

  Kurtosis -1.19 means most of the data points are present in high proximity of the mean.

- **Bmi:-**

  Most of the customers Bmi ranges between 26 and 35

  Median of bmi is 305.

  Skewness is 1.09 means that the data are extremely skewed.

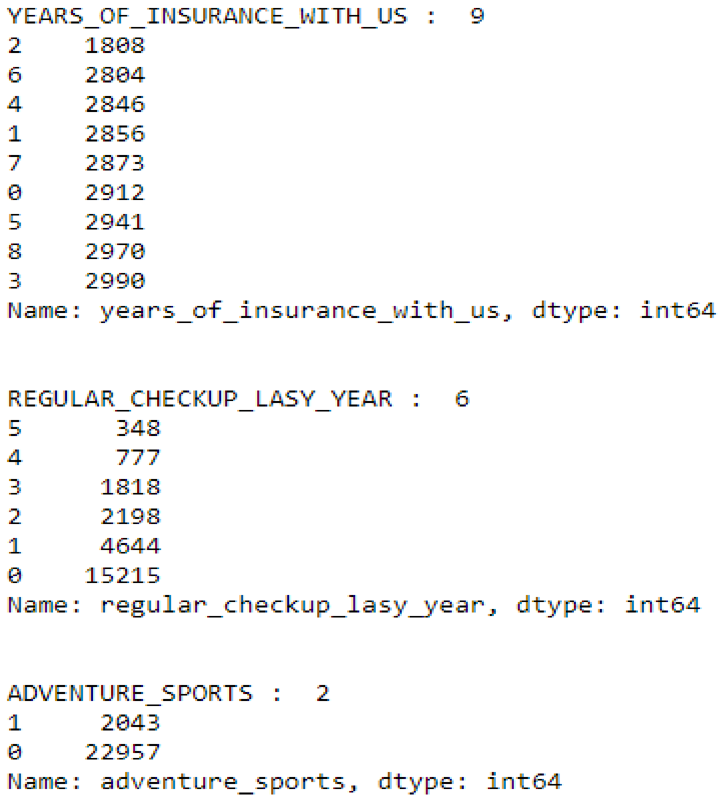
  Kurtosis 3.64 means which means there are more chances of outliers.

- **Weight:-**
  Most of the customers Weight ranges between 64 and 78.
  Median of weight is 72.
  Skewness is 0.10 means that the data are nearly symmetric.
  Kurtosis -0.63 means most of the data points are present in high proximity of the mean.

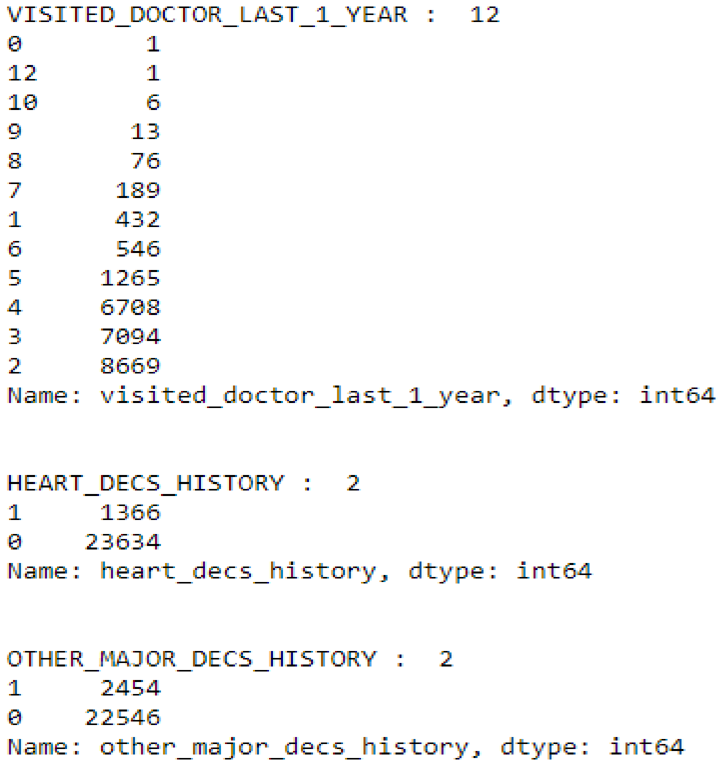
- **Fat Percentage:-**
  Fat percentage of most of the customers ranges between 21 and 36.
  Median of the weight is 31.
  Skewness is -0.36 (negatively skewed) means that the data is slightly skewed.
  Kurtosis -1.05 means most of the data points are present in high proximity of the mean.

- **Insurance cost:-**
  Insurance cost of most the customers ranges between 16042 and 37020.
  Median of the insurance cost is 27148.
  Skewness is 0.33 (positively skewed) means that the data is slightly skewed.
  Kurtosis -0.50 means most of the data points are present in high proximity of the mean.

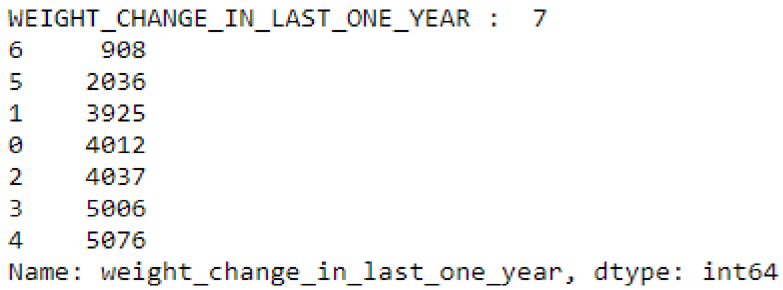
Univariate analysis of discrete numerical variable:-



```
YEARS_OF_INSURANCE_WITH_US :   9
2     1808
6     2804
4     2846
1     2856
7     2873
0     2912
5     2941
8     2970
3     2990
Name: years_of_insurance_with_us, dtype: int64


REGULAR_CHECKUP_LASY_YEAR :   6
5       348
4       777
3      1818
2      2198
1      4644
0     15215
Name: regular_checkup_lasy_year, dtype: int64


ADVENTURE_SPORTS :   2
1      2043
0     22957
Name: adventure_sports, dtype: int64
```



```
VISITED_DOCTOR_LAST_1_YEAR :   12
0         1
12        1
10        6
9        13
8        76
7       189
1       432
6       546
5      1265
4      6708
3      7094
2      8669
Name: visited_doctor_last_1_year, dtype: int64


HEART_DECS_HISTORY :   2
1      1366
0     23634
Name: heart_decs_history, dtype: int64


OTHER_MAJOR_DECS_HISTORY :   2
1      2454
0     22546
Name: other_major_decs_history, dtype: int64
```



```
WEIGHT_CHANGE_IN_LAST_ONE_YEAR :  7
6     908
5    2036
1    3925
0    4012
2    4037
3    5006
4    5076
Name: weight_change_in_last_one_year, dtype: int64
```

Countplot for years_of_insurance with us

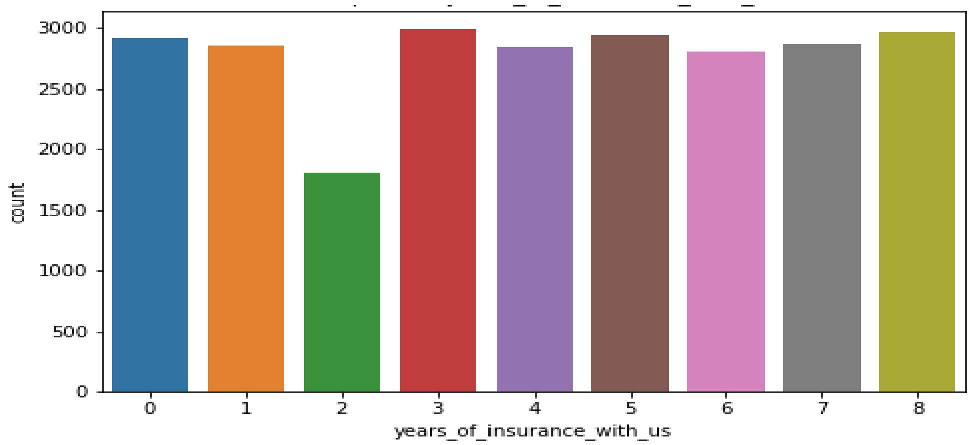


Image3.Countplot_years_of_insurance_with_us

The above countplot shows that people are insured with us from a very long time. It is consistence from past 8 years. The customers who are with us for 2 years are less as compared to other years.

CountPlot for regular_checkup_lasy_year

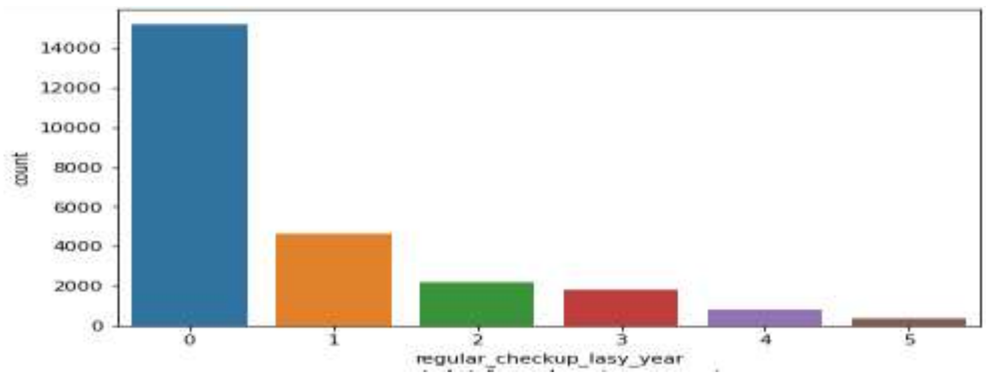


Image4.Countplot_regularcheckup_lasy_year

From the above countplot shows the trend by which we can say that most of the customers haven't done the regular heath checkup in last one year. And there are very less number of customers who went for regular checkup.

CountPlot for Adventure_sports

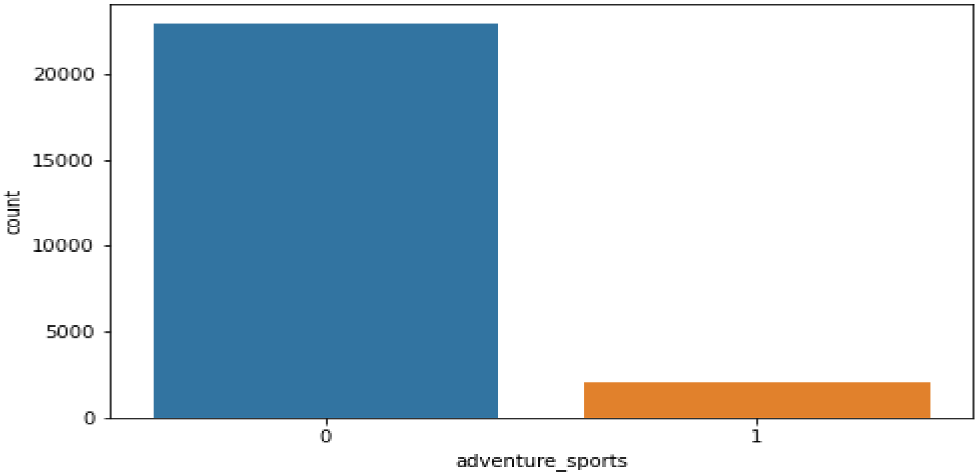


Image4.Countplot_adventure_sports

From the above countplot we can say that most of the customers insured with health insurance doesn't involved with adventure sports like climbing, trekking, cycling etc.

CountPlot for visited_doctor_last_1_year

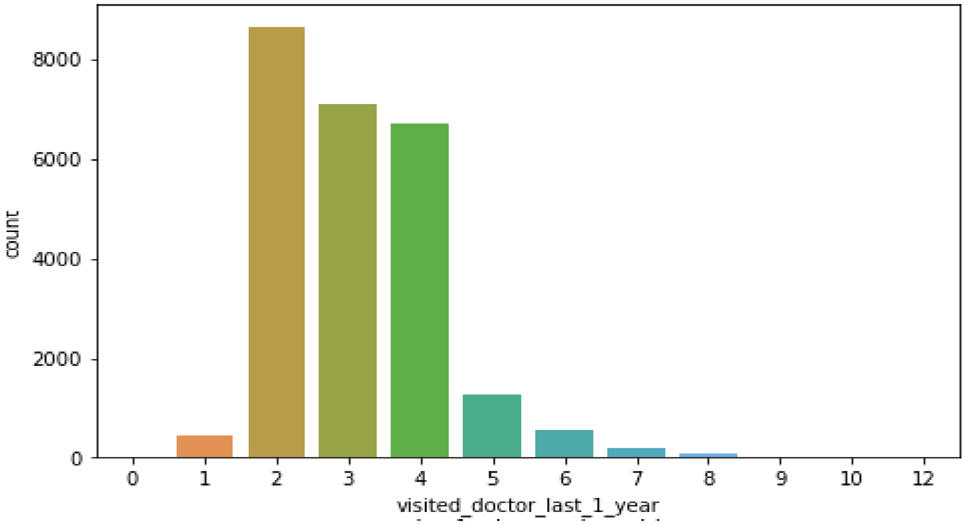


Image6.Countplot_visited_doctor_last_1_year

The above countplot shows that most of the customer who have taken the health insurance visited the doctor 2, 3 and 4 times in last one year.

CountPlot for heart_decs_history

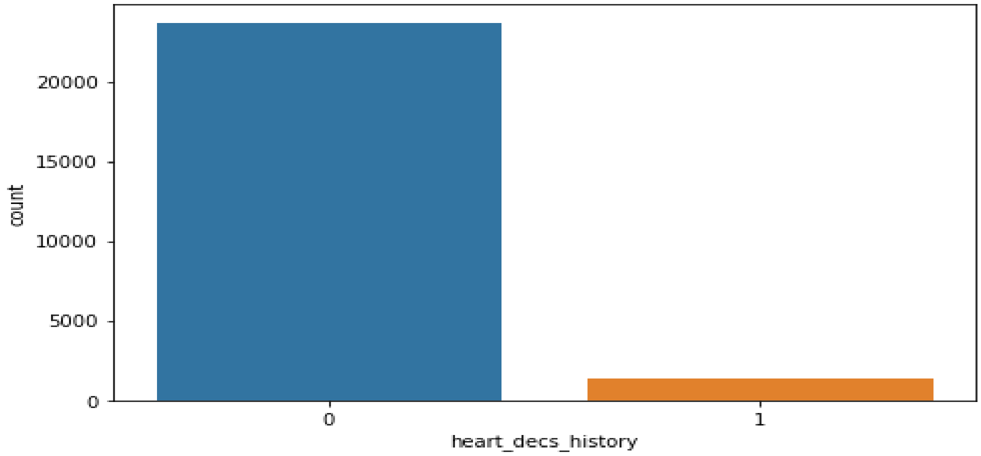


Image7.Countplot_heart_decs_history

From the above countplot it shows that most the number of customers doesn't have any past heart disease history.

CountPlot for other_major_decs_history

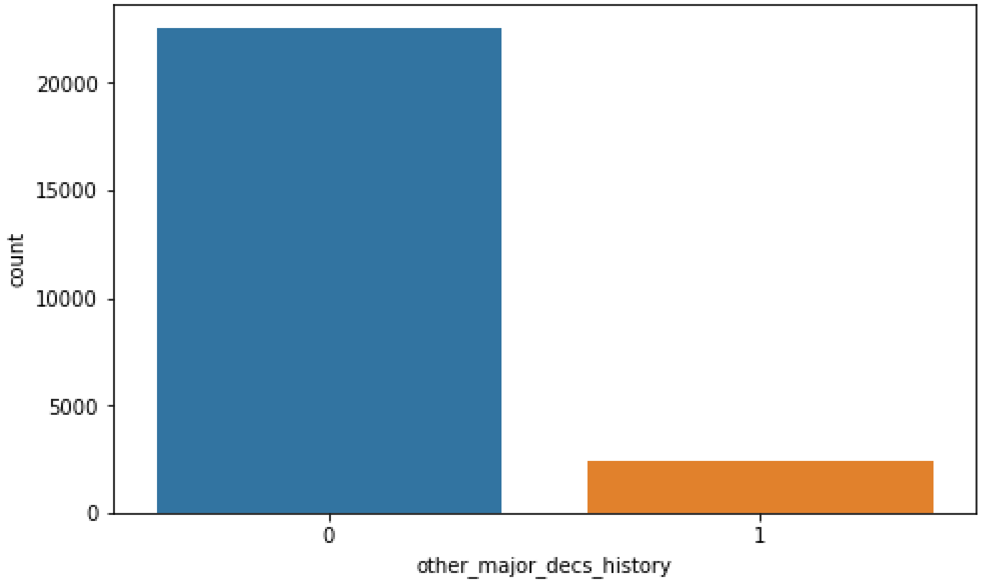


Image8.Countplot_other_major_decs_history

From the above countplot it can be seen that most of the customers doesn't have any past major diseases apart from heart like any operation.

CountPlot for weight change in last one year

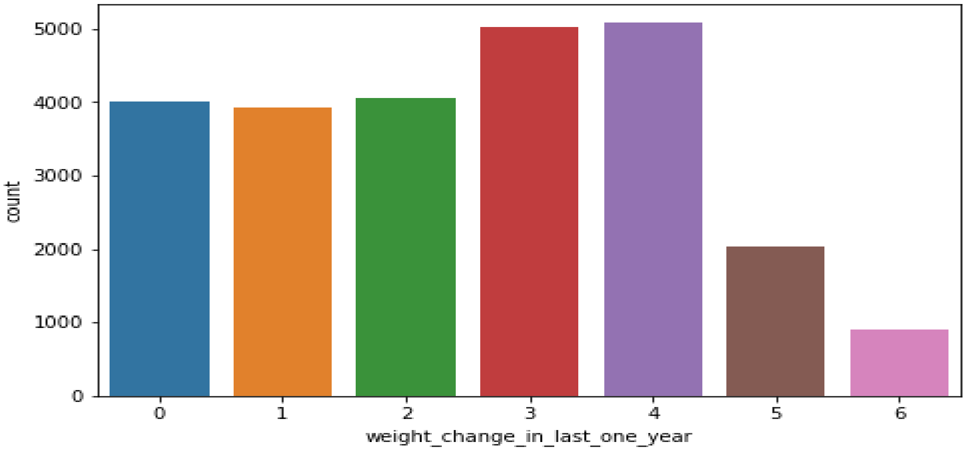


Image9.Countplot_weight_change_in_last_1year

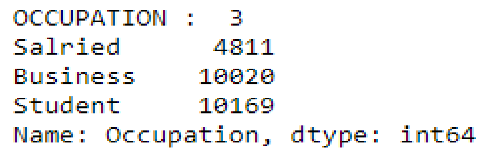
The weight change in last one year also affect the number of customers who are insured with us. As the weight changes the chances of getting ill which will also affect the insurance cost.

Observation:-

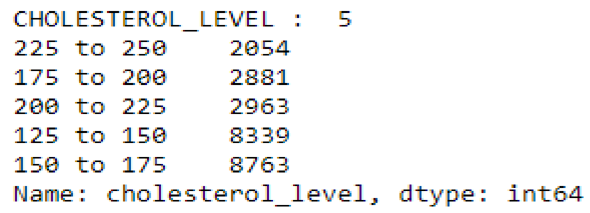
- years_of_insurance_with_us variable has 9 unique values, regular_checkup_lasy_year has 6 unique values, visited_doctor_last_1_year has 13 unique values and weight_change_in_last_one_year has 7 unique values.
- Most of the customers are insured with us from a very long time. It is consistence from past 8 years.
- Most of the customers haven't done the regular heath checkup in last one year.
- Most of the customers insured with health insurance doesn't involved with adventure sports like climbing, trekking, cycling etc.
- Most of the customer who have taken the health insurance visited the doctor 2, 3 and 4 times in last one year.
- The number of customers who doesn't have any past heart disease history are higher as compared to the customers who have heart disease.
- Most of the customers doesn't have any past major diseases apart from heart like any operation.

- The weight change in last one year also affect the number of customers who are insured with us. As the weight changes the chances of getting ill which will also affect the insurance cost. But the number of customers weight affected in last one year is 3 or 4 kg.

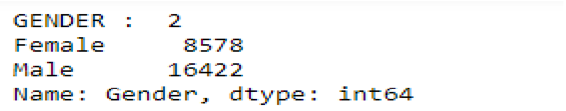
Univariate Analysis - Categorical Variable



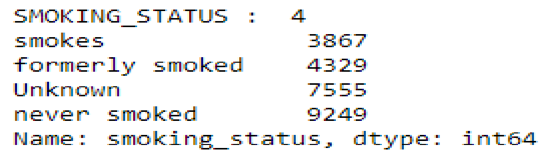
```
OCCUPATION :  3
Salried      4811
Business    10020
Student     10169
Name: Occupation, dtype: int64
```



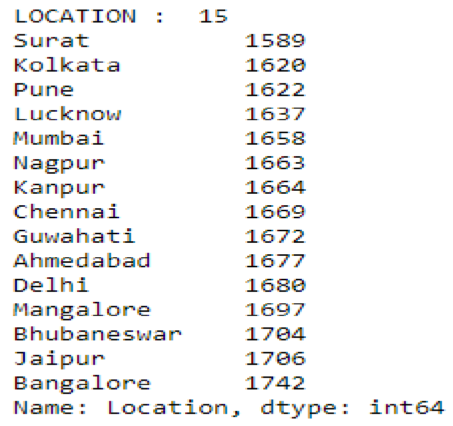
```
CHOLESTEROL_LEVEL :  5
225 to 250    2054
175 to 200    2881
200 to 225    2963
125 to 150    8339
150 to 175    8763
Name: cholesterol_level, dtype: int64
```



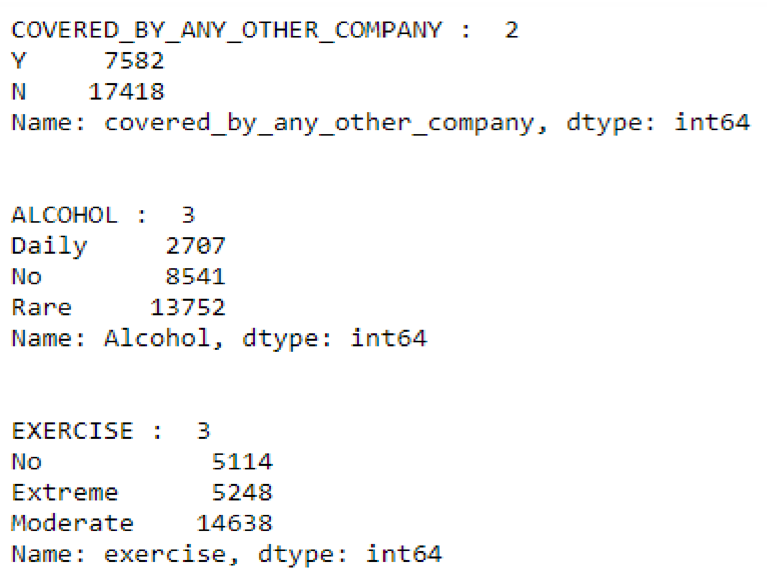
```
GENDER :  2
Female     8578
Male      16422
Name: Gender, dtype: int64
```



```
SMOKING_STATUS :  4
smokes             3867
formerly smoked    4329
Unknown            7555
never smoked       9249
Name: smoking_status, dtype: int64
```



```
LOCATION :  15
Surat          1589
Kolkata        1620
Pune           1622
Lucknow        1637
Mumbai         1658
Nagpur         1663
Kanpur         1664
Chennai        1669
Guwahati       1672
Ahmedabad      1677
Delhi          1680
Mangalore      1697
Bhubaneswar    1704
Jaipur         1706
Bangalore      1742
Name: Location, dtype: int64
```



```
COVERED_BY_ANY_OTHER_COMPANY :  2
Y     7582
N    17418
Name: covered_by_any_other_company, dtype: int64


ALCOHOL :  3
Daily     2707
No        8541
Rare     13752
Name: Alcohol, dtype: int64


EXERCISE :  3
No           5114
Extreme      5248
Moderate    14638
Name: exercise, dtype: int64
```

Count plot for Occupation:

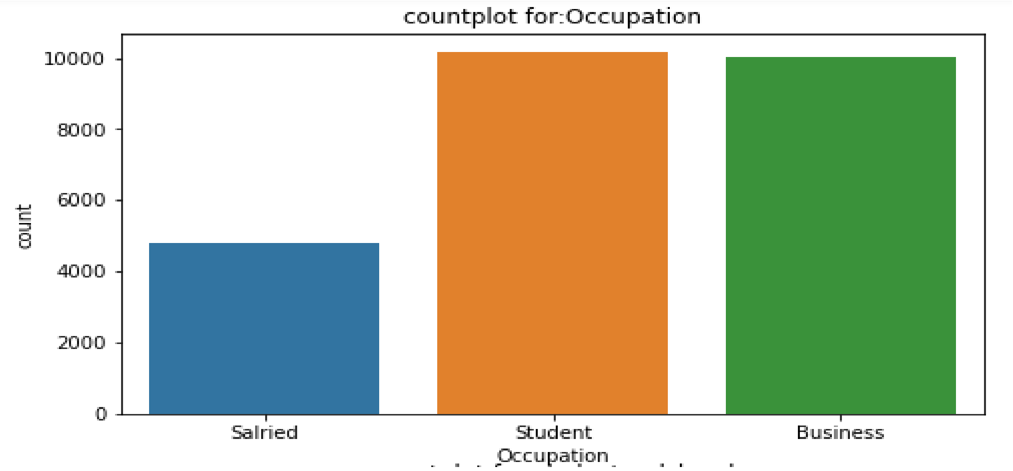


Image10.Countplot_Occupation

From the plot it can be infer that:-

- Customers whose occupation is student have taken max insurance.
- Customers whose occupation is salaried have taken less insurance.

Count plot for Cholesterol level:-

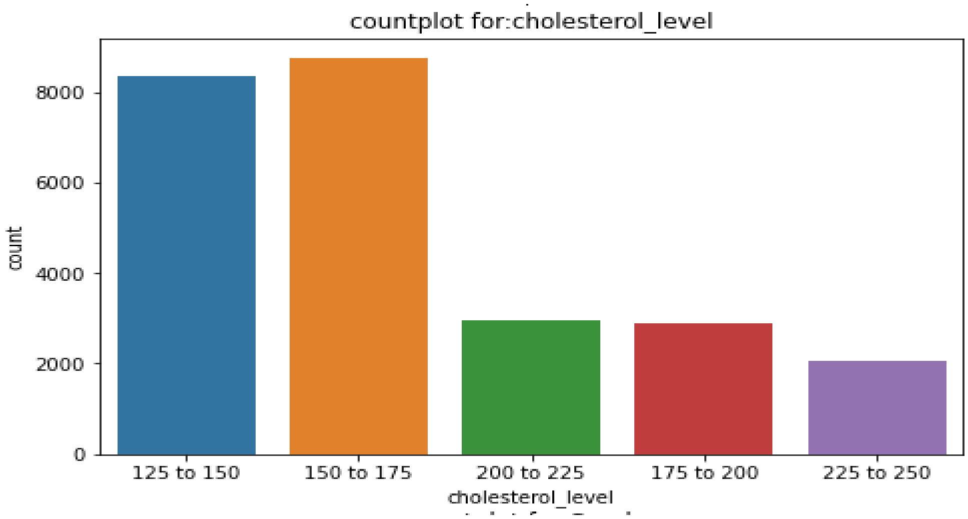


Image11.Countplot_Cholesterol_level

From the plot it can be infer that:-

- Customers whose Cholesterol level is between 150 to 175 and 125 to 150 have taken max number of health insurance. Which mean the healthy people are taking health insurance.
- Customers whose Cholesterol level is between 225 and 250 have taken min number of health insurance. Which mean the unhealthy people are not taking health insurance.
- The normal cholesterol level in body is less than 200 mg/dl.

Count plot for Gender

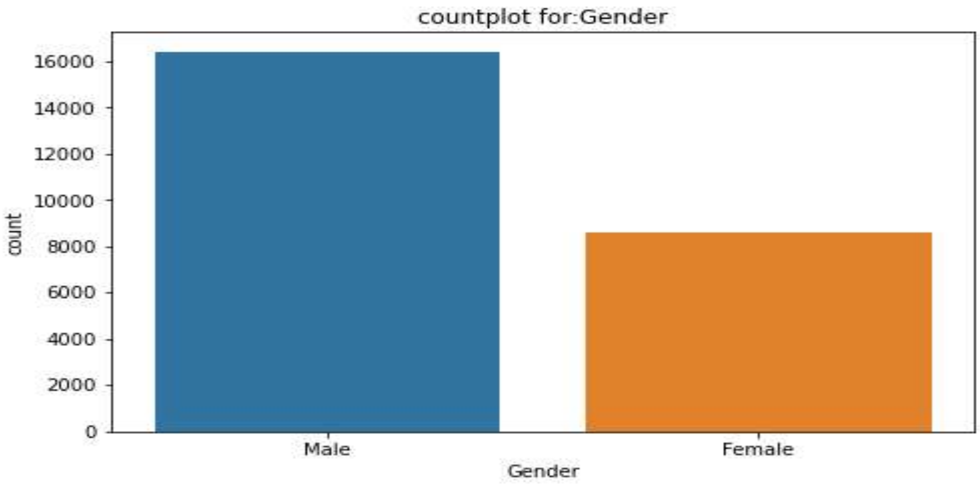


Image12.Countplot_gender

From the above plot it can be infer that Males are taking more health insurance than Female.

Count plot for Smoking Status:-

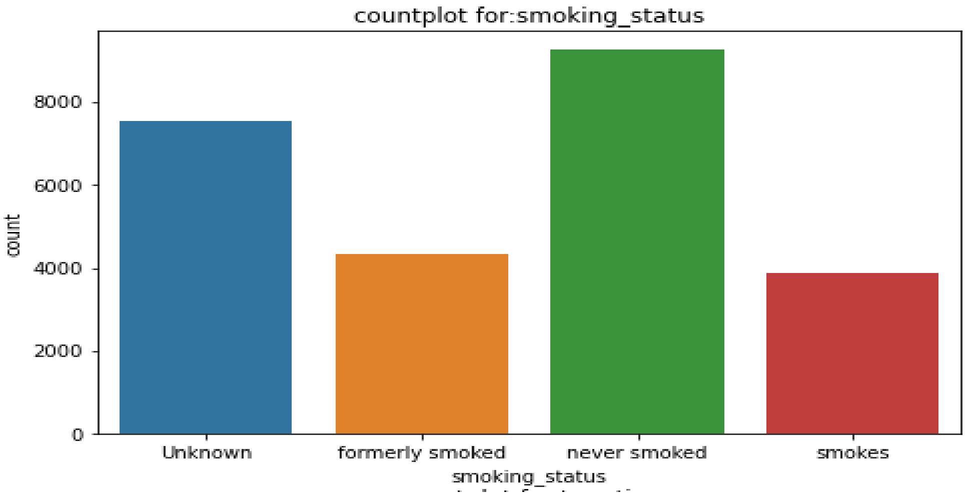


Image 13.Count plot of Smoking status

From the above plot it can be infer that:-

- The customers who never smoked have taken maximum health insurance.
- Whereas customers who smokes have taken min health insurance. Which means healthy people are taking health insurance rather than unhealthy people.

Count plot for Location

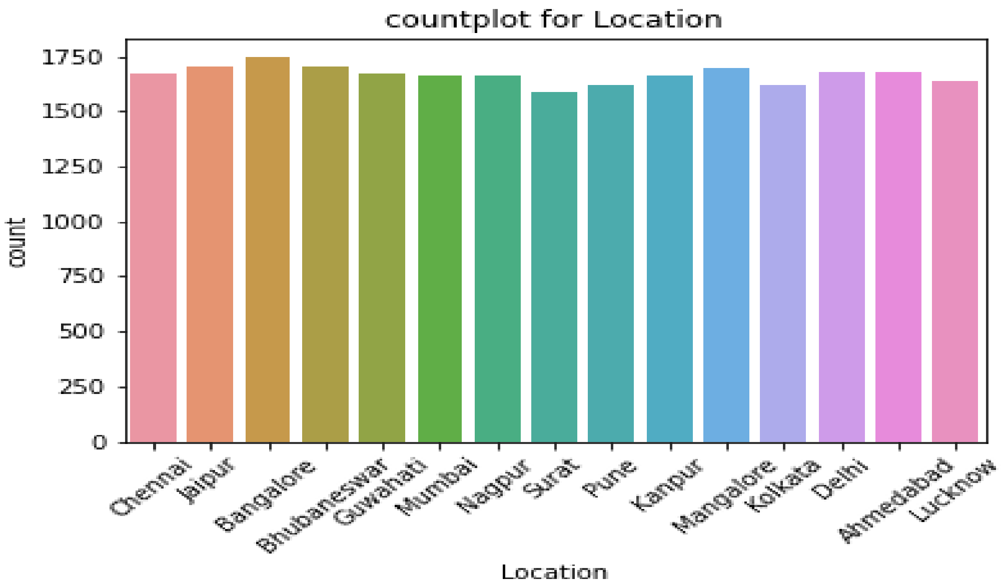


Image14.Countplot_location

By looking into the above plot we can say that health insurance are taken equally for all the hospital location.

Count plot of Alcohol:-

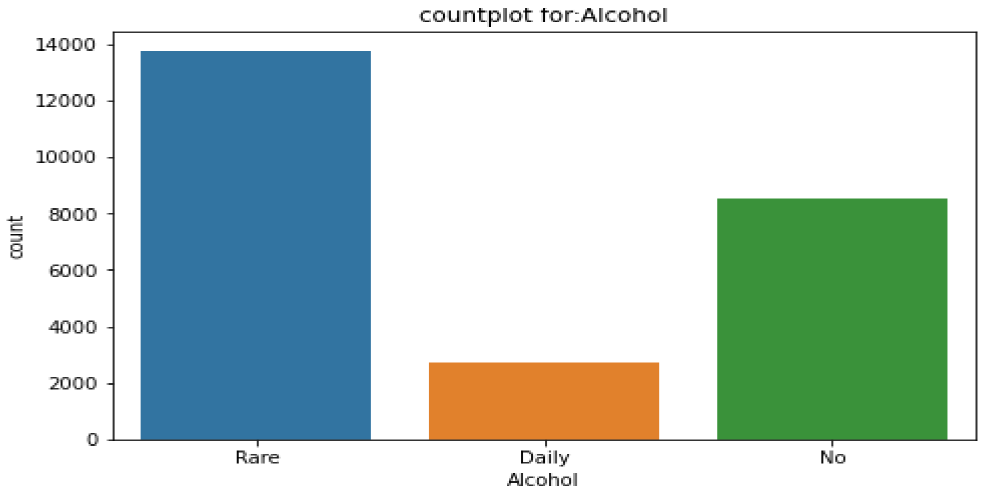


Image6.Countplot_Alcohol

From the above plot we can say that the customers whose alcohol consumption is rare have taken max health insurance. And customers whose alcohol consumption is daily have taken min health insurance.

Count plot of Exercise:-

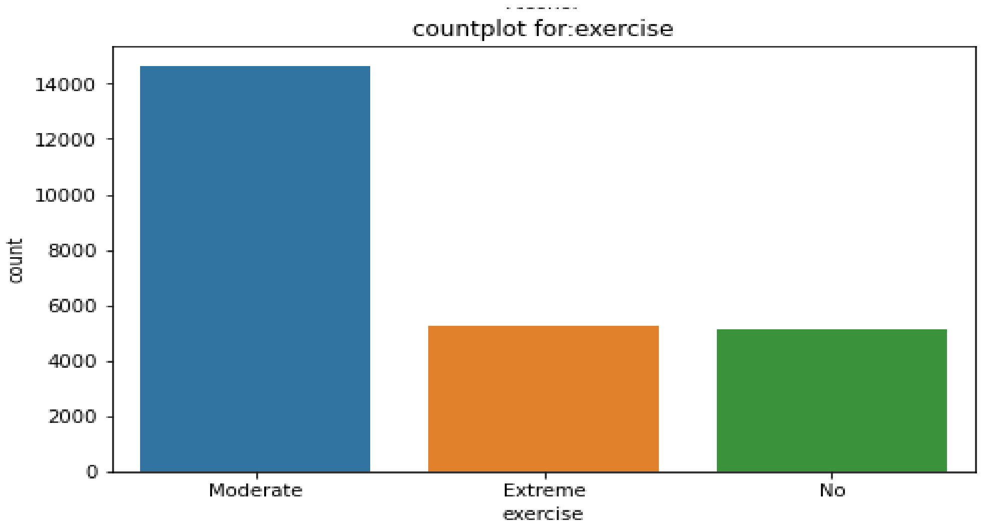


Image16.Countplot_exercise

From the above plot it can be infer that:-

- Customers whose exercise frequently have taken max health insurance rather than who never do exercise.

Count plot of covered_by_any_other_company:-

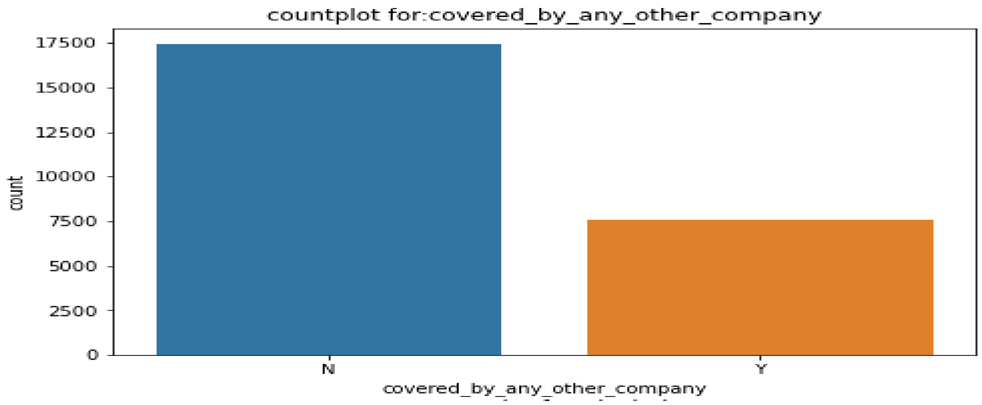


Image17.Countplot_covered_by_other_company

There are more number of customer who are not covered by any other insurance company.

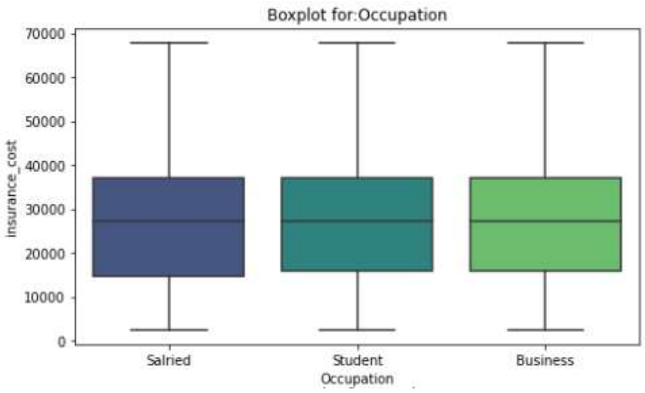
Bivariate analysis with Target variable:-

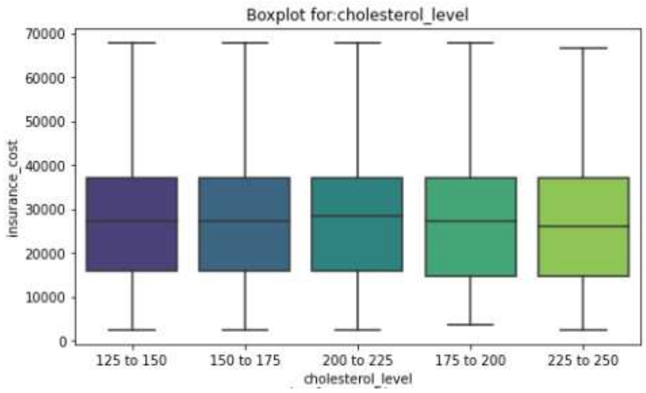
The term bivariate analysis refers to the analysis of two variables. The purpose of bivariate analysis is to understand the relationship between two variables.

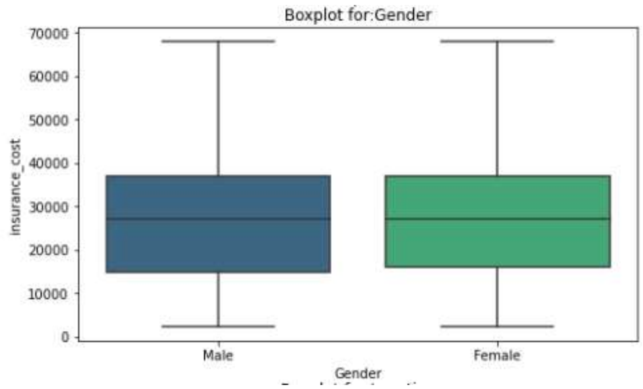
There are two common ways to perform bivariate analysis:

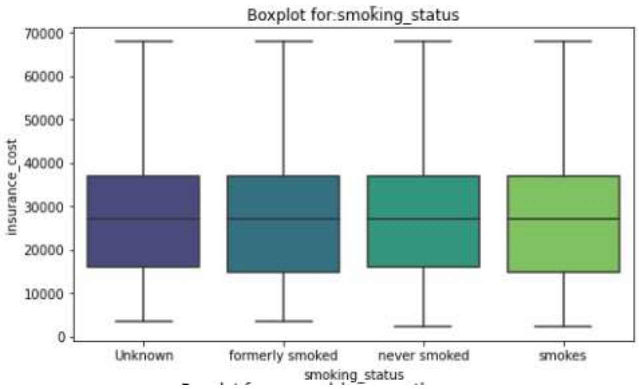
- Scatterplots (for continuous variable), Barplot (discrete numerical variable) and boxplot (categorical variables).
- Correlation Coefficients.

Dependent variable – Independent categorical variable:-









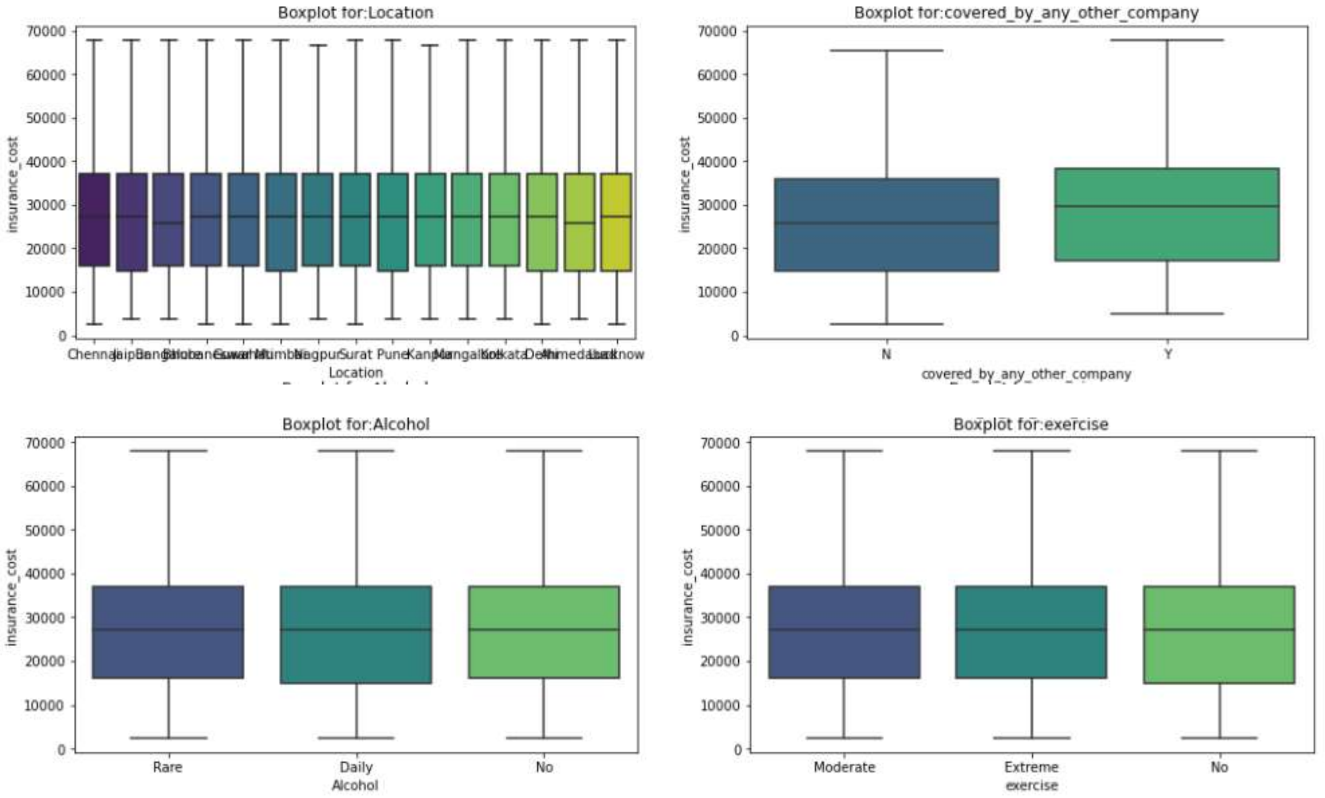


Image18.Boxplot of categorical variable

Key Observation:-

- If we see box plot of all variable vs insurance cost, the range and median of all unique values of all variable with respect to insurance_cost are somewhat same. They are right skewed and have outliers, which can be seen in the box plot and is clear in the density plot.
- Among the categorical variable, none of the variable is showing a strong predictor as compared together. Both the plots are suggesting the same.

Dependent variable – Independent Continuous variable

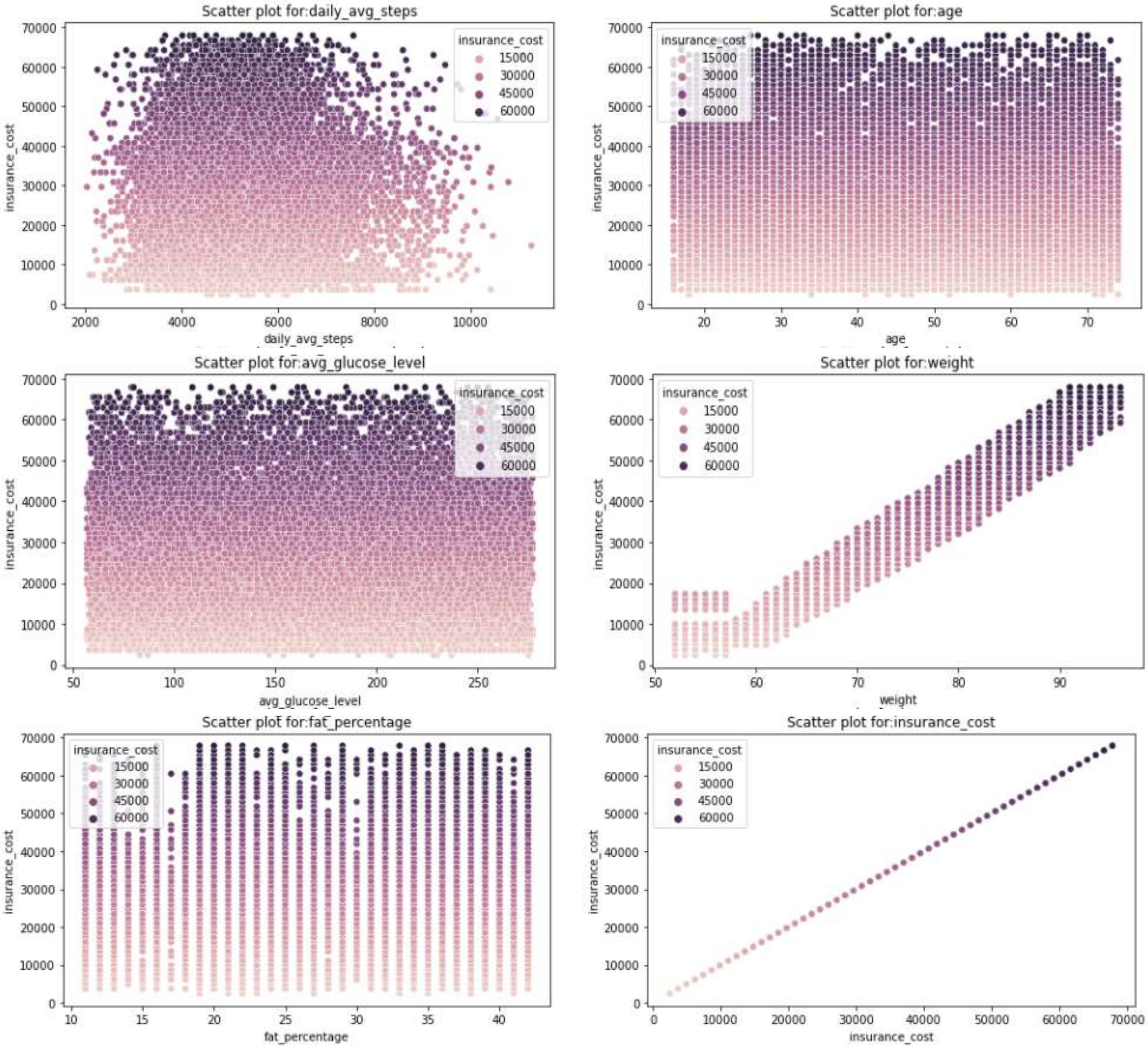
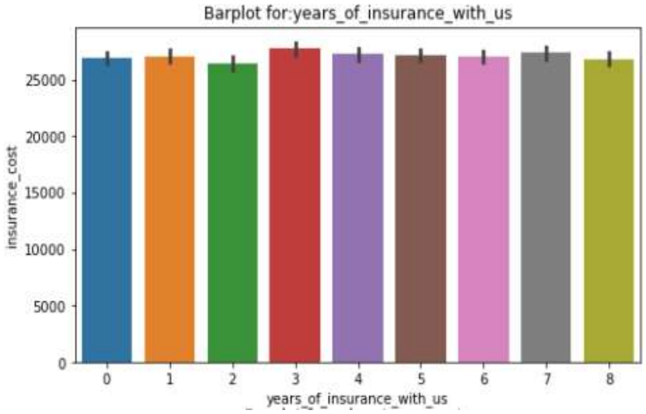


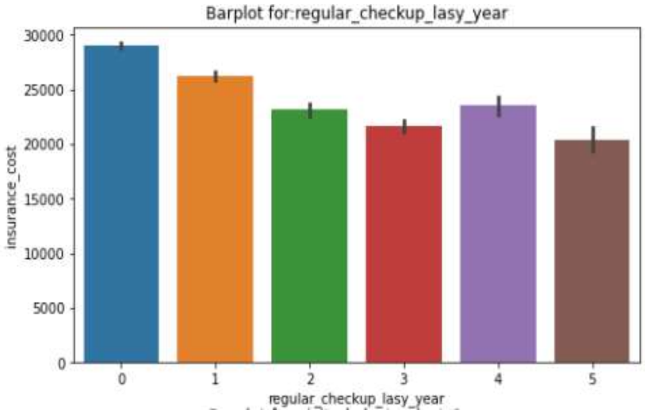
Image19.Scatterplot of continuous variable

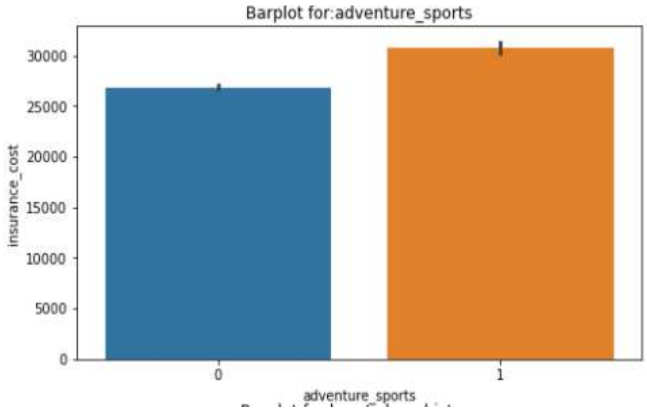
Key Observation:-

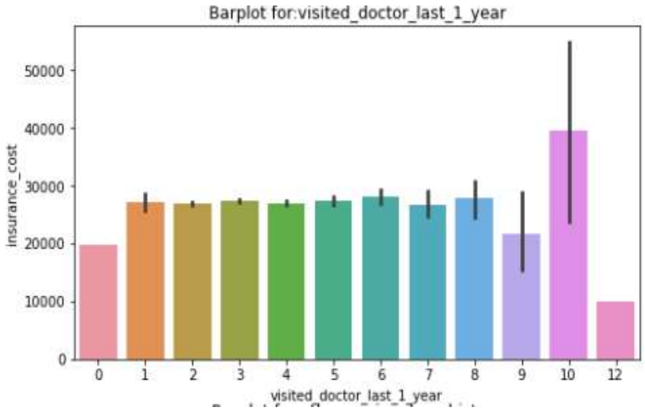
Only weight is forming a straight line when plotted, which means weight is showing a positive correlation. We can also say that it is a strong predictor of insurance cost.

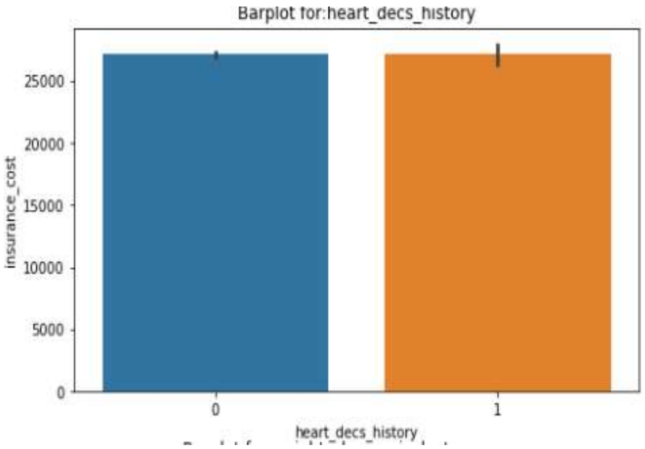
Dependent – Independent Continuous discrete variable

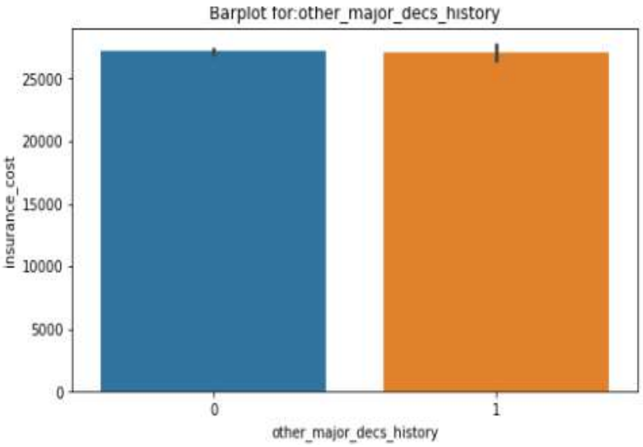

Barplot for:years_of_insurance_with_us
insurance_cost
0
5000
10000
15000
20000
25000
0
1
2
3
4
5
6
7
8
years_of_insurance_with_us


Barplot for:regular_checkup_lasy_year
insurance_cost
0
5000
10000
15000
20000
25000
30000
0
1
2
3
4
5
regular_checkup_lasy_year


Barplot for:adventure_sports
insurance_cost
0
5000
10000
15000
20000
25000
30000
0
1
adventure_sports


Barplot for:visited_doctor_last_1_year
insurance_cost
0
10000
20000
30000
40000
50000
0
1
2
3
4
5
6
7
8
9
10
12
visited_doctor_last_1_year


Barplot for:heart_decs_history
insurance_cost
0
5000
10000
15000
20000
25000
0
1
heart_decs_history


Barplot for:other_major_decs_history
insurance_cost
0
5000
10000
15000
20000
25000
0
1
other_major_decs_history

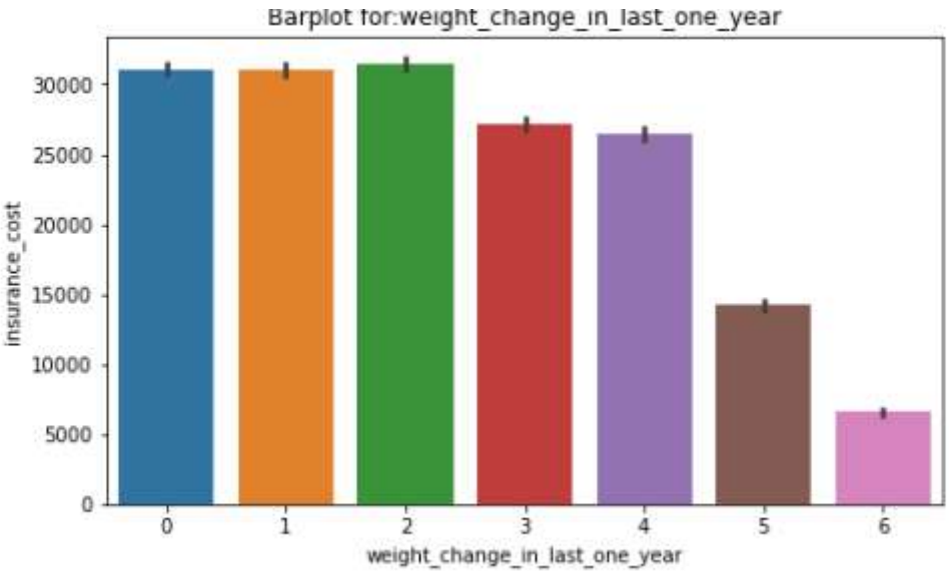


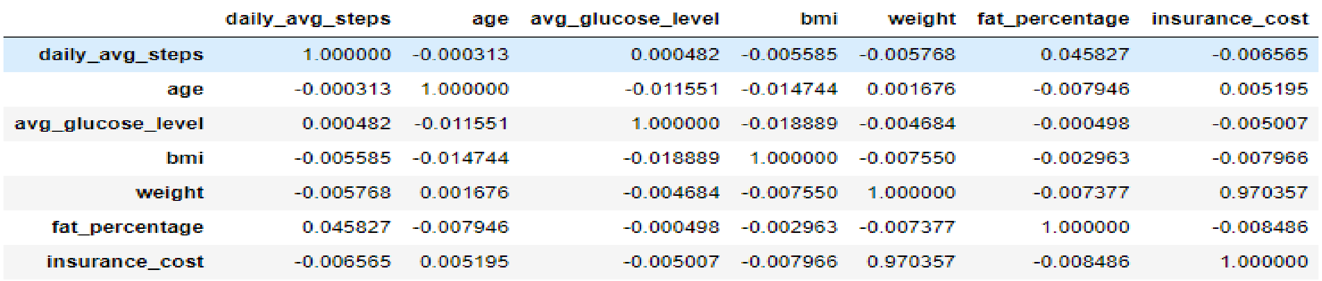
Image20.barplot of continuous categorical variable

Key Observation:-

- Years of insurance with us vs insurance cost- the customers who are insured with the company for more number of years. We can't see any variation in the plot. So it doesn't have correlation with insurance cost.
- Regular checkup lasy year vs insurance cost- the customers who haven't went for regular checkup have maximum health insurance cost. And customers who went for regular checkup 5 times have less insurance. Which means it has strong correlation with insurance cost. And it is a strong predictor.
- Heart disease history and other disease vs insurance cost - The insurance cost of customer who have heart disease history and customers who don't have heart disease is almost same. We can't see any difference in both the category.
- Visited doctor last year vs insurance cost – There are almost same insurance cost of all customer who have visited doctor in last one year. But there is slight difference that customer who visited doctor 13 times have min health insurance cost. And the customer who visited doctor 10 times have highest insurance cost
- Weight change in last one year vs insurance cost – the customer whose weight change 2kg in last one year have max insurance cost which means it is a strong predictor of insurance cost.

Heatmap:

To check the correlation between dependent and independent variable using heat map



	daily_avg_steps	age	avg_glucose_level	bmi	weight	fat_percentage	insurance_cost
daily_avg_steps	1.000000	-0.000313	0.000482	-0.005585	-0.005768	0.045827	-0.006565
age	-0.000313	1.000000	-0.011551	-0.014744	0.001676	-0.007946	0.005195
avg_glucose_level	0.000482	-0.011551	1.000000	-0.018889	-0.004684	-0.000498	-0.005007
bmi	-0.005585	-0.014744	-0.018889	1.000000	-0.007550	-0.002963	-0.007966
weight	-0.005768	0.001676	-0.004684	-0.007550	1.000000	-0.007377	0.970357
fat_percentage	0.045827	-0.007946	-0.000498	-0.002963	-0.007377	1.000000	-0.008486
insurance_cost	-0.006565	0.005195	-0.005007	-0.007966	0.970357	-0.008486	1.000000

Table3.Correlation matrix

```
<AxesSubplot:>
```

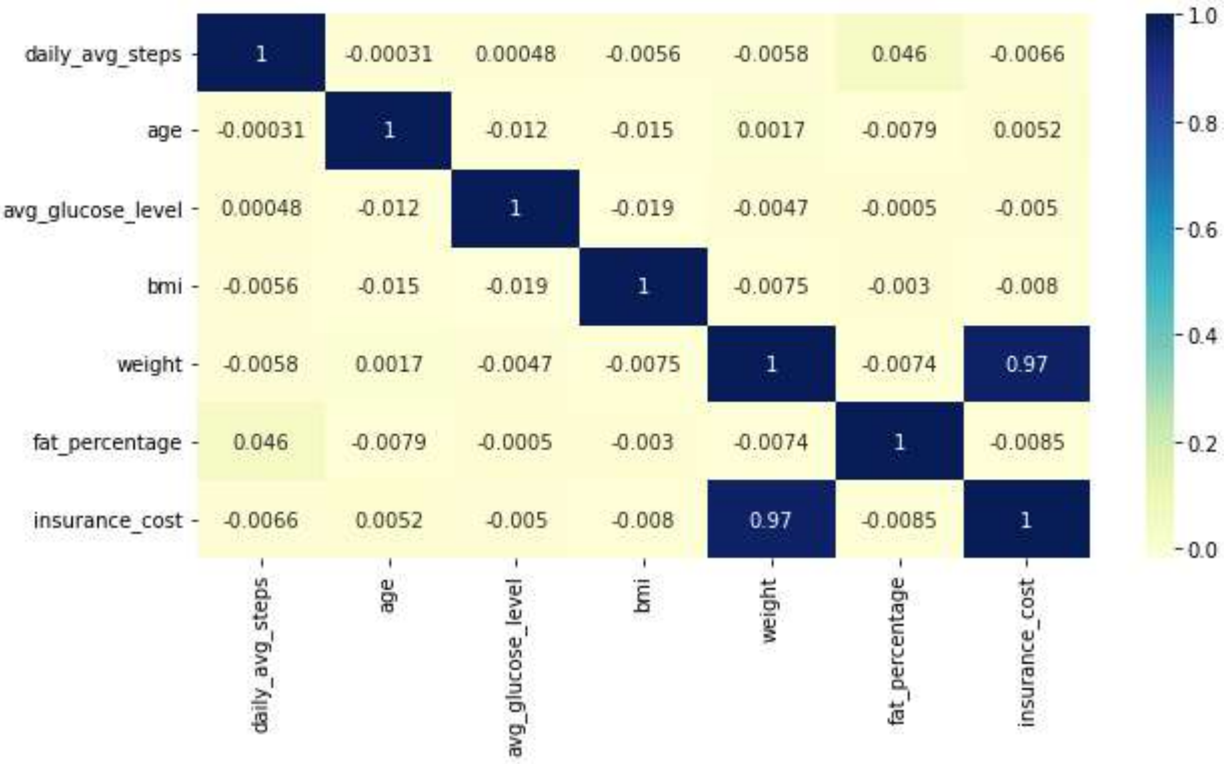


Image21.Heatmap

Observation:-

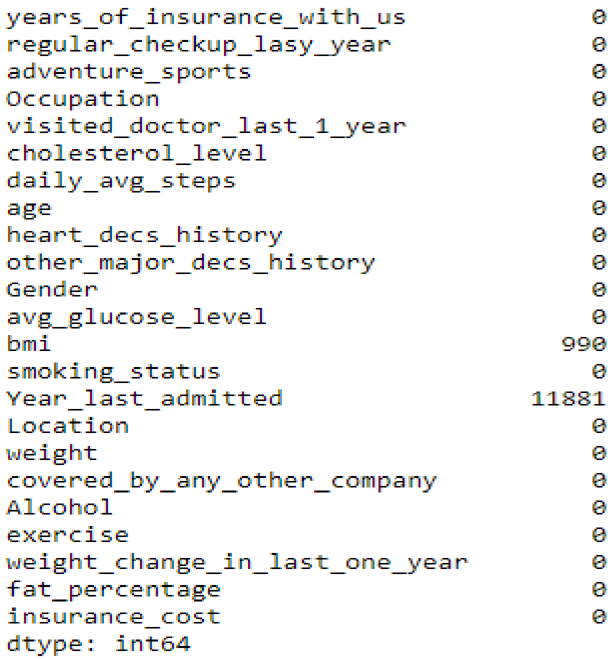
- There is high positive correlation between Insurance cost - our dependent variable and weight.

- Slightly negative correlation can be seen between weight change in last one year and insurance cost, weight change in last one year and weight.

3. Data Cleaning and Pre-processing

Missing Value treatment

Let's Check the missing values in the dataset:

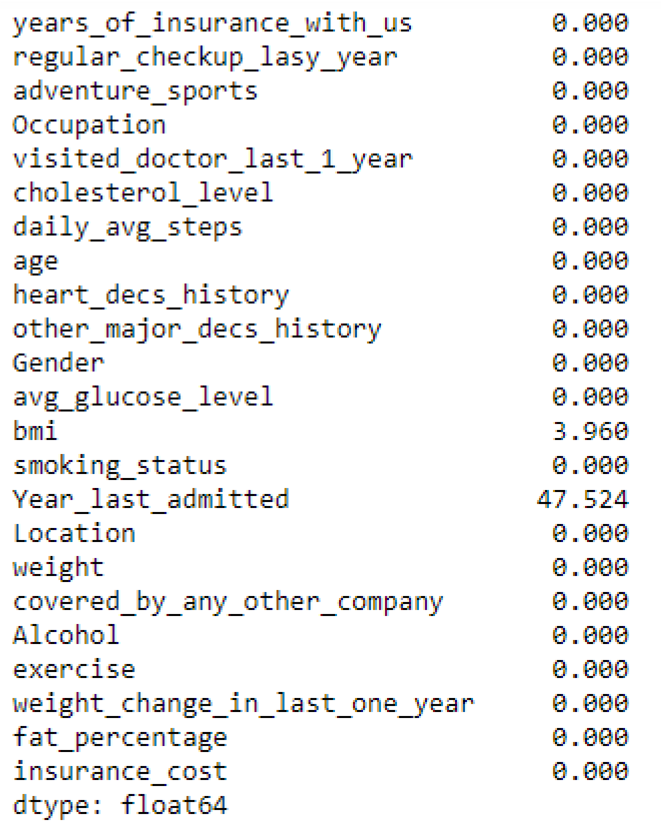


```
years_of_insurance_with_us           0
regular_checkup_lasy_year            0
adventure_sports                     0
Occupation                           0
visited_doctor_last_1_year           0
cholesterol_level                    0
daily_avg_steps                      0
age                                  0
heart_decs_history                   0
other_major_decs_history             0
Gender                               0
avg_glucose_level                    0
bmi                                990
smoking_status                       0
Year_last_admitted               11881
Location                             0
weight                               0
covered_by_any_other_company         0
Alcohol                              0
exercise                             0
weight_change_in_last_one_year       0
fat_percentage                       0
insurance_cost                       0
dtype: int64
```

Image 2.Missing Value

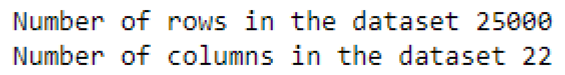
The dataset contains missing value. 990 values are missing in bmi column and 11881 values are missing in year_last_admitted column.

Let's calculate the percentage of missing value



```
years_of_insurance_with_us          0.000
regular_checkup_lasy_year           0.000
adventure_sports                    0.000
Occupation                          0.000
visited_doctor_last_1_year          0.000
cholesterol_level                   0.000
daily_avg_steps                     0.000
age                                 0.000
heart_decs_history                  0.000
other_major_decs_history            0.000
Gender                              0.000
avg_glucose_level                   0.000
bmi                                 3.960
smoking_status                      0.000
Year_last_admitted                 47.524
Location                            0.000
weight                              0.000
covered_by_any_other_company        0.000
Alcohol                             0.000
exercise                            0.000
weight_change_in_last_one_year      0.000
fat_percentage                      0.000
insurance_cost                      0.000
dtype: float64
```

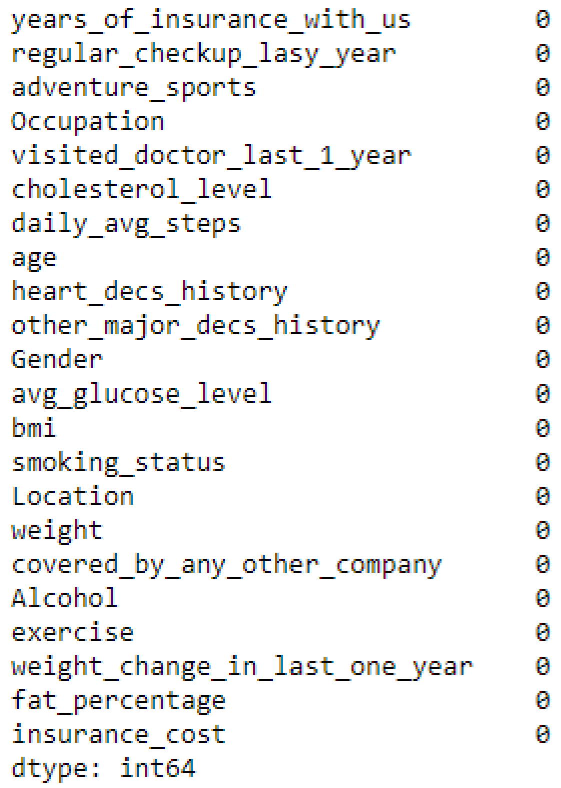
Year_last_admitted has more than 15-20% missing value (15% is used as a norm, imputation of more than 15% is not recommended) that's why we have removed Year_last_admitted column from the model building dataset.
After dropping the column let's check the shape of the dataset.



```
Number of rows in the dataset 25000
Number of columns in the dataset 22
```

Bmi column also has missing value. Since the variables have outliers, median is the best measure of central tendency to fill in missing values accordingly, the missing values have been treated using the SimpleImputer with strategy 'median'.

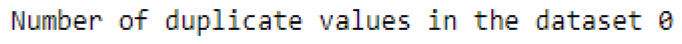
Let's check the missing value after imputing the missing value



```
years_of_insurance_with_us          0
regular_checkup_lasy_year           0
adventure_sports                    0
Occupation                          0
visited_doctor_last_1_year          0
cholesterol_level                   0
daily_avg_steps                     0
age                                 0
heart_decs_history                  0
other_major_decs_history            0
Gender                              0
avg_glucose_level                   0
bmi                                 0
smoking_status                      0
Location                            0
weight                              0
covered_by_any_other_company        0
Alcohol                             0
exercise                            0
weight_change_in_last_one_year      0
fat_percentage                      0
insurance_cost                      0
dtype: int64
```

Now there is no more missing values in the dataset.

Let’s Check the duplicate values:-



```
Number of duplicate values in the dataset 0
```

There is zero duplicate values in the dataset.

Outlier treatment –

Detecting outliers

Conventionally, outliers are identified based on the inter-quartile distance as follows:
Q1 – 25th Percentile
Q3 – 75th Percentile
IQR = Q3 – Q1

Lower outlier = Value < 1.5 * IQR
Upper Outlier = Value > 1.5 * IQR
Based on this definition, a view for each of the individual variables is as follows:

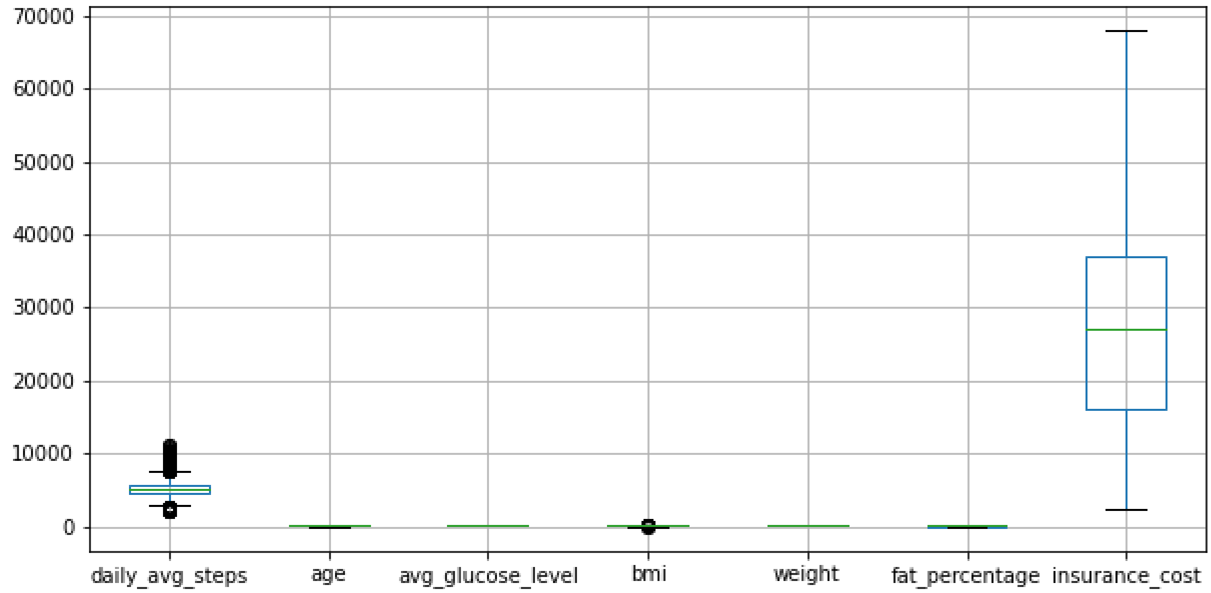


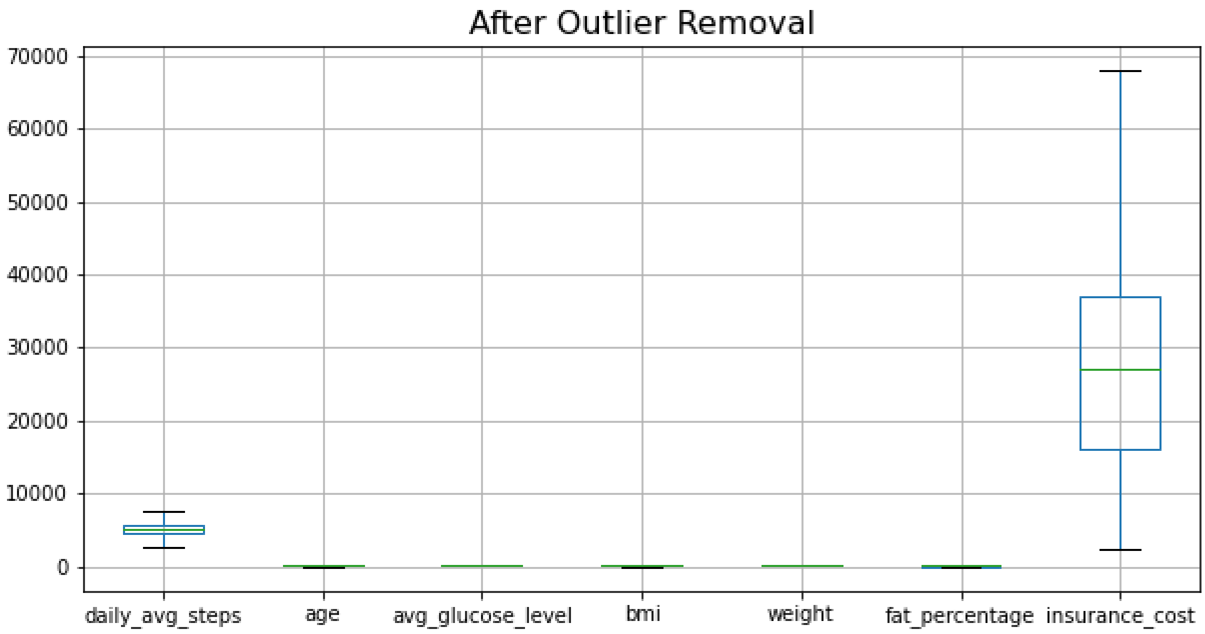
Image 22.Outliers boxplot

Observation:-

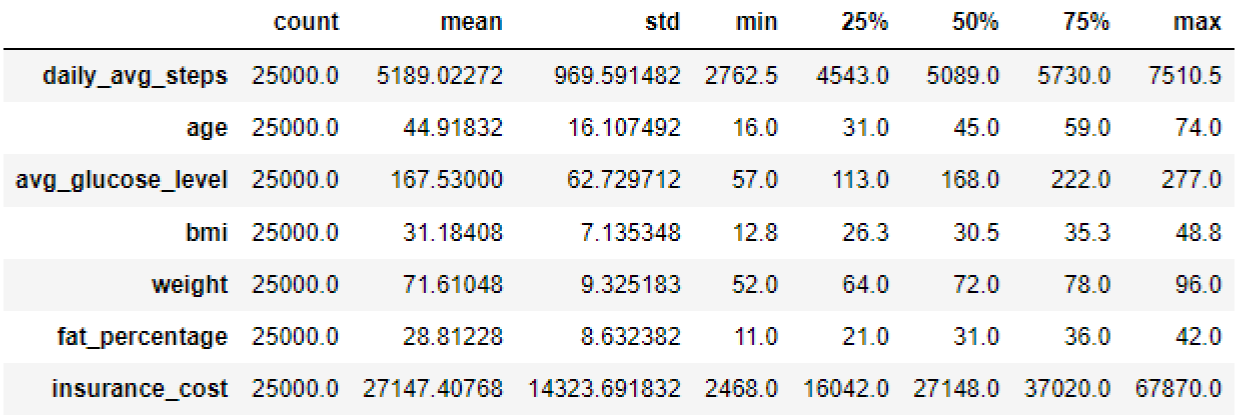
- Many variables have outliers as defined by the above criteria.
- If all the records with outliers are dropped, only few records remain. Modeling cannot be done with only these records, so, dropping records is not an option.

Rules:

- If Value < 1.5 * IQR, then replace it with lower_range
- If Value > 1.5 * IQR, then replace it with Upper_range.



We have treated all the outliers and as is to be expected, the median values of the original data and after outlier treatment do not change.



	count	mean	std	min	25%	50%	75%	max
daily_avg_steps	25000.0	5189.02272	969.591482	2762.5	4543.0	5089.0	5730.0	7510.5
age	25000.0	44.91832	16.107492	16.0	31.0	45.0	59.0	74.0
avg_glucose_level	25000.0	167.53000	62.729712	57.0	113.0	168.0	222.0	277.0
bmi	25000.0	31.18408	7.135348	12.8	26.3	30.5	35.3	48.8
weight	25000.0	71.61048	9.325183	52.0	64.0	72.0	78.0	96.0
fat_percentage	25000.0	28.81228	8.632382	11.0	21.0	31.0	36.0	42.0
insurance_cost	25000.0	27147.40768	14323.691832	2468.0	16042.0	27148.0	37020.0	67870.0

Encoding Categorical variable:-

Machine learning algorithm cannot work with categorical data and needs to be converted into numer ical data. The dataset contains categorical variable like Occupation, Gender, Smoking_status, Location , covered_by_any_other_company, Alcohol and Exercise. These variables have no specific order of pr eference and also since the data is string labels, machine learning models misinterpreted that there is some sort of hierarchy in them.

One approach to solve this problem can be label encoding where we will assign a numerical value to these labels for example Male and Female mapped to 0 and 1. But this can add bias in our model as it will start giving higher preference to the Female parameter as 1>0 and ideally both labels are equally important in the dataset.

To deal with this issue we will use One Hot Encoding technique.

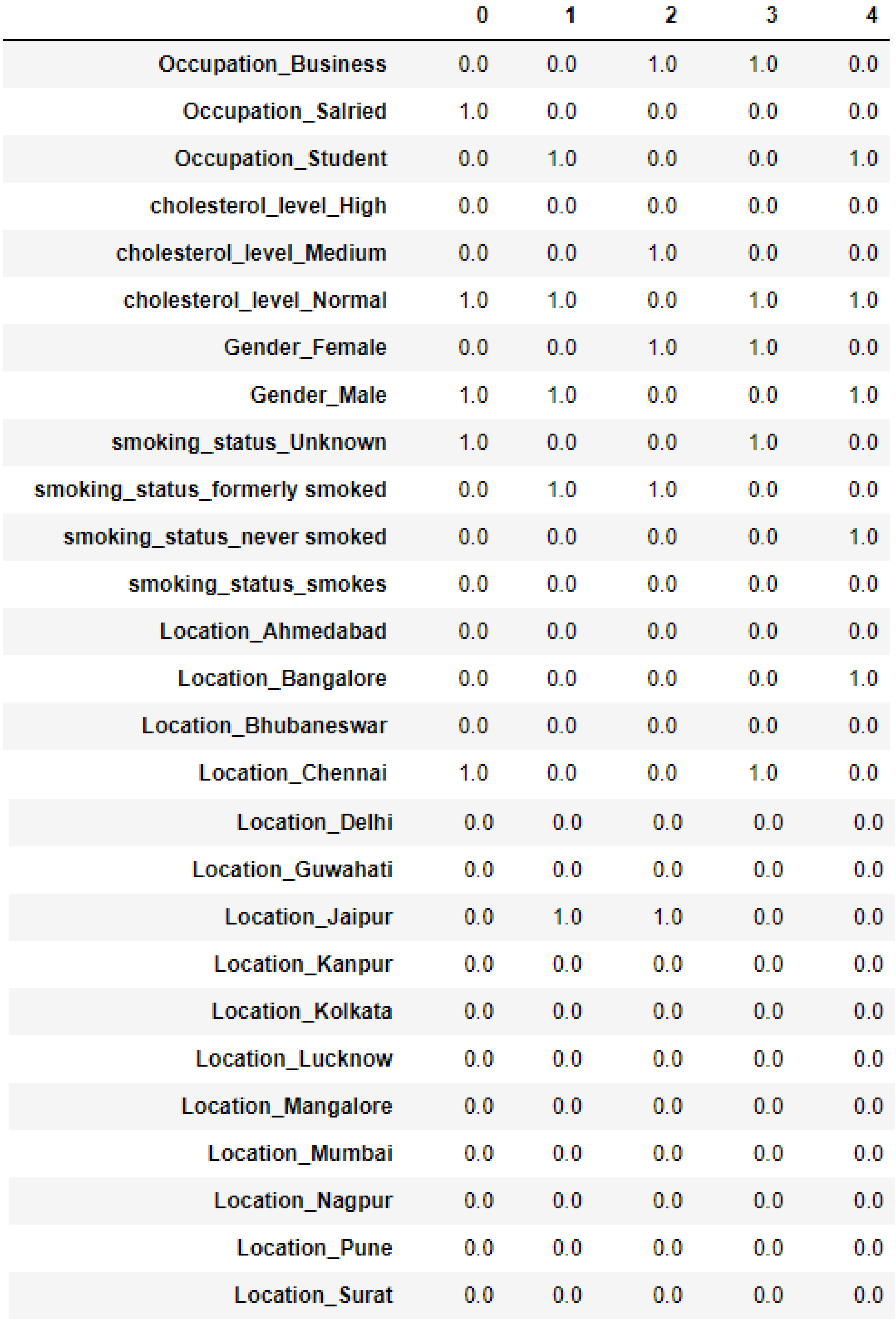
One hot encoding is a technique used to represent categorical variables as numerical values in a machine learning model. The advantages of using one hot encoding include:

1. It allows the use of categorical variables in models that require numerical input.
2. It can improve model performance by providing more information to the model about the categorical variable.
3. It can help to avoid the problem of ordinality, which can occur when a categorical variable has a natural ordering (e.g. “small”, “medium”, “large”).

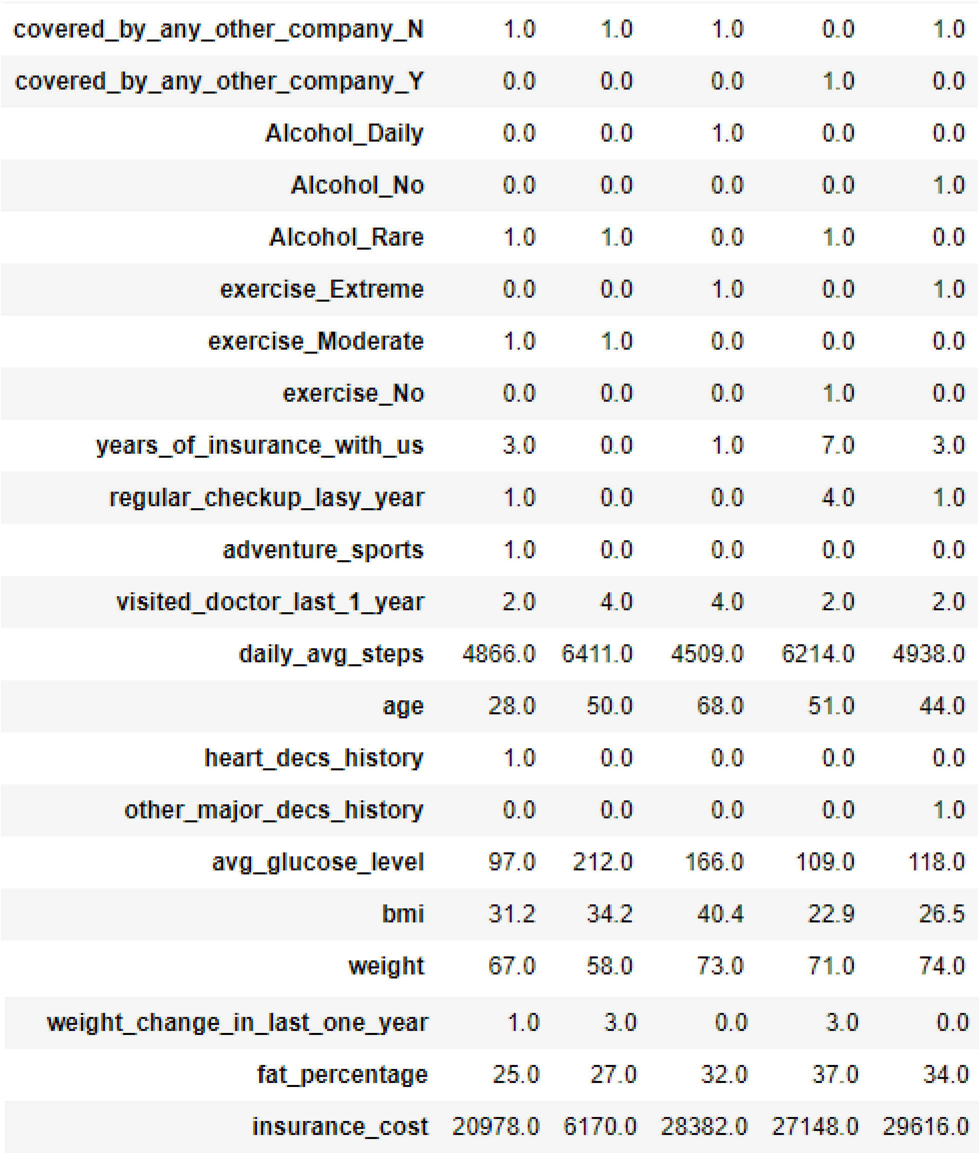
We have converted the Cholestrol_level into different category:-
Normal: <200
Medium:- 200 to 225
High:- 225 to 250

And then performed the one hot encoding to all the categorical variable to convert it into numerical value

The below table shows the dataset of after performing one hot encoding:-



	0	1	2	3	4
Occupation_Business	0.0	0.0	1.0	1.0	0.0
Occupation_Salried	1.0	0.0	0.0	0.0	0.0
Occupation_Student	0.0	1.0	0.0	0.0	1.0
cholesterol_level_High	0.0	0.0	0.0	0.0	0.0
cholesterol_level_Medium	0.0	0.0	1.0	0.0	0.0
cholesterol_level_Normal	1.0	1.0	0.0	1.0	1.0
Gender_Female	0.0	0.0	1.0	1.0	0.0
Gender_Male	1.0	1.0	0.0	0.0	1.0
smoking_status_Unknown	1.0	0.0	0.0	1.0	0.0
smoking_status_formerly smoked	0.0	1.0	1.0	0.0	0.0
smoking_status_never smoked	0.0	0.0	0.0	0.0	1.0
smoking_status_smokes	0.0	0.0	0.0	0.0	0.0
Location_Ahmedabad	0.0	0.0	0.0	0.0	0.0
Location_Bangalore	0.0	0.0	0.0	0.0	1.0
Location_Bhubaneswar	0.0	0.0	0.0	0.0	0.0
Location_Chennai	1.0	0.0	0.0	1.0	0.0
Location_Delhi	0.0	0.0	0.0	0.0	0.0
Location_Guwahati	0.0	0.0	0.0	0.0	0.0
Location_Jaipur	0.0	1.0	1.0	0.0	0.0
Location_Kanpur	0.0	0.0	0.0	0.0	0.0
Location_Kolkata	0.0	0.0	0.0	0.0	0.0
Location_Lucknow	0.0	0.0	0.0	0.0	0.0
Location_Mangalore	0.0	0.0	0.0	0.0	0.0
Location_Mumbai	0.0	0.0	0.0	0.0	0.0
Location_Nagpur	0.0	0.0	0.0	0.0	0.0
Location_Pune	0.0	0.0	0.0	0.0	0.0
Location_Surat	0.0	0.0	0.0	0.0	0.0

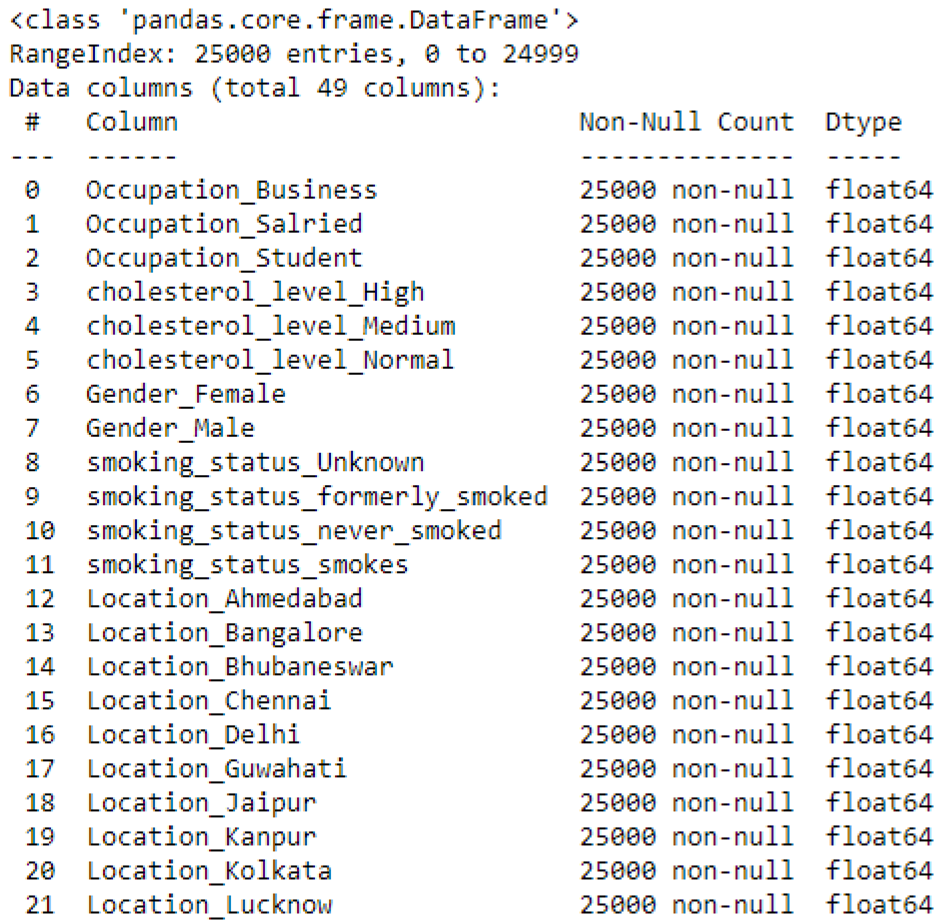


covered_by_any_other_company_N	1.0	1.0	1.0	0.0	1.0
covered_by_any_other_company_Y	0.0	0.0	0.0	1.0	0.0
Alcohol_Daily	0.0	0.0	1.0	0.0	0.0
Alcohol_No	0.0	0.0	0.0	0.0	1.0
Alcohol_Rare	1.0	1.0	0.0	1.0	0.0
exercise_Extreme	0.0	0.0	1.0	0.0	1.0
exercise_Moderate	1.0	1.0	0.0	0.0	0.0
exercise_No	0.0	0.0	0.0	1.0	0.0
years_of_insurance_with_us	3.0	0.0	1.0	7.0	3.0
regular_checkup_lasy_year	1.0	0.0	0.0	4.0	1.0
adventure_sports	1.0	0.0	0.0	0.0	0.0
visited_doctor_last_1_year	2.0	4.0	4.0	2.0	2.0
daily_avg_steps	4866.0	6411.0	4509.0	6214.0	4938.0
age	28.0	50.0	68.0	51.0	44.0
heart_decs_history	1.0	0.0	0.0	0.0	0.0
other_major_decs_history	0.0	0.0	0.0	0.0	1.0
avg_glucose_level	97.0	212.0	166.0	109.0	118.0
bmi	31.2	34.2	40.4	22.9	26.5
weight	67.0	58.0	73.0	71.0	74.0
weight_change_in_last_one_year	1.0	3.0	0.0	3.0	0.0
fat_percentage	25.0	27.0	32.0	37.0	34.0
insurance_cost	20978.0	6170.0	28382.0	27148.0	29616.0

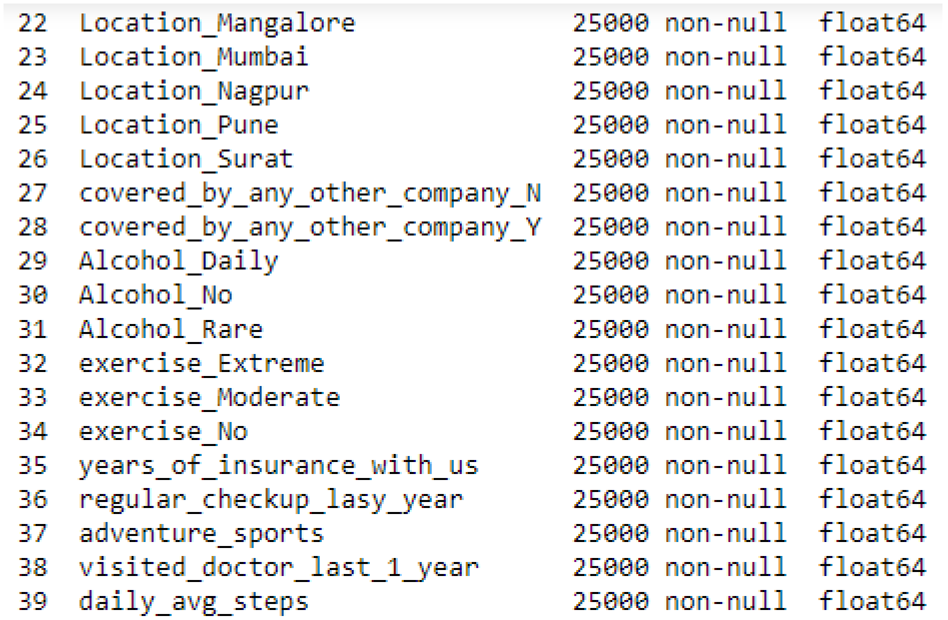
Image23.Encoded_dataset

Lets convert the datatype of the continous descrete variable from category to float64 as Machine learning algorithm cannot work with categorical data

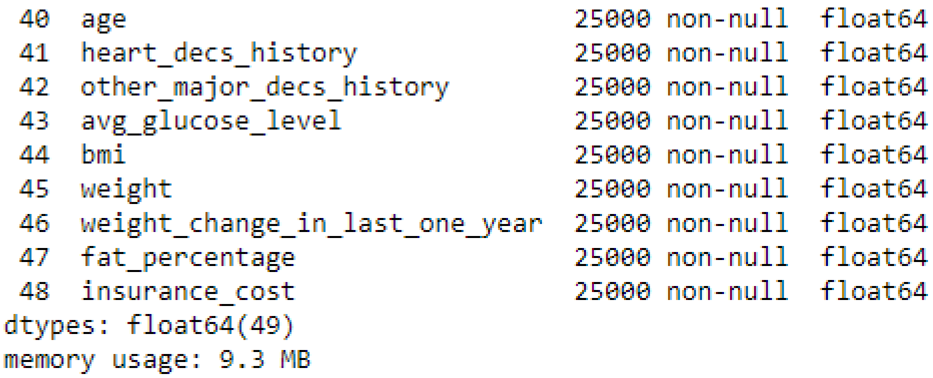
The info of all the variables:-



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 49 columns):
 #   Column                          Non-Null Count  Dtype  
---  ------                          --------------  -----  
 0   Occupation_Business             25000 non-null  float64
 1   Occupation_Salried              25000 non-null  float64
 2   Occupation_Student              25000 non-null  float64
 3   cholesterol_level_High          25000 non-null  float64
 4   cholesterol_level_Medium        25000 non-null  float64
 5   cholesterol_level_Normal        25000 non-null  float64
 6   Gender_Female                   25000 non-null  float64
 7   Gender_Male                     25000 non-null  float64
 8   smoking_status_Unknown          25000 non-null  float64
 9   smoking_status_formerly_smoked  25000 non-null  float64
 10  smoking_status_never_smoked     25000 non-null  float64
 11  smoking_status_smokes           25000 non-null  float64
 12  Location_Ahmedabad              25000 non-null  float64
 13  Location_Bangalore              25000 non-null  float64
 14  Location_Bhubaneswar            25000 non-null  float64
 15  Location_Chennai                25000 non-null  float64
 16  Location_Delhi                  25000 non-null  float64
 17  Location_Guwahati               25000 non-null  float64
 18  Location_Jaipur                 25000 non-null  float64
 19  Location_Kanpur                 25000 non-null  float64
 20  Location_Kolkata                25000 non-null  float64
 21  Location_Lucknow                25000 non-null  float64
```

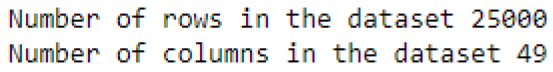


```
22  Location_Mangalore              25000 non-null  float64
23  Location_Mumbai                 25000 non-null  float64
24  Location_Nagpur                 25000 non-null  float64
25  Location_Pune                   25000 non-null  float64
26  Location_Surat                  25000 non-null  float64
27  covered_by_any_other_company_N  25000 non-null  float64
28  covered_by_any_other_company_Y  25000 non-null  float64
29  Alcohol_Daily                   25000 non-null  float64
30  Alcohol_No                      25000 non-null  float64
31  Alcohol_Rare                    25000 non-null  float64
32  exercise_Extreme                25000 non-null  float64
33  exercise_Moderate               25000 non-null  float64
34  exercise_No                     25000 non-null  float64
35  years_of_insurance_with_us      25000 non-null  float64
36  regular_checkup_lasy_year       25000 non-null  float64
37  adventure_sports                25000 non-null  float64
38  visited_doctor_last_1_year      25000 non-null  float64
39  daily_avg_steps                 25000 non-null  float64
```



```
 40  age                              25000 non-null  float64
 41  heart_decs_history               25000 non-null  float64
 42  other_major_decs_history         25000 non-null  float64
 43  avg_glucose_level                25000 non-null  float64
 44  bmi                              25000 non-null  float64
 45  weight                           25000 non-null  float64
 46  weight_change_in_last_one_year   25000 non-null  float64
 47  fat_percentage                   25000 non-null  float64
 48  insurance_cost                   25000 non-null  float64
dtypes: float64(49)
memory usage: 9.3 MB
```

Shape of the dataset:-



```
 Number of rows in the dataset 25000
 Number of columns in the dataset 49
```

Data Imbalancing:-

- Checking data balance is very critical for Classification problems.
- In case of Regression, if a variable is significant then there should be adequate rows for all values of that variable.
- It's applicable to regression problem where observations are less in a particular segment, class etc. for which one cannot do regression.
- Data imbalance is of two types:
    1. Under-representation of a class in one or more independent variable.
    2. Under-representation of one class in the dependent variable.
- Many machine-learning techniques such as neural networks, make more reliable predictions from being trained with balanced data.
- Certain analytical method, however, notably linear regression and logistic regression do not benefit from balancing approach.
- Our problem is a linear regression problem so we will not work on balancing the data.
- When it comes to data approach for imbalanced regression we have two techniques that were heavily inspired on imbalanced regression data.
    3. SMOTER
    4. SMOGN

Scaling:-

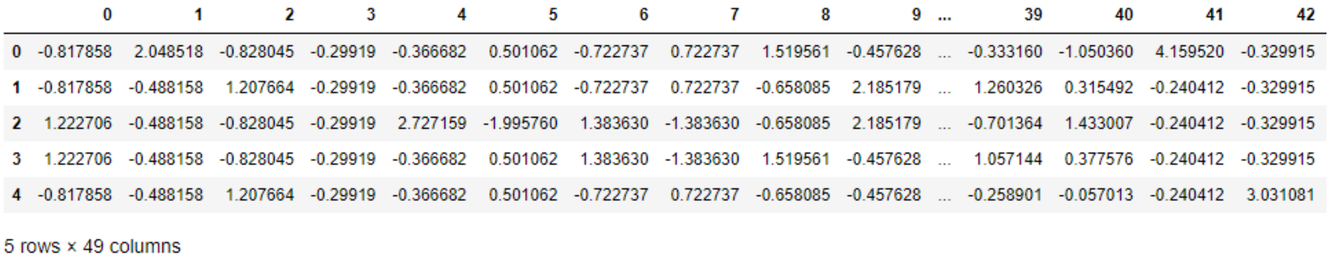
Scaling is done so that the data which belongs to different range can be bought together in similar range. Generally we perform feature scaling while dealing with Gradient Descent Based algorithm such as Linear and logistic regression as these are very sensitive to the range of data points. It is also useful in checking and reducing the multicollinearity in the data.

So it totally depends on the model we are building whether scaling is required or not. Usually the distance based model uses Euclidean distance between two data points in their computation.

Feature Scaling:-

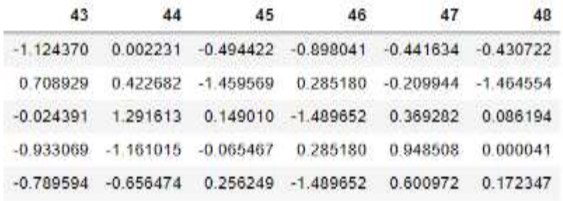
- For the given dataset scaling is required as all the variables are in different units.
- If feature scaling is not done, then a machine learning algorithm tends to weigh greater values, higher and consider smaller values as the lower values, regardless of the unit of the values
- Feature scaling is an important step while training a model. Often, the data which we receive in real world is on a different scale.
- For example: if we can have a dataset that has a column say avg_step and age (in years). Here, age can have values <100 years and avg_steps can have any values say 1000-5000. If we train a model based on this data, the model will be highly biased towards and will give more importance to it.
- To overcome this, we generally scale the data. Some common scalars are MinMax saclar and Standard Scalar.

The following result shows the dataset after scaling the dataset using standard scaler technique



	0	1	2	3	4	5	6	7	8	9	...	39	40	41	42
0	-0.817858	2.048518	-0.828045	-0.29919	-0.366682	0.501062	-0.722737	0.722737	1.519561	-0.457628	...	-0.333160	-1.050360	4.159520	-0.329915
1	-0.817858	-0.488158	1.207664	-0.29919	-0.366682	0.501062	-0.722737	0.722737	-0.658085	2.185179	...	1.260326	0.315492	-0.240412	-0.329915
2	1.222706	-0.488158	-0.828045	-0.29919	2.727159	-1.995760	1.383630	-1.383630	-0.658085	2.185179	...	-0.701364	1.433007	-0.240412	-0.329915
3	1.222706	-0.488158	-0.828045	-0.29919	-0.366682	0.501062	1.383630	-1.383630	1.519561	-0.457628	...	1.057144	0.377576	-0.240412	-0.329915
4	-0.817858	-0.488158	1.207664	-0.29919	-0.366682	0.501062	-0.722737	0.722737	-0.658085	-0.457628	...	-0.258901	-0.057013	-0.240412	3.031081

5 rows × 49 columns



43	44	45	46	47	48
-1.124370	0.002231	-0.494422	-0.898041	-0.441634	-0.430722
0.708929	0.422682	-1.459569	0.285180	-0.209944	-1.464554
-0.024391	1.291613	0.149010	-1.489652	0.369282	0.086194
-0.933069	-1.161015	-0.065467	0.285180	0.948508	0.000041
-0.789594	-0.656474	0.256249	-1.489652	0.600972	0.172347

Table4.Scaled_dataset

Clustering:-

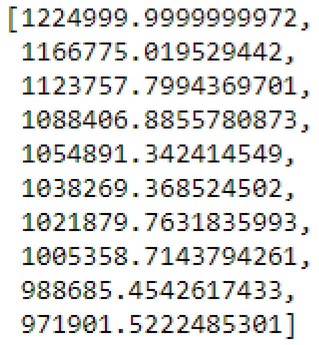
Clustering is the task of dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group and dissimilar to the data points in other groups.

Here we create cluster to group/segment the data into high, low & medium insurance cost.

I have used K-means clustering technique to form a cluster. To decide the number of clusters that are formed in the table used the values of inertia with k=1,2,3,4,5,6,7,8,9,10 and formed a pointplot.

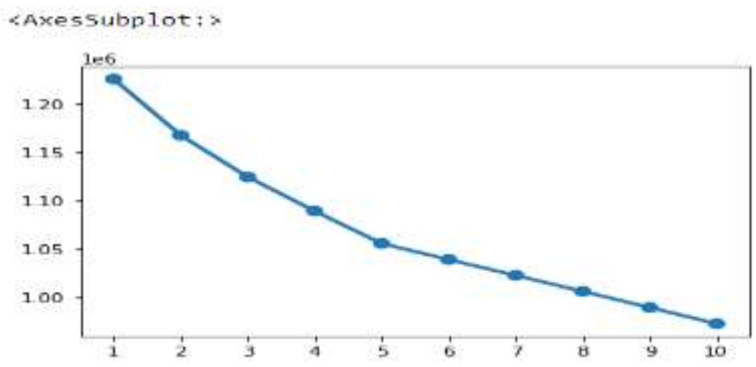
After that used the silhouette score to evaluate the number of clusters.

K-means inertia



```
[1224999.9999999972,
 1166775.019529442,
 1123757.7994369701,
 1088406.8855780873,
 1054891.342414549,
 1038269.368524502,
 1021879.7631835993,
 1005358.7143794261,
 988685.4542617433,
 971901.5222485301]
```

Point plot

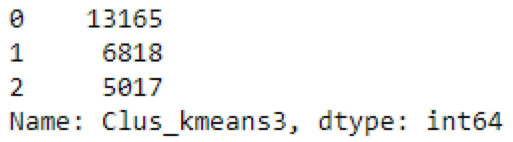


```
<AxesSubplot:>
```

Image25.pointplot

After looking into the silhouette score decided to form a cluster with k=3.

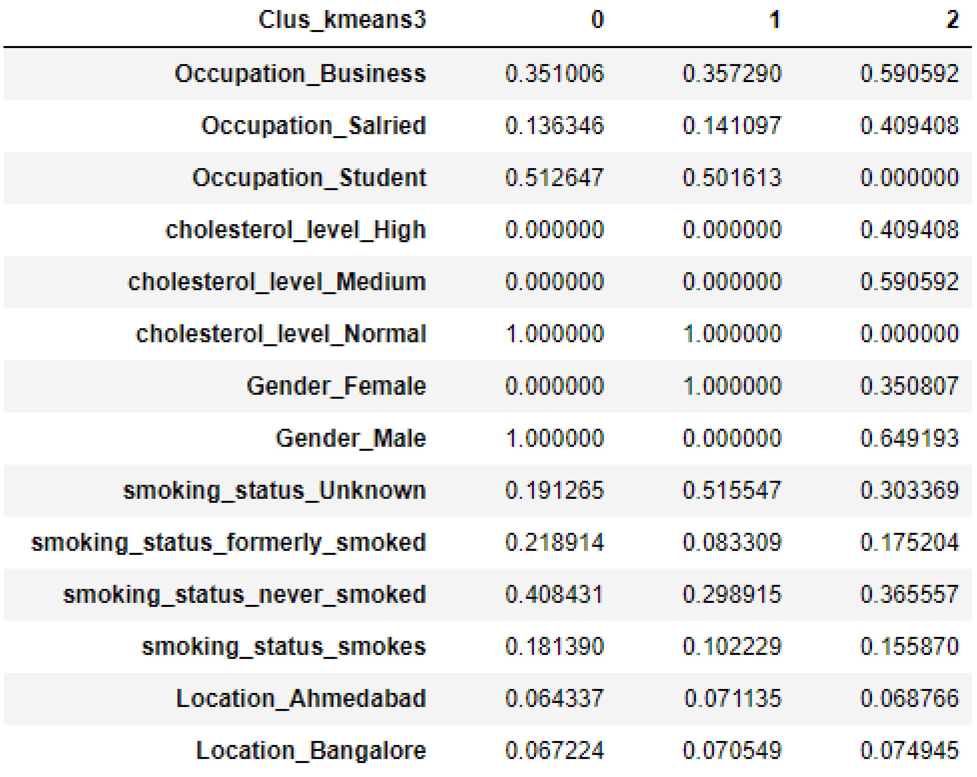
The cluster formed:



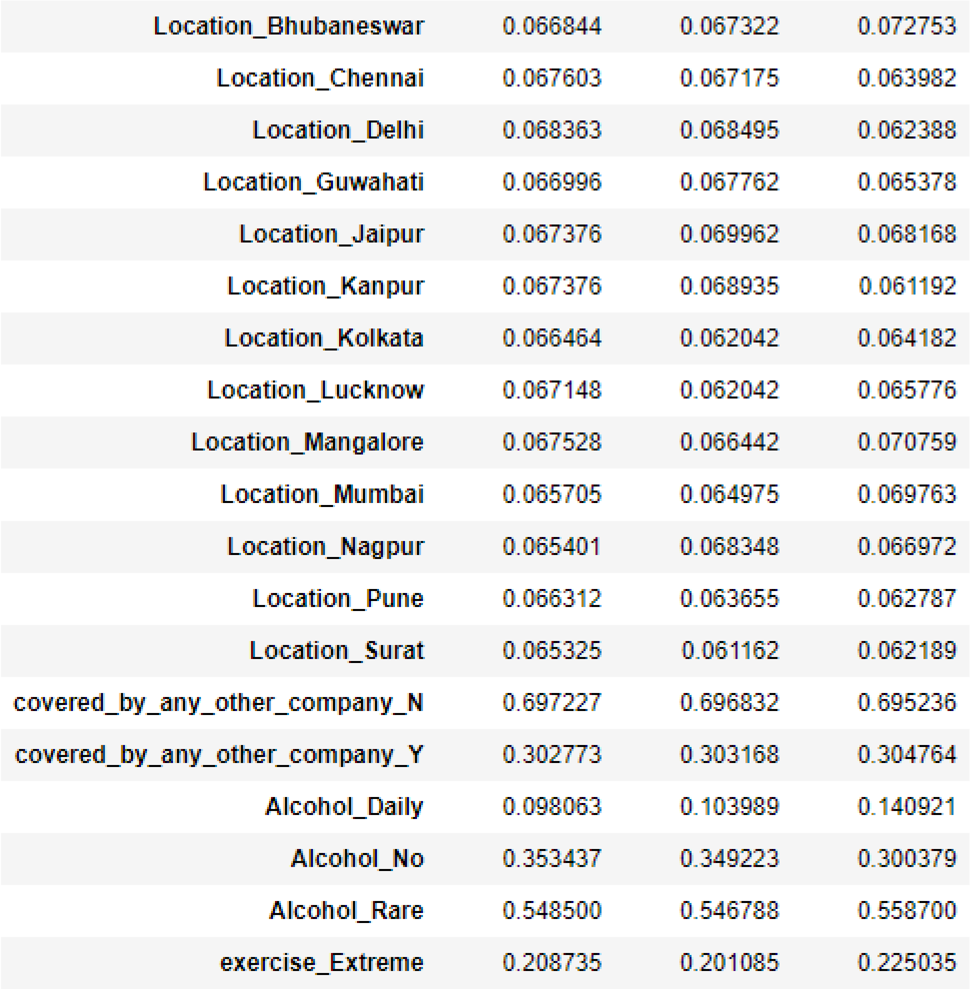
```
0    13165
1     6818
2     5017
Name: Clus_kmeans3, dtype: int64
```

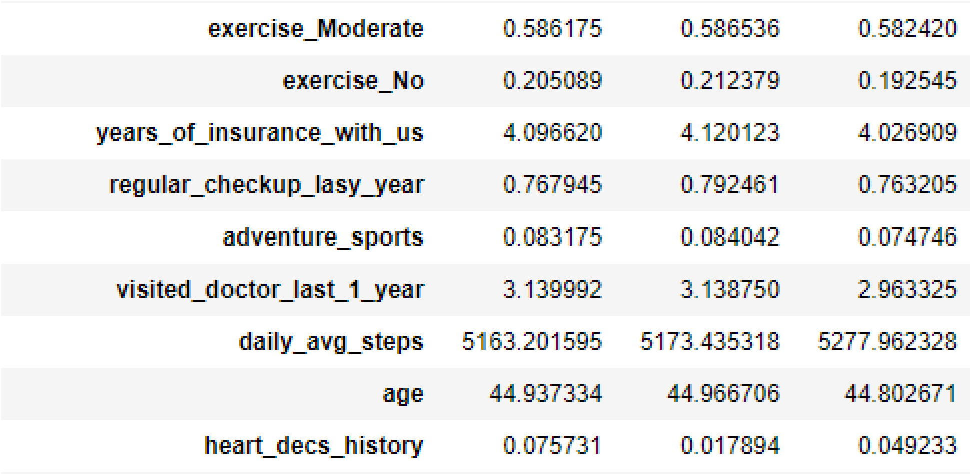
Three clusters formed with these frequencies.

The following table shows the result of the dataset after creating cluster which will help to decide the high, medium and low insurance cost.

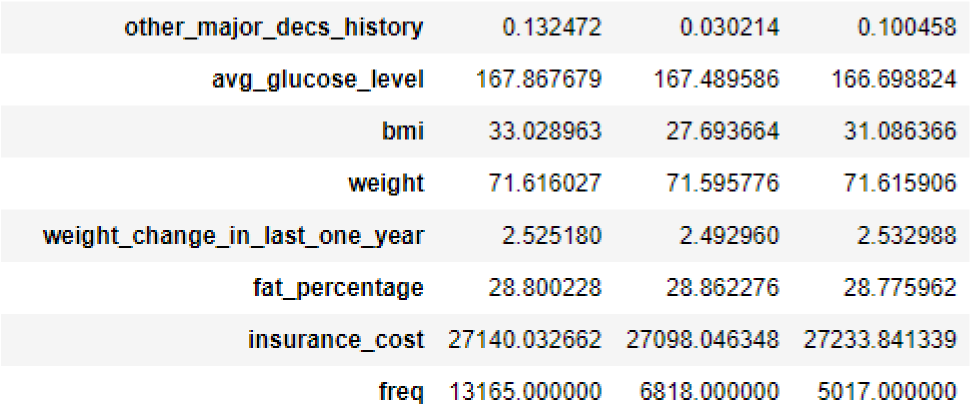


Clus_kmeans3	0	1	2
Occupation_Business	0.351006	0.357290	0.590592
Occupation_Salried	0.136346	0.141097	0.409408
Occupation_Student	0.512647	0.501613	0.000000
cholesterol_level_High	0.000000	0.000000	0.409408
cholesterol_level_Medium	0.000000	0.000000	0.590592
cholesterol_level_Normal	1.000000	1.000000	0.000000
Gender_Female	0.000000	1.000000	0.350807
Gender_Male	1.000000	0.000000	0.649193
smoking_status_Unknown	0.191265	0.515547	0.303369
smoking_status_formerly_smoked	0.218914	0.083309	0.175204
smoking_status_never_smoked	0.408431	0.298915	0.365557
smoking_status_smokes	0.181390	0.102229	0.155870
Location_Ahmedabad	0.064337	0.071135	0.068766
Location_Bangalore	0.067224	0.070549	0.074945





Location_Bhubaneswar	0.066844	0.067322	0.072753
Location_Chennai	0.067603	0.067175	0.063982
Location_Delhi	0.068363	0.068495	0.062388
Location_Guwahati	0.066996	0.067762	0.065378
Location_Jaipur	0.067376	0.069962	0.068168
Location_Kanpur	0.067376	0.068935	0.061192
Location_Kolkata	0.066464	0.062042	0.064182
Location_Lucknow	0.067148	0.062042	0.065776
Location_Mangalore	0.067528	0.066442	0.070759
Location_Mumbai	0.065705	0.064975	0.069763
Location_Nagpur	0.065401	0.068348	0.066972
Location_Pune	0.066312	0.063655	0.062787
Location_Surat	0.065325	0.061162	0.062189
covered_by_any_other_company_N	0.697227	0.696832	0.695236
covered_by_any_other_company_Y	0.302773	0.303168	0.304764
Alcohol_Daily	0.098063	0.103989	0.140921
Alcohol_No	0.353437	0.349223	0.300379
Alcohol_Rare	0.548500	0.546788	0.558700
exercise_Extreme	0.208735	0.201085	0.225035
exercise_Moderate	0.586175	0.586536	0.582420
exercise_No	0.205089	0.212379	0.192545
years_of_insurance_with_us	4.096620	4.120123	4.026909
regular_checkup_lasy_year	0.767945	0.792461	0.763205
adventure_sports	0.083175	0.084042	0.074746
visited_doctor_last_1_year	3.139992	3.138750	2.963325
daily_avg_steps	5163.201595	5173.435318	5277.962328
age	44.937334	44.966706	44.802671
heart_decs_history	0.075731	0.017894	0.049233



other_major_decs_history	0.132472	0.030214	0.100458
avg_glucose_level	167.867679	167.489586	166.698824
bmi	33.028963	27.693664	31.086366
weight	71.616027	71.595776	71.615906
weight_change_in_last_one_year	2.525180	2.492960	2.532988
fat_percentage	28.800228	28.862276	28.775962
insurance_cost	27140.032662	27098.046348	27233.841339
freq	13165.000000	6818.000000	5017.000000

Table5.Clustered_dataset

From the above table shows all the clusters doesn't have much difference. But if talk about frequency of all clusters then high insurance cost cluster have minimum frequency than the others.

The medium insurance cost have maximum frequency compared to other clusters.

Cluster 0:- Medium insurance cost with high bmi, high avg_glucose_level and less daily avg steps.
Cluster 1:- Low insurance cost with less bmi, less other_major_decs_history, less weight and high age.
Cluster 2:- High insurance cost with less avg_glucose_level, high weight_change_in_last_one_year, less age, high daily_avg_steps, less visited doctor in last one year, less in adventure sports.

Recommendation from the clusters:-

- In cluster 0 the customers have high bmi high avg_glucose_level and less daily_avg_steps. So we should
- In cluster 2 the customers have high weight_change_in_last_one_year,

4. Model Building

Insurance Cost prediction is a linear regression problem (establishing a relationship between a dependent variable and one or more independent variables), since we have to do continuous price prediction instead of logical operator type solution.

Regression analysis includes several variations, such as linear, multiple linear, and nonlinear. The most common models are simple linear and multiple linear. Nonlinear regression analysis is commonly used for more complicated data sets in which the dependent and independent variables show a nonlinear relationship.

Types of model that we will use solve this regression problem:-

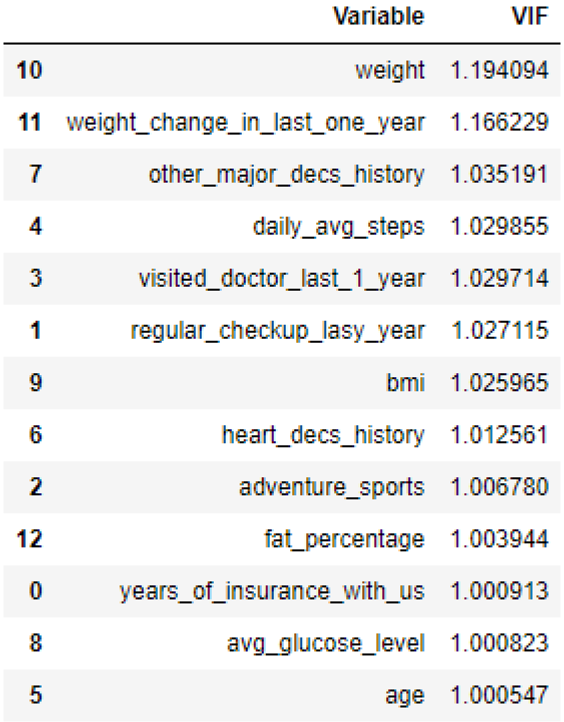
- **Linear Regression** – is one of the most common model for regression problems
  1. Simple Linear regression Model
  2. Multiple Linear regression Model
- **Lasso** - The model is penalized for the sum of absolute values of the weights. Introduces a new hyper parameter, alpha, the coefficient to penalize weights.
- **Ridge** - It takes a step further and penalizes the model for the sum of squared value of the weights.
- **Elastic net Elastic Net**- is a hybrid of Lasso and Ridge, where both the absolute value penalization and squared penalization are included, being regulated with another coefficient l1_ratio.
- **Decision Tree / Random Forest Regressor** - are one of the most commonly used, practical approaches for supervised learning. It can be used to solve both Regression and Classification.

Multicollinearity happens when independent variables in regression model are highly correlated to each other. So before building the model we will first check the multicollinearity of the independent variables.

Check Multicollinearity:

From the correlation matrix very few variables are correlated to each other. Not all collinearity problems can be detected by correlation matrix. This is because it shows correlation between pairs of variable. Sometimes collinearity exists between 3 or more variables which is known as multicollinearity. It makes it hard to interpret of model and also create an overfitting problem.

So we will check the multicollinearity by using Variance Inflation factor (VIF) for each independent variable



	Variable	VIF
10	weight	1.194094
11	weight_change_in_last_one_year	1.166229
7	other_major_decs_history	1.035191
4	daily_avg_steps	1.029855
3	visited_doctor_last_1_year	1.029714
1	regular_checkup_lasy_year	1.027115
9	bmi	1.025965
6	heart_decs_history	1.012561
2	adventure_sports	1.006780
12	fat_percentage	1.003944
0	years_of_insurance_with_us	1.000913
8	avg_glucose_level	1.000823
5	age	1.000547

Image25.VIF

We will not drop any variable as we see that the value of VIF is not high for any variables. Generally, we drop variables with VIF more than 5 (very high correlation) & build our model. But here all the VIF values are less than 5.

Now we will split the data into train and test data.

Train –Test Split:-

Let's break the X and y data frames into training set and test set in the ratio 75:25. For this we will use Sklearn package's data splitting function which is based on random function. We will split the dataset into training and testing data and perform the linear regression model on the dataset.

Feature importance:-

Feature (variable) importance indicates how much each feature contributes to the model prediction. Basically, it determines the degree of usefulness of a specific variable for a current model and prediction.

We represent feature importance using a numeric value that we call the score, where the higher the score value has, the more important it is.

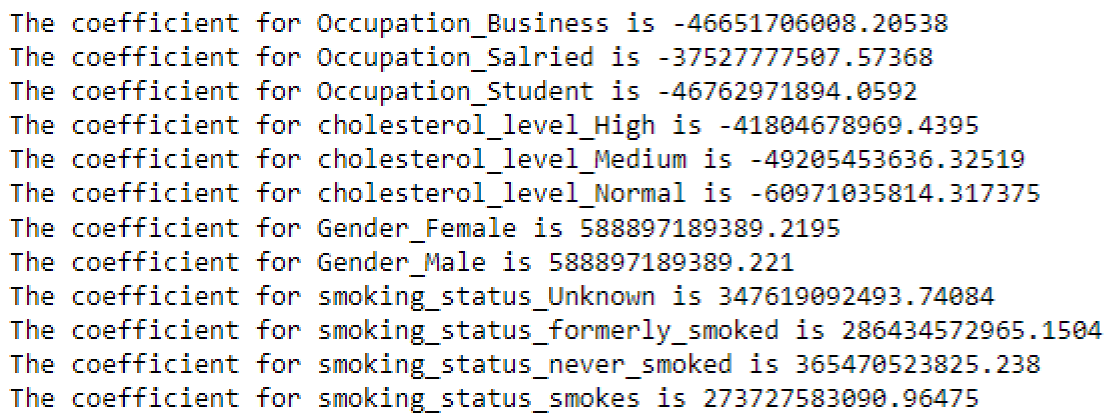
Model dependent feature importance.

1. **Linear Regression feature importance:-**
   We first fit a linear regression model and then extract coefficients that will show the importance of each input variable. Besides simple linear regression, a linear regression with an L1 regularization parameter, called Lasso regression, is commonly used, especially for feature selection. Lasso regression has regularization parameter that controls the degree of regularization and shrinks the coefficients to become smaller. Also, Lasso might arbitrarily shrink correlated variables by selecting only one of them

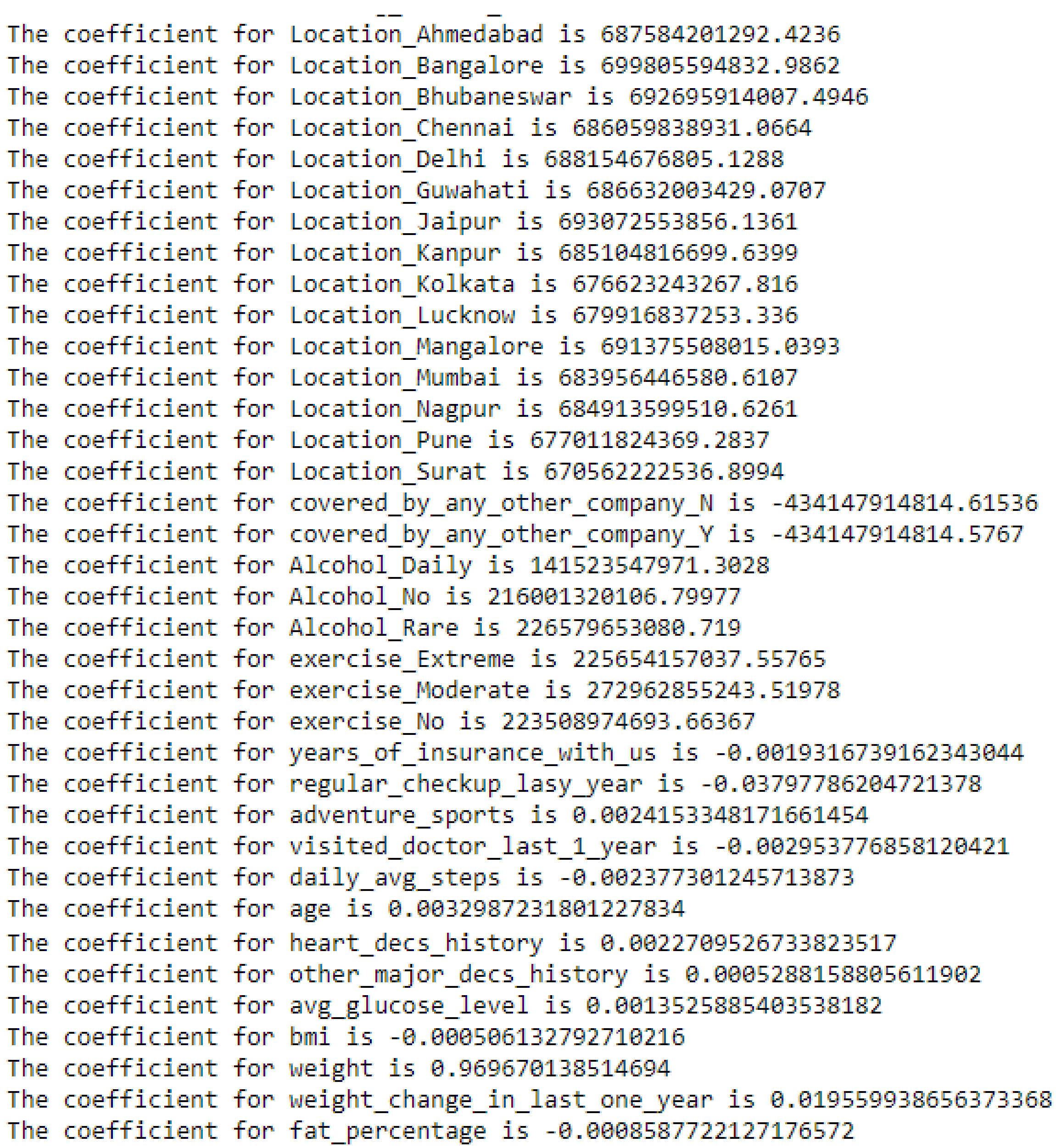
2. **Decision Tree / Random Forest Regressor (Nonlinear Model) feature importance:-**
   These Models provide feature importance scores based on reducing the criterion used to select split points. Usually, they are based on **Gini** or entropy impurity measurements. Also, the same approach can be used for all algorithms based on decision trees such as random forest and gradient boosting.

Model1:- Linear Regression

First fit the data and then explore the coefficients for each of the independent attributes.



```
The coefficient for Occupation_Business is -46651706008.20538
The coefficient for Occupation_Salried is -37527777507.57368
The coefficient for Occupation_Student is -46762971894.0592
The coefficient for cholesterol_level_High is -41804678969.4395
The coefficient for cholesterol_level_Medium is -49205453636.32519
The coefficient for cholesterol_level_Normal is -60971035814.317375
The coefficient for Gender_Female is 588897189389.2195
The coefficient for Gender_Male is 588897189389.221
The coefficient for smoking_status_Unknown is 347619092493.74084
The coefficient for smoking_status_formerly_smoked is 286434572965.1504
The coefficient for smoking_status_never_smoked is 365470523825.238
The coefficient for smoking_status_smokes is 273727583090.96475
```

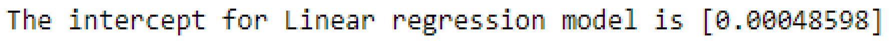


```
The coefficient for Location_Ahmedabad is 687584201292.4236
The coefficient for Location_Bangalore is 699805594832.9862
The coefficient for Location_Bhubaneswar is 692695914007.4946
The coefficient for Location_Chennai is 686059838931.0664
The coefficient for Location_Delhi is 688154676805.1288
The coefficient for Location_Guwahati is 686632003429.0707
The coefficient for Location_Jaipur is 693072553856.1361
The coefficient for Location_Kanpur is 685104816699.6399
The coefficient for Location_Kolkata is 676623243267.816
The coefficient for Location_Lucknow is 679916837253.336
The coefficient for Location_Mangalore is 691375508015.0393
The coefficient for Location_Mumbai is 683956446580.6107
The coefficient for Location_Nagpur is 684913599510.6261
The coefficient for Location_Pune is 677011824369.2837
The coefficient for Location_Surat is 670562222536.8994
The coefficient for covered_by_any_other_company_N is -434147914814.61536
The coefficient for covered_by_any_other_company_Y is -434147914814.5767
The coefficient for Alcohol_Daily is 141523547971.3028
The coefficient for Alcohol_No is 216001320106.79977
The coefficient for Alcohol_Rare is 226579653080.719
The coefficient for exercise_Extreme is 225654157037.55765
The coefficient for exercise_Moderate is 272962855243.51978
The coefficient for exercise_No is 223508974693.66367
The coefficient for years_of_insurance_with_us is -0.0019316739162343044
The coefficient for regular_checkup_lasy_year is -0.03797786204721378
The coefficient for adventure_sports is 0.0024153348171661454
The coefficient for visited_doctor_last_1_year is -0.002953776858120421
The coefficient for daily_avg_steps is -0.002377301245713873
The coefficient for age is 0.0032987231801227834
The coefficient for heart_decs_history is 0.0022709526733823517
The coefficient for other_major_decs_history is 0.0005288158805611902
The coefficient for avg_glucose_level is 0.0013525885403538182
The coefficient for bmi is -0.000506132792710216
The coefficient for weight is 0.969670138514694
The coefficient for weight_change_in_last_one_year is 0.019559938656373368
The coefficient for fat_percentage is -0.0008587722127176572
```

Variables showing Positive effect on regression model are Gender, smoking_status, Location, Alcohol, exercise, adventure_sports, age, heart_decs_history,other_major_decs_history , avg_glucose_level, weight and weight_change_in_last_one_year .

These factors highly influencing our model. The higher the value of the beta coefficient, the higher is the impact

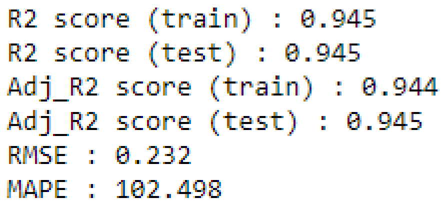
Let us check the intercept for the model



```
The intercept for Linear regression model is [0.00048598]
```

Evaluation of Linear regression model-

Evaluation helps to judge the performance of any machine learning model that would provide best results to our test data.
Fundamentally three types of evaluation metrics are used to evaluate linear regression model.
- R2/ Adjusted R2 Score
- Mean Square Error (MSE) / Root Mean Square Error (RMSE)
- Mean Absolute Error (MAE) / Mean Absolute percentage error (MAPE)



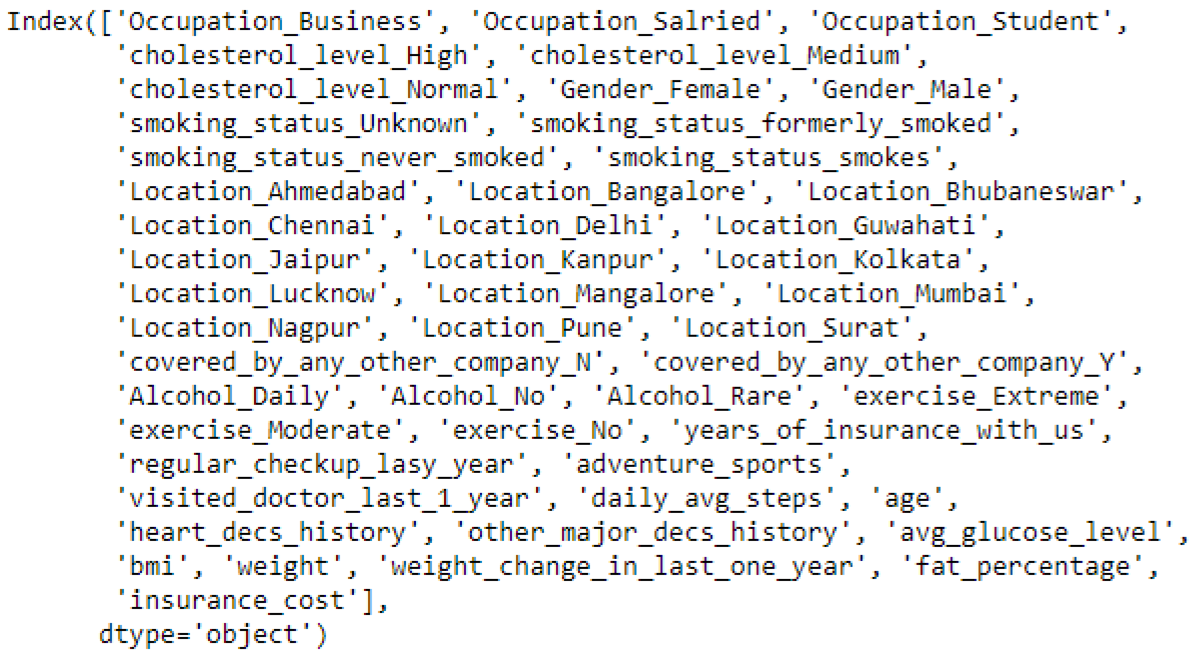
```
R2 score (train) : 0.945
R2 score (test) : 0.945
Adj_R2 score (train) : 0.944
Adj_R2 score (test) : 0.945
RMSE : 0.232
MAPE : 102.498
```

As we had seen, some of the independent variables are correlated and thus we can see they are causing problem of multicollinearity in the model. The above model coefficients also indicating the problem of multicollinearity.

OLS Output:-

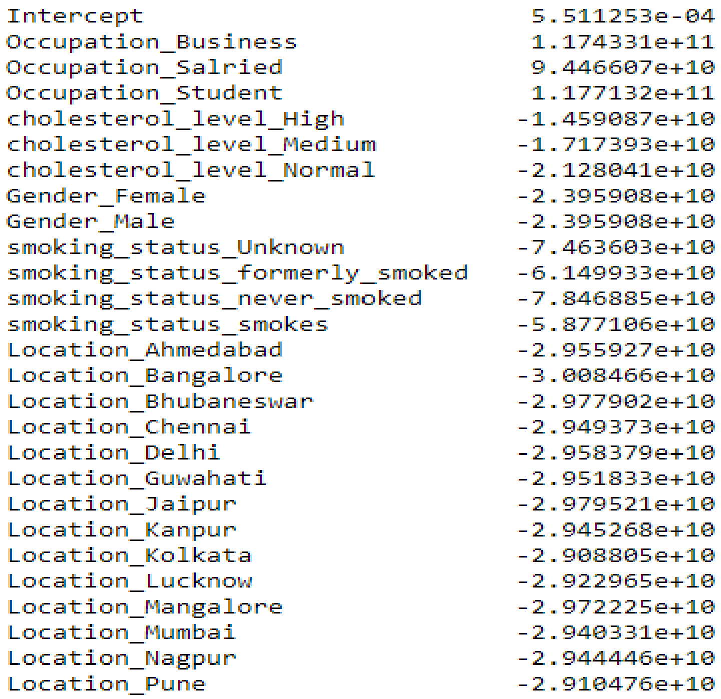
R^2 is not a reliable metric as it always increases with addition of more attributes even if the attributes have no influence on the predicted variable. Instead we use adjusted R^2 which removes the statistical chance that improves R^2.
Scikit does not provide a facility for adjusted R^2... so we use statsmodel, a library that gives results similar to

Data Columns:-

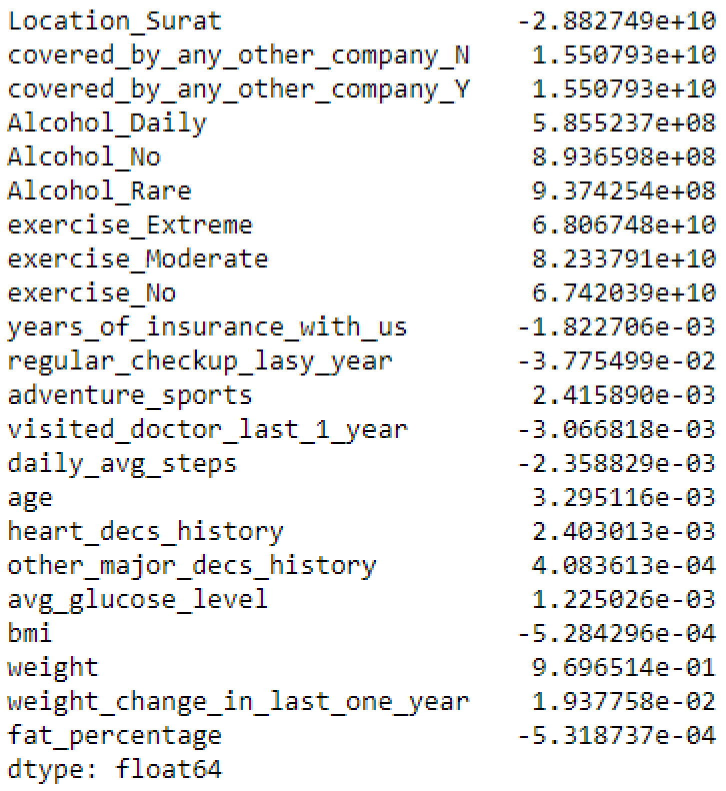


```
Index(['Occupation_Business', 'Occupation_Salried', 'Occupation_Student',
       'cholesterol_level_High', 'cholesterol_level_Medium',
       'cholesterol_level_Normal', 'Gender_Female', 'Gender_Male',
       'smoking_status_Unknown', 'smoking_status_formerly_smoked',
       'smoking_status_never_smoked', 'smoking_status_smokes',
       'Location_Ahmedabad', 'Location_Bangalore', 'Location_Bhubaneswar',
       'Location_Chennai', 'Location_Delhi', 'Location_Guwahati',
       'Location_Jaipur', 'Location_Kanpur', 'Location_Kolkata',
       'Location_Lucknow', 'Location_Mangalore', 'Location_Mumbai',
       'Location_Nagpur', 'Location_Pune', 'Location_Surat',
       'covered_by_any_other_company_N', 'covered_by_any_other_company_Y',
       'Alcohol_Daily', 'Alcohol_No', 'Alcohol_Rare', 'exercise_Extreme',
       'exercise_Moderate', 'exercise_No', 'years_of_insurance_with_us',
       'regular_checkup_lasy_year', 'adventure_sports',
       'visited_doctor_last_1_year', 'daily_avg_steps', 'age',
       'heart_decs_history', 'other_major_decs_history', 'avg_glucose_level',
       'bmi', 'weight', 'weight_change_in_last_one_year', 'fat_percentage',
       'insurance_cost'],
      dtype='object')
```

The parameter after building Linear regression using statsmodel:

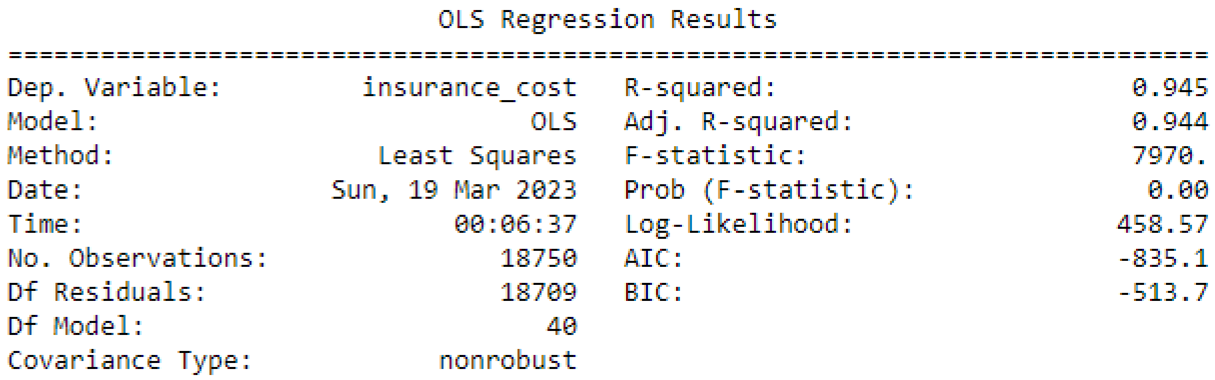


```
Intercept                           5.511253e-04
Occupation_Business                 1.174331e+11
Occupation_Salried                  9.446607e+10
Occupation_Student                  1.177132e+11
cholesterol_level_High             -1.459087e+10
cholesterol_level_Medium           -1.717393e+10
cholesterol_level_Normal           -2.128041e+10
Gender_Female                      -2.395908e+10
Gender_Male                        -2.395908e+10
smoking_status_Unknown             -7.463603e+10
smoking_status_formerly_smoked     -6.149933e+10
smoking_status_never_smoked        -7.846885e+10
smoking_status_smokes              -5.877106e+10
Location_Ahmedabad                 -2.955927e+10
Location_Bangalore                 -3.008466e+10
Location_Bhubaneswar               -2.977902e+10
Location_Chennai                   -2.949373e+10
Location_Delhi                     -2.958379e+10
Location_Guwahati                  -2.951833e+10
Location_Jaipur                    -2.979521e+10
Location_Kanpur                    -2.945268e+10
Location_Kolkata                   -2.908805e+10
Location_Lucknow                   -2.922965e+10
Location_Mangalore                 -2.972225e+10
Location_Mumbai                    -2.940331e+10
Location_Nagpur                    -2.944446e+10
Location_Pune                      -2.910476e+10
```

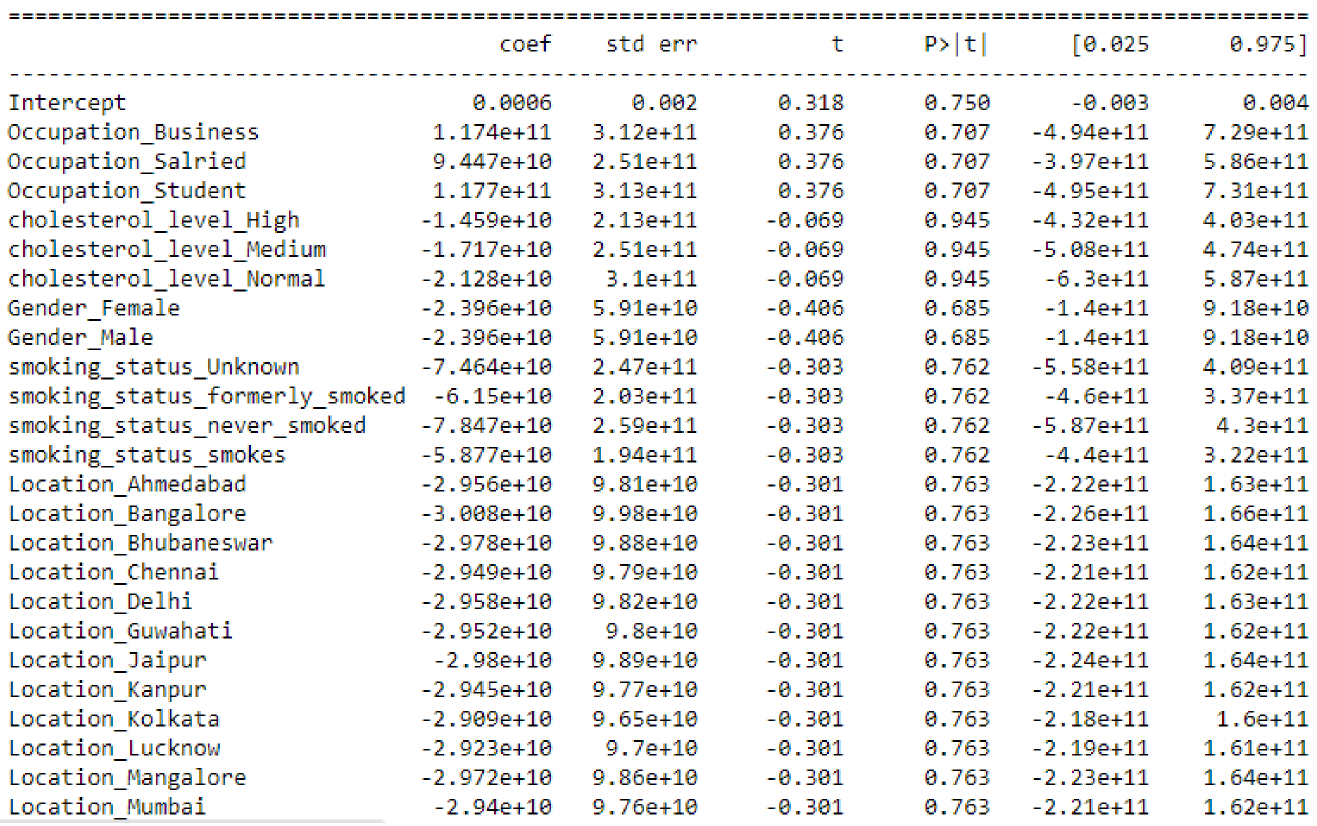


```
Location_Surat                     -2.882749e+10
covered_by_any_other_company_N      1.550793e+10
covered_by_any_other_company_Y      1.550793e+10
Alcohol_Daily                       5.855237e+08
Alcohol_No                          8.936598e+08
Alcohol_Rare                        9.374254e+08
exercise_Extreme                    6.806748e+10
exercise_Moderate                   8.233791e+10
exercise_No                         6.742039e+10
years_of_insurance_with_us         -1.822706e-03
regular_checkup_lasy_year          -3.775499e-02
adventure_sports                    2.415890e-03
visited_doctor_last_1_year         -3.066818e-03
daily_avg_steps                    -2.358829e-03
age                                 3.295116e-03
heart_decs_history                  2.403013e-03
other_major_decs_history            4.083613e-04
avg_glucose_level                   1.225026e-03
bmi                                -5.284296e-04
weight                              9.696514e-01
weight_change_in_last_one_year      1.937758e-02
fat_percentage                     -5.318737e-04
dtype: float64
```

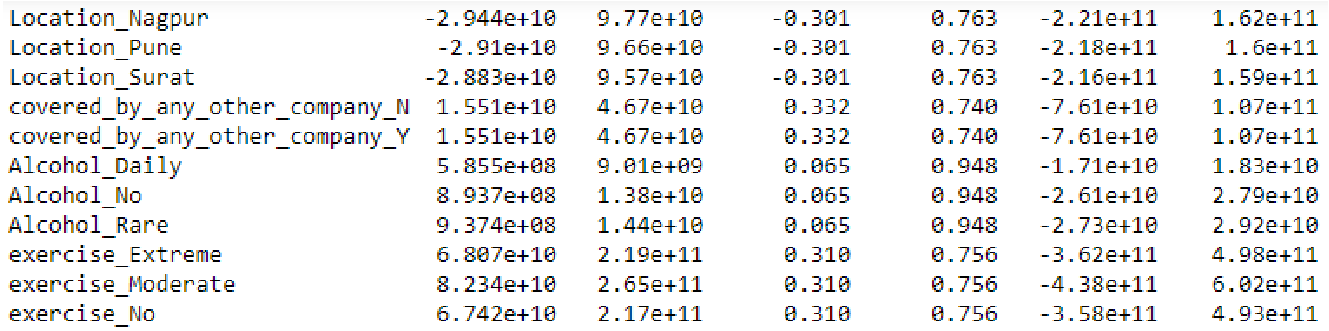
OLS Summary Result:-



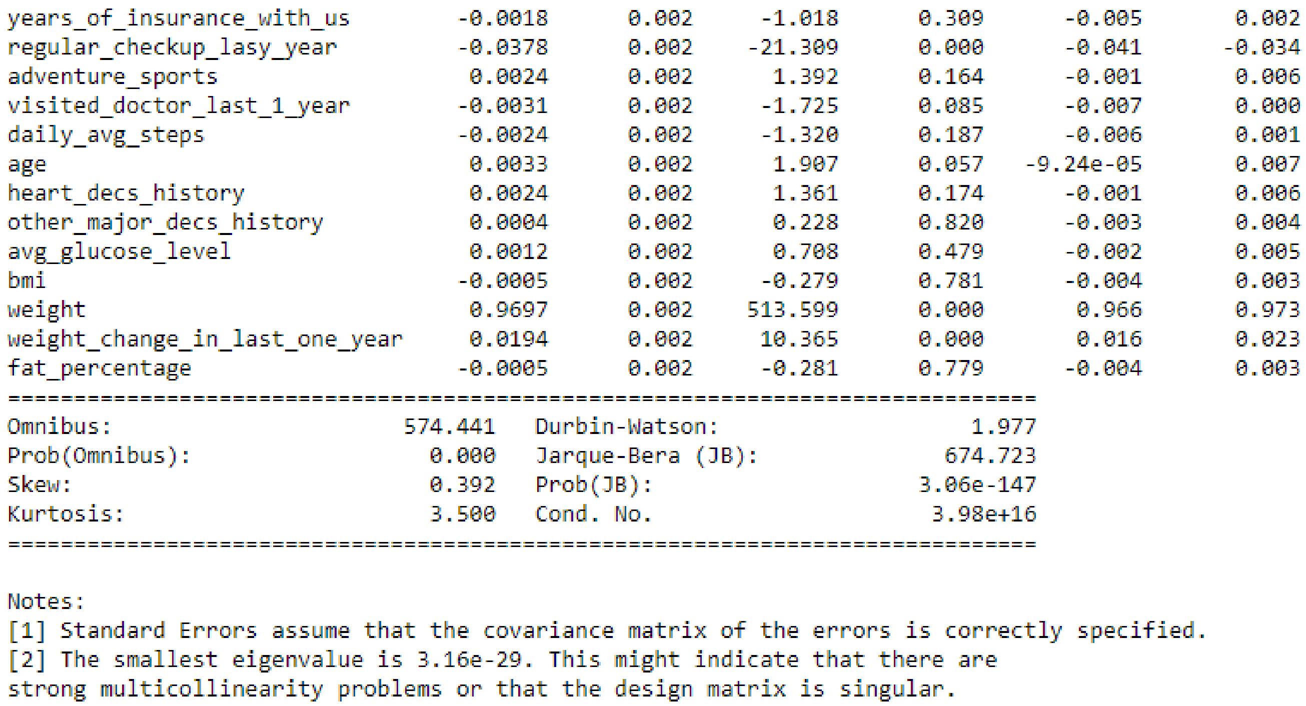
```
                            OLS Regression Results
==============================================================================
Dep. Variable:         insurance_cost   R-squared:                       0.945
Model:                            OLS   Adj. R-squared:                  0.944
Method:                 Least Squares   F-statistic:                     7970.
Date:                Sun, 19 Mar 2023   Prob (F-statistic):               0.00
Time:                        00:06:37   Log-Likelihood:                 458.57
No. Observations:               18750   AIC:                            -835.1
Df Residuals:                   18709   BIC:                            -513.7
Df Model:                          40
Covariance Type:            nonrobust
```



```
===============================================================================================
                                      coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------
Intercept                           0.0006      0.002      0.318      0.750      -0.003       0.004
Occupation_Business              1.174e+11   3.12e+11      0.376      0.707   -4.94e+11    7.29e+11
Occupation_Salried               9.447e+10   2.51e+11      0.376      0.707   -3.97e+11    5.86e+11
Occupation_Student               1.177e+11   3.13e+11      0.376      0.707   -4.95e+11    7.31e+11
cholesterol_level_High          -1.459e+10   2.13e+11     -0.069      0.945   -4.32e+11    4.03e+11
cholesterol_level_Medium        -1.717e+10   2.51e+11     -0.069      0.945   -5.08e+11    4.74e+11
cholesterol_level_Normal        -2.128e+10    3.1e+11     -0.069      0.945    -6.3e+11    5.87e+11
Gender_Female                   -2.396e+10   5.91e+10     -0.406      0.685    -1.4e+11    9.18e+10
Gender_Male                     -2.396e+10   5.91e+10     -0.406      0.685    -1.4e+11    9.18e+10
smoking_status_Unknown          -7.464e+10   2.47e+11     -0.303      0.762   -5.58e+11    4.09e+11
smoking_status_formerly_smoked   -6.15e+10   2.03e+11     -0.303      0.762    -4.6e+11    3.37e+11
smoking_status_never_smoked     -7.847e+10   2.59e+11     -0.303      0.762   -5.87e+11     4.3e+11
smoking_status_smokes           -5.877e+10   1.94e+11     -0.303      0.762    -4.4e+11    3.22e+11
Location_Ahmedabad              -2.956e+10   9.81e+10     -0.301      0.763   -2.22e+11    1.63e+11
Location_Bangalore              -3.008e+10   9.98e+10     -0.301      0.763   -2.26e+11    1.66e+11
Location_Bhubaneswar            -2.978e+10   9.88e+10     -0.301      0.763   -2.23e+11    1.64e+11
Location_Chennai                -2.949e+10   9.79e+10     -0.301      0.763   -2.21e+11    1.62e+11
Location_Delhi                  -2.958e+10   9.82e+10     -0.301      0.763   -2.22e+11    1.63e+11
Location_Guwahati               -2.952e+10    9.8e+10     -0.301      0.763   -2.22e+11    1.62e+11
Location_Jaipur                  -2.98e+10   9.89e+10     -0.301      0.763   -2.24e+11    1.64e+11
Location_Kanpur                 -2.945e+10   9.77e+10     -0.301      0.763   -2.21e+11    1.62e+11
Location_Kolkata                -2.909e+10   9.65e+10     -0.301      0.763   -2.18e+11     1.6e+11
Location_Lucknow                -2.923e+10    9.7e+10     -0.301      0.763   -2.19e+11    1.61e+11
Location_Mangalore              -2.972e+10   9.86e+10     -0.301      0.763   -2.23e+11    1.64e+11
Location_Mumbai                  -2.94e+10   9.76e+10     -0.301      0.763   -2.21e+11    1.62e+11
```



```
Location_Nagpur                 -2.944e+10   9.77e+10     -0.301      0.763   -2.21e+11    1.62e+11
Location_Pune                    -2.91e+10   9.66e+10     -0.301      0.763   -2.18e+11     1.6e+11
Location_Surat                  -2.883e+10   9.57e+10     -0.301      0.763   -2.16e+11    1.59e+11
covered_by_any_other_company_N   1.551e+10   4.67e+10      0.332      0.740   -7.61e+10    1.07e+11
covered_by_any_other_company_Y   1.551e+10   4.67e+10      0.332      0.740   -7.61e+10    1.07e+11
Alcohol_Daily                    5.855e+08   9.01e+09      0.065      0.948   -1.71e+10    1.83e+10
Alcohol_No                       8.937e+08   1.38e+10      0.065      0.948   -2.61e+10    2.79e+10
Alcohol_Rare                     9.374e+08   1.44e+10      0.065      0.948   -2.73e+10    2.92e+10
exercise_Extreme                 6.807e+10   2.19e+11      0.310      0.756   -3.62e+11    4.98e+11
exercise_Moderate                8.234e+10   2.65e+11      0.310      0.756   -4.38e+11    6.02e+11
exercise_No                      6.742e+10   2.17e+11      0.310      0.756   -3.58e+11    4.93e+11
```



```
years_of_insurance_with_us         -0.0018      0.002     -1.018      0.309      -0.005       0.002
regular_checkup_lasy_year          -0.0378      0.002    -21.309      0.000      -0.041      -0.034
adventure_sports                    0.0024      0.002      1.392      0.164      -0.001       0.006
visited_doctor_last_1_year         -0.0031      0.002     -1.725      0.085      -0.007       0.000
daily_avg_steps                    -0.0024      0.002     -1.320      0.187      -0.006       0.001
age                                 0.0033      0.002      1.907      0.057   -9.24e-05       0.007
heart_decs_history                  0.0024      0.002      1.361      0.174      -0.001       0.006
other_major_decs_history            0.0004      0.002      0.228      0.820      -0.003       0.004
avg_glucose_level                   0.0012      0.002      0.708      0.479      -0.002       0.005
bmi                                -0.0005      0.002     -0.279      0.781      -0.004       0.003
weight                              0.9697      0.002    513.599      0.000       0.966       0.973
weight_change_in_last_one_year      0.0194      0.002     10.365      0.000       0.016       0.023
fat_percentage                     -0.0005      0.002     -0.281      0.779      -0.004       0.003
==============================================================================
Omnibus:                      574.441   Durbin-Watson:                   1.977
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              674.723
Skew:                           0.392   Prob(JB):                    3.06e-147
Kurtosis:                       3.500   Cond. No.                     3.98e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 3.16e-29. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

Image25.Ols_Summary

If we look at the p-values of some of the variables, the values seem to be pretty high, which means they aren’t significant. That means we can drop those variables from the model.

Key Observartion:-

1. Adjusted. R-squared reflects the fit of the model. R-squared values range from 0 to 1, where a higher value generally indicates a better fit, assuming certain conditions are met.
2. const coefficient is your Y-intercept. It means that if both the interest_rate and unemployment_rate coefficients are zero, then the expected output (i.e., the Y) would be equal to the const coefficient.
3. interest_rate coefficient represents the change in the output Y due to a change of one unit in the interest rate (everything else held constant)
4. unemployment_rate coefficient represents the change in the output Y due to a change of one unit in the unemployment rate (everything else held constant)
5. std err reflects the level of accuracy of the coefficients. The lower it is, the higher is the level of accuracy
6. P >|t| is your *p-value*. A p-value of less than 0.05 is considered to be statistically significant
7. Confidence Interval represents the range in which our coefficients are likely to fall (with a likelihood of 95%)

From the above summary of the model, we can see that all of the features are significant as some of the features have pvalues > 0.05.In addition, the output from the sklearn's Linear Regression & statsmodel's OLS are similar. Therefore, we will continue using sklearn's Linear Regression model for further analysis.

Since this is regression, plot the predicted y value vs actual y values for the test data. A good model's prediction will be close to actual leading to high R and R2 values

```
<matplotlib.collections.PathCollection at 0x2432e1c8d90>
```

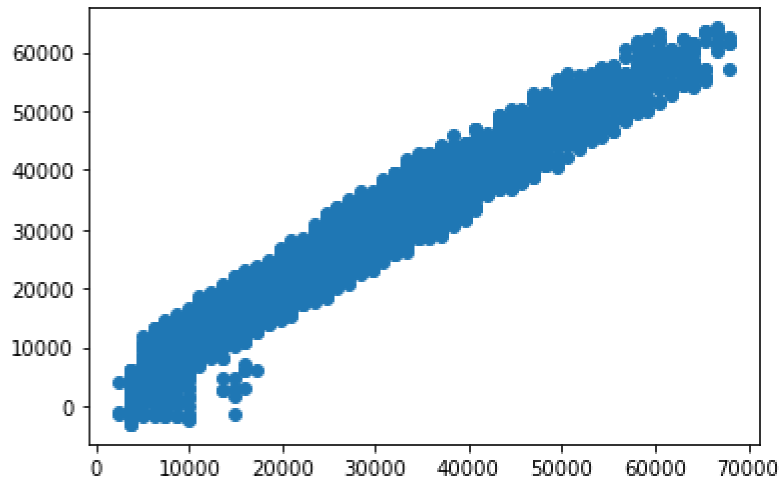
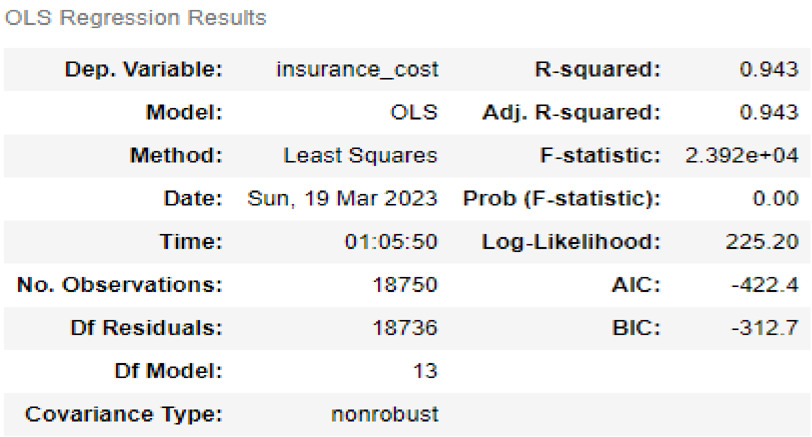


Image26.Scatter_plot

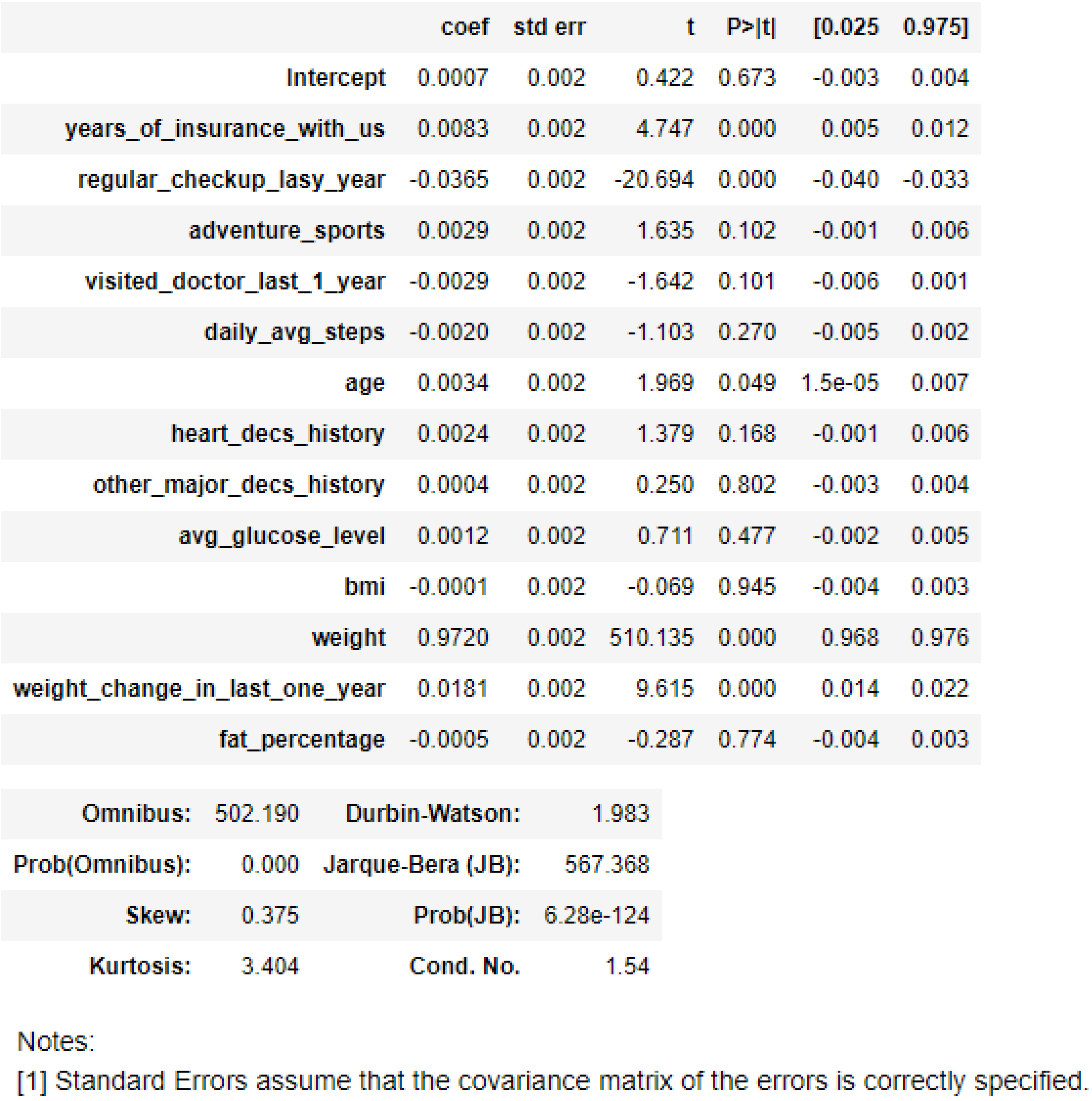
Linear Regression using statsmodel

OLS output after finding important features:-



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	2.392e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:05:50	Log-Likelihood:	225.20
No. Observations:	18750	AIC:	-422.4
Df Residuals:	18736	BIC:	-312.7
Df Model:	13		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0007	0.002	0.422	0.673	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.747	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.694	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.635	0.102	-0.001	0.006
visited_doctor_last_1_year	-0.0029	0.002	-1.642	0.101	-0.006	0.001
daily_avg_steps	-0.0020	0.002	-1.103	0.270	-0.005	0.002
age	0.0034	0.002	1.969	0.049	1.5e-05	0.007
heart_decs_history	0.0024	0.002	1.379	0.168	-0.001	0.006
other_major_decs_history	0.0004	0.002	0.250	0.802	-0.003	0.004
avg_glucose_level	0.0012	0.002	0.711	0.477	-0.002	0.005
bmi	-0.0001	0.002	-0.069	0.945	-0.004	0.003
weight	0.9720	0.002	510.135	0.000	0.968	0.976
weight_change_in_last_one_year	0.0181	0.002	9.615	0.000	0.014	0.022
fat_percentage	-0.0005	0.002	-0.287	0.774	-0.004	0.003

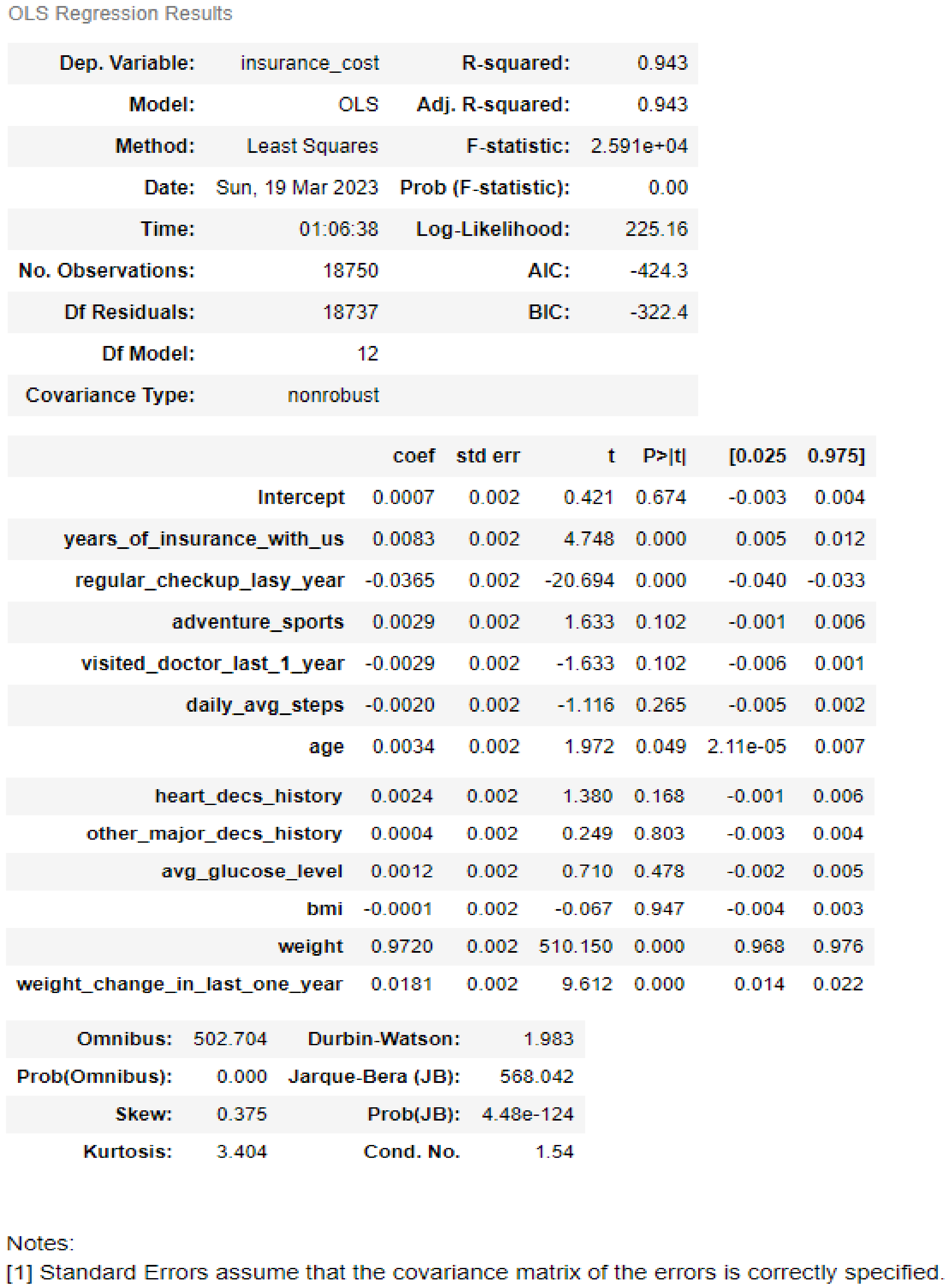
Omnibus:	502.190	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	567.368
Skew:	0.375	Prob(JB):	6.28e-124
Kurtosis:	3.404	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

From the above ols summary we get fat_percentage have pvalue> 0.05. So we drop it from the expression and build the model.

Model 2:-

Ols Summary:-



OLS Regression Results

Dep. Variable:	insurance_cost	**R-squared:**	0.943
Model:	OLS	**Adj. R-squared:**	0.943
Method:	Least Squares	**F-statistic:**	2.591e+04
Date:	Sun, 19 Mar 2023	**Prob (F-statistic):**	0.00
Time:	01:06:38	**Log-Likelihood:**	225.16
No. Observations:	18750	**AIC:**	-424.3
Df Residuals:	18737	**BIC:**	-322.4
Df Model:	12		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.0007	0.002	0.421	0.674	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.748	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.694	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.633	0.102	-0.001	0.006
visited_doctor_last_1_year	-0.0029	0.002	-1.633	0.102	-0.006	0.001
daily_avg_steps	-0.0020	0.002	-1.116	0.265	-0.005	0.002
age	0.0034	0.002	1.972	0.049	2.11e-05	0.007
heart_decs_history	0.0024	0.002	1.380	0.168	-0.001	0.006
other_major_decs_history	0.0004	0.002	0.249	0.803	-0.003	0.004
avg_glucose_level	0.0012	0.002	0.710	0.478	-0.002	0.005
bmi	-0.0001	0.002	-0.067	0.947	-0.004	0.003
weight	0.9720	0.002	510.150	0.000	0.968	0.976
weight_change_in_last_one_year	0.0181	0.002	9.612	0.000	0.014	0.022

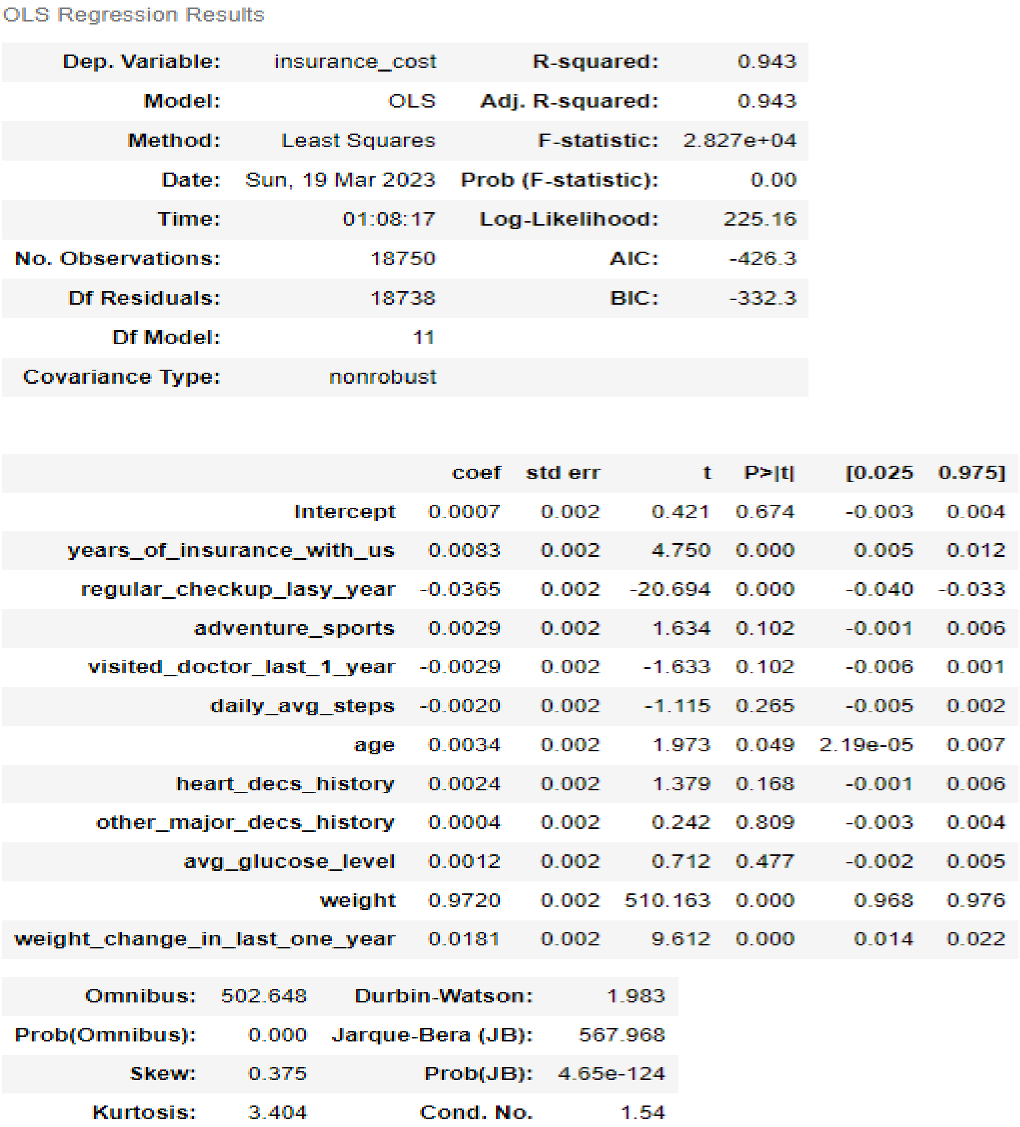
Omnibus:	502.704	**Durbin-Watson:**	1.983
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	568.042
Skew:	0.375	**Prob(JB):**	4.48e-124
Kurtosis:	3.404	**Cond. No.**	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

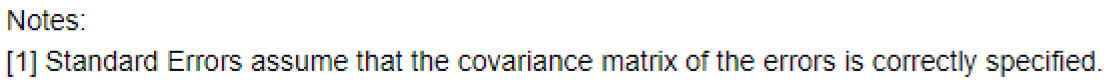
The ols summary still have variable whose p- value> 0.05. Now we drop bmi because it has p- value> 0.05.

Model 3:-

Ols Summary:-



```
OLS Regression Results
       Dep. Variable:   insurance_cost        R-squared:      0.943
               Model:              OLS   Adj. R-squared:      0.943
              Method:    Least Squares      F-statistic:  2.827e+04
                Date: Sun, 19 Mar 2023 Prob (F-statistic):     0.00
                Time:         01:08:17   Log-Likelihood:     225.16
   No. Observations:            18750              AIC:     -426.3
       Df Residuals:            18738              BIC:     -332.3
            Df Model:               11
    Covariance Type:        nonrobust

                                 coef  std err        t   P>|t|    [0.025   0.975]
                    Intercept  0.0007    0.002    0.421   0.674    -0.003    0.004
      years_of_insurance_with_us  0.0083    0.002    4.750   0.000     0.005    0.012
      regular_checkup_lasy_year -0.0365    0.002  -20.694   0.000    -0.040   -0.033
              adventure_sports  0.0029    0.002    1.634   0.102    -0.001    0.006
       visited_doctor_last_1_year -0.0029    0.002   -1.633   0.102    -0.006    0.001
               daily_avg_steps -0.0020    0.002   -1.115   0.265    -0.005    0.002
                           age  0.0034    0.002    1.973   0.049  2.19e-05    0.007
            heart_decs_history  0.0024    0.002    1.379   0.168    -0.001    0.006
      other_major_decs_history  0.0004    0.002    0.242   0.809    -0.003    0.004
              avg_glucose_level  0.0012    0.002    0.712   0.477    -0.002    0.005
                        weight  0.9720    0.002  510.163   0.000     0.968    0.976
 weight_change_in_last_one_year  0.0181    0.002    9.612   0.000     0.014    0.022

      Omnibus:  502.648     Durbin-Watson:      1.983
Prob(Omnibus):    0.000  Jarque-Bera (JB):    567.968
         Skew:    0.375          Prob(JB):  4.65e-124
     Kurtosis:    3.404         Cond. No.       1.54
```

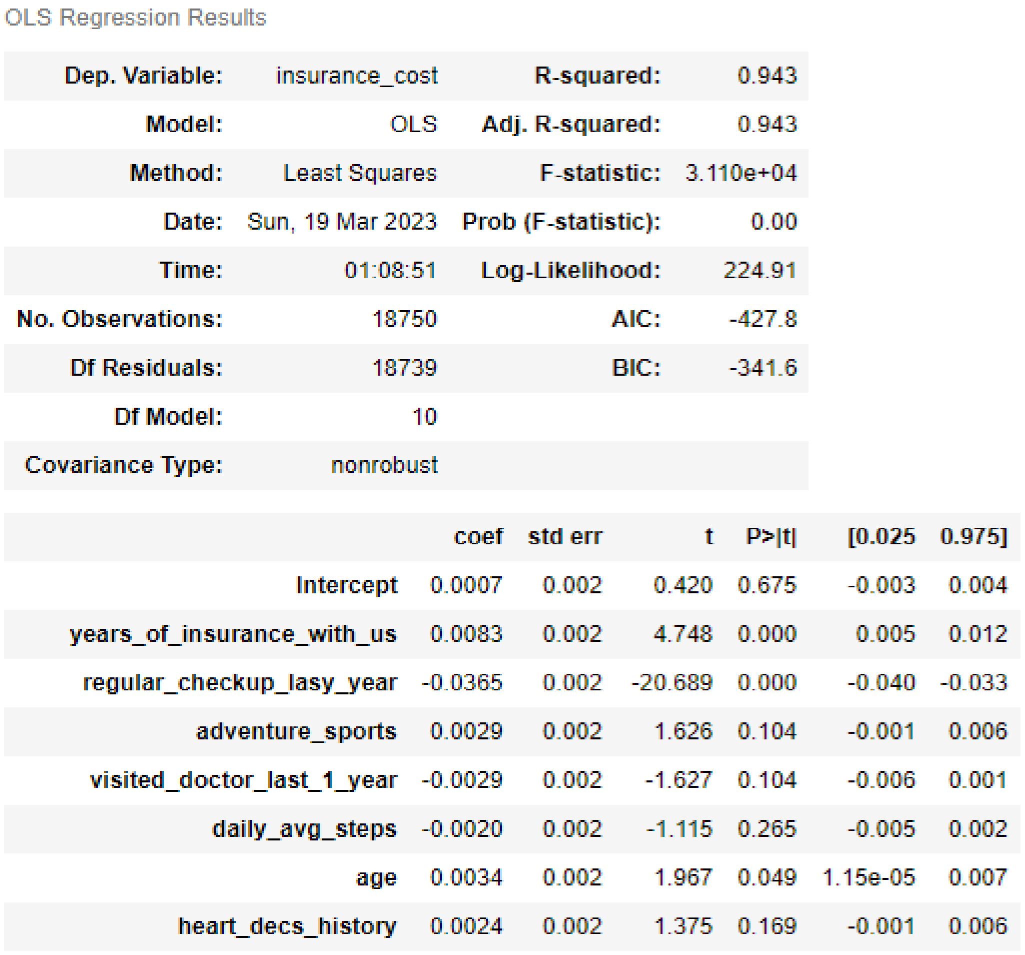


Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we drop avg_glucose_level

Model 4:-

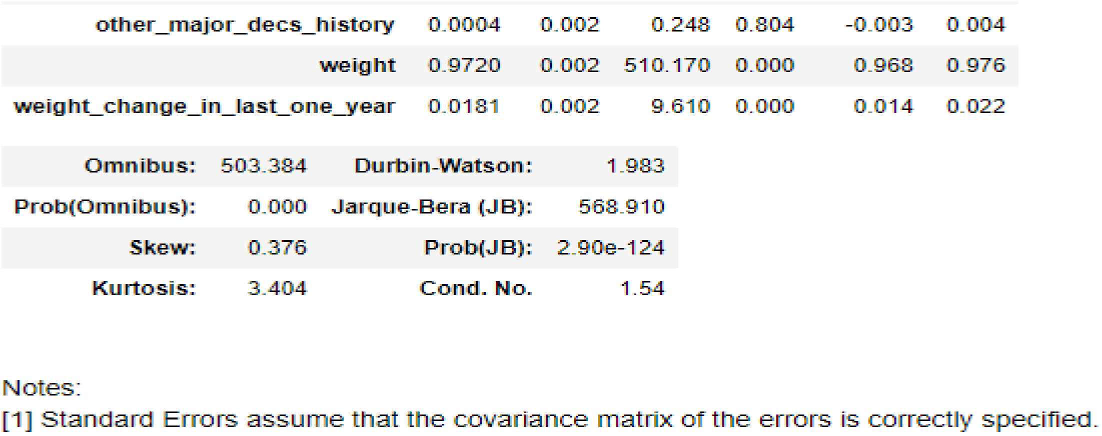
Ols Summary



OLS Regression Results

Dep. Variable:	insurance_cost	**R-squared:**	0.943
Model:	OLS	**Adj. R-squared:**	0.943
Method:	Least Squares	**F-statistic:**	3.110e+04
Date:	Sun, 19 Mar 2023	**Prob (F-statistic):**	0.00
Time:	01:08:51	**Log-Likelihood:**	224.91
No. Observations:	18750	**AIC:**	-427.8
Df Residuals:	18739	**BIC:**	-341.6
Df Model:	10		
Covariance Type:	nonrobust		

	coef	**std err**	**t**	**P>|t|**	**[0.025**	**0.975]**
Intercept	0.0007	0.002	0.420	0.675	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.748	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.689	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.626	0.104	-0.001	0.006
visited_doctor_last_1_year	-0.0029	0.002	-1.627	0.104	-0.006	0.001
daily_avg_steps	-0.0020	0.002	-1.115	0.265	-0.005	0.002
age	0.0034	0.002	1.967	0.049	1.15e-05	0.007
heart_decs_history	0.0024	0.002	1.375	0.169	-0.001	0.006



other_major_decs_history	0.0004	0.002	0.248	0.804	-0.003	0.004
weight	0.9720	0.002	510.170	0.000	0.968	0.976
weight_change_in_last_one_year	0.0181	0.002	9.610	0.000	0.014	0.022

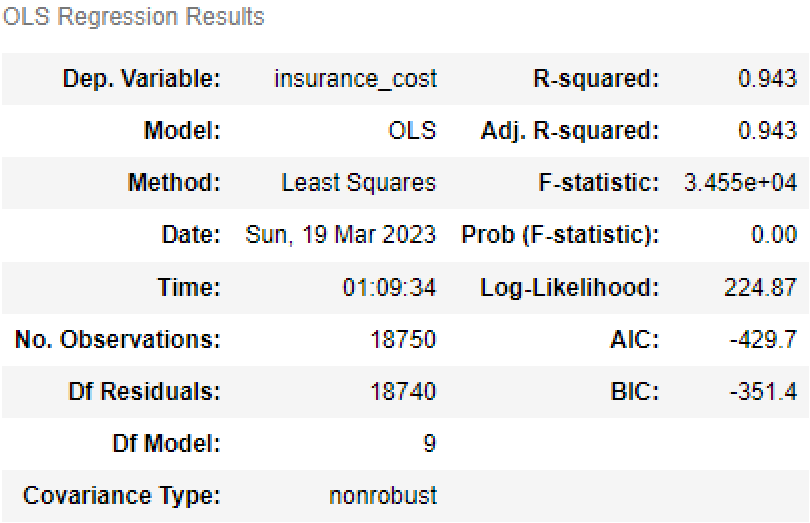
Omnibus:	503.384	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	568.910
Skew:	0.376	Prob(JB):	2.90e-124
Kurtosis:	3.404	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we will drop other_major_decs_history

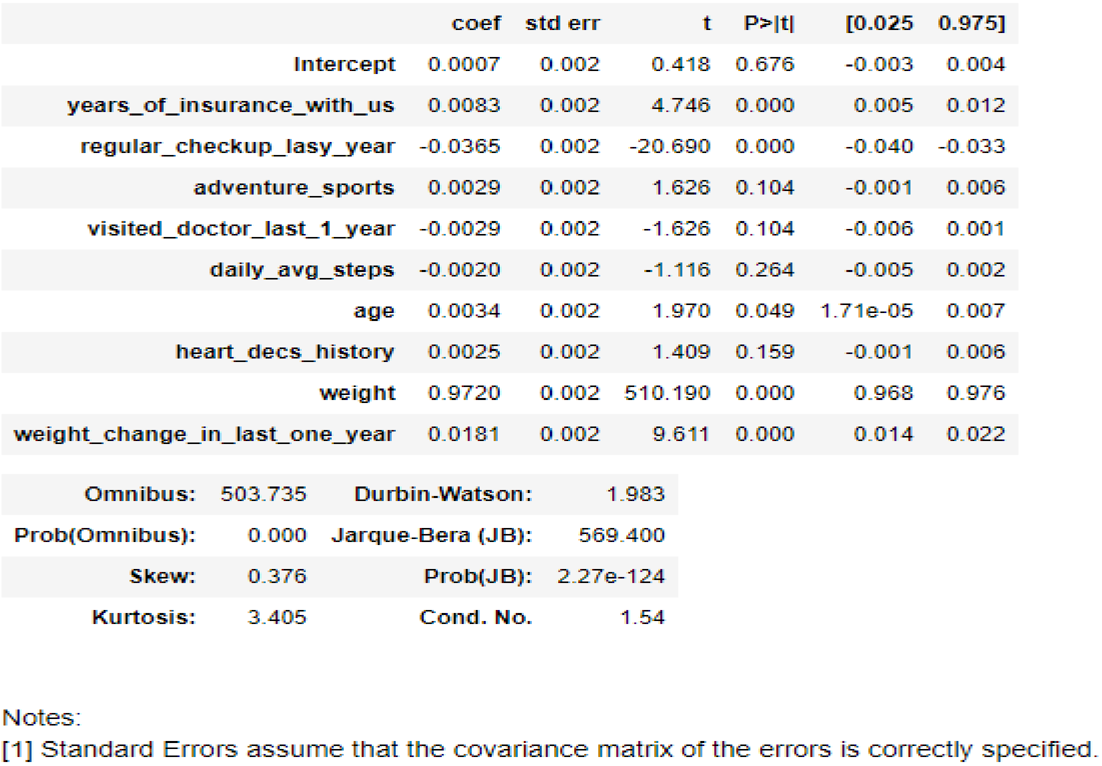
Model 5:-

OLS Summary



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	3.455e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:09:34	Log-Likelihood:	224.87
No. Observations:	18750	AIC:	-429.7
Df Residuals:	18740	BIC:	-351.4
Df Model:	9		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0007	0.002	0.418	0.676	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.746	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.690	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.626	0.104	-0.001	0.006
visited_doctor_last_1_year	-0.0029	0.002	-1.626	0.104	-0.006	0.001
daily_avg_steps	-0.0020	0.002	-1.116	0.264	-0.005	0.002
age	0.0034	0.002	1.970	0.049	1.71e-05	0.007
heart_decs_history	0.0025	0.002	1.409	0.159	-0.001	0.006
weight	0.9720	0.002	510.190	0.000	0.968	0.976
weight_change_in_last_one_year	0.0181	0.002	9.611	0.000	0.014	0.022

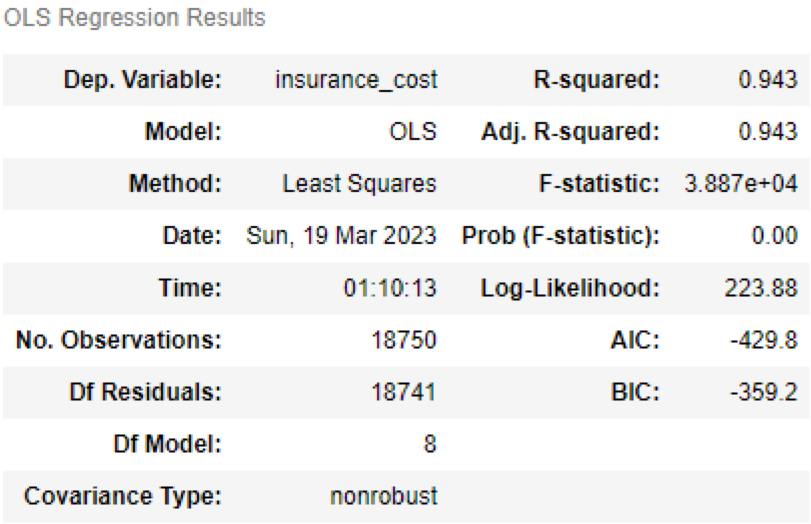
Omnibus:	503.735	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	569.400
Skew:	0.376	Prob(JB):	2.27e-124
Kurtosis:	3.405	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we will drop heart_dec_history

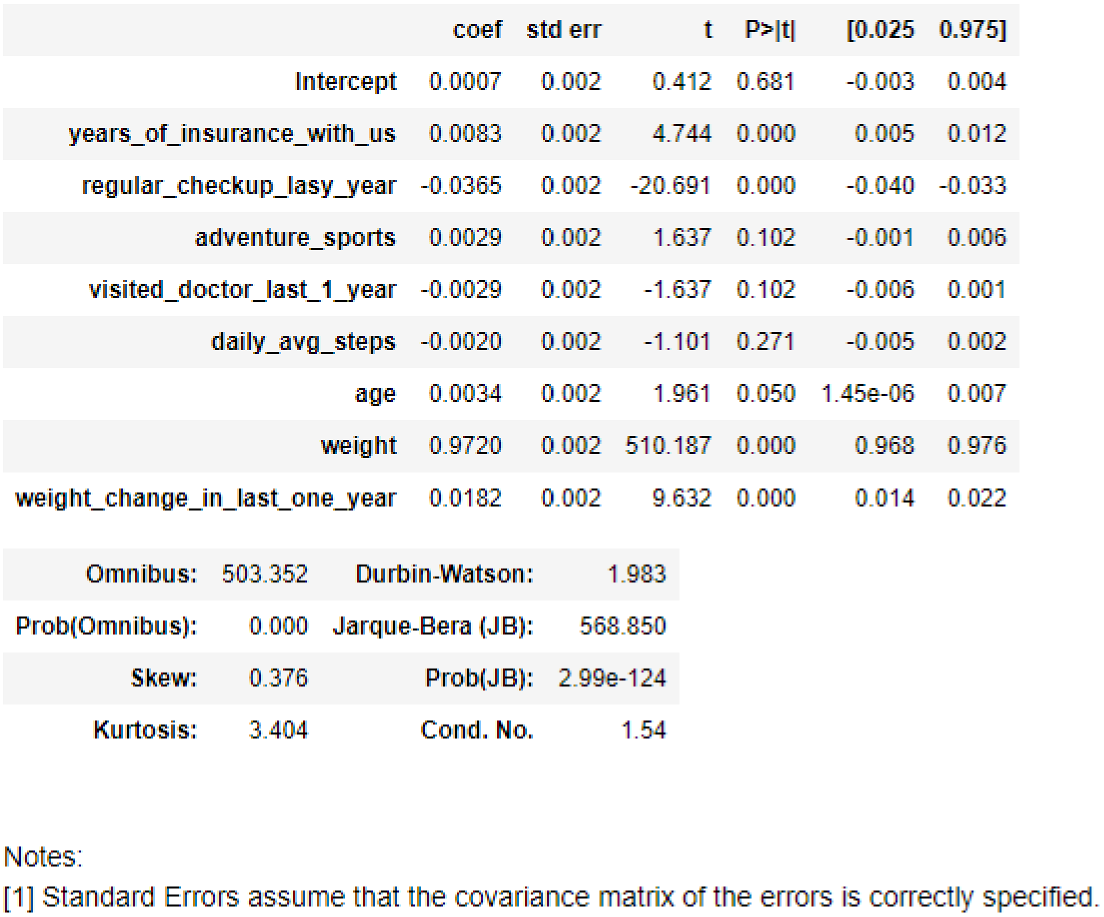
Model 6:-

OLS Summary:-



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	3.887e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:10:13	Log-Likelihood:	223.88
No. Observations:	18750	AIC:	-429.8
Df Residuals:	18741	BIC:	-359.2
Df Model:	8		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0007	0.002	0.412	0.681	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.744	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.691	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.637	0.102	-0.001	0.006
visited_doctor_last_1_year	-0.0029	0.002	-1.637	0.102	-0.006	0.001
daily_avg_steps	-0.0020	0.002	-1.101	0.271	-0.005	0.002
age	0.0034	0.002	1.961	0.050	1.45e-06	0.007
weight	0.9720	0.002	510.187	0.000	0.968	0.976
weight_change_in_last_one_year	0.0182	0.002	9.632	0.000	0.014	0.022

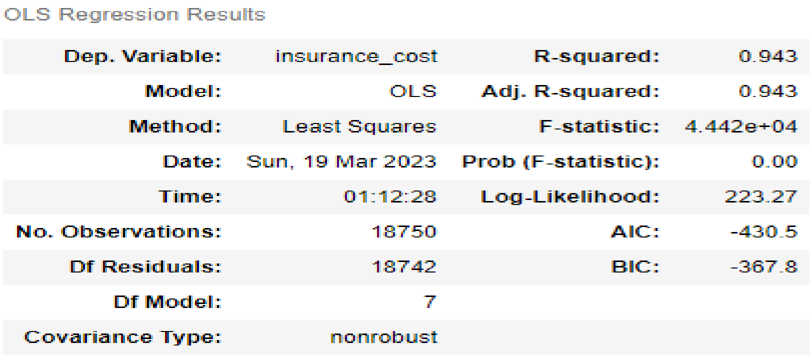
Omnibus:	503.352	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	568.850
Skew:	0.376	Prob(JB):	2.99e-124
Kurtosis:	3.404	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we will drop daily_avg_steps

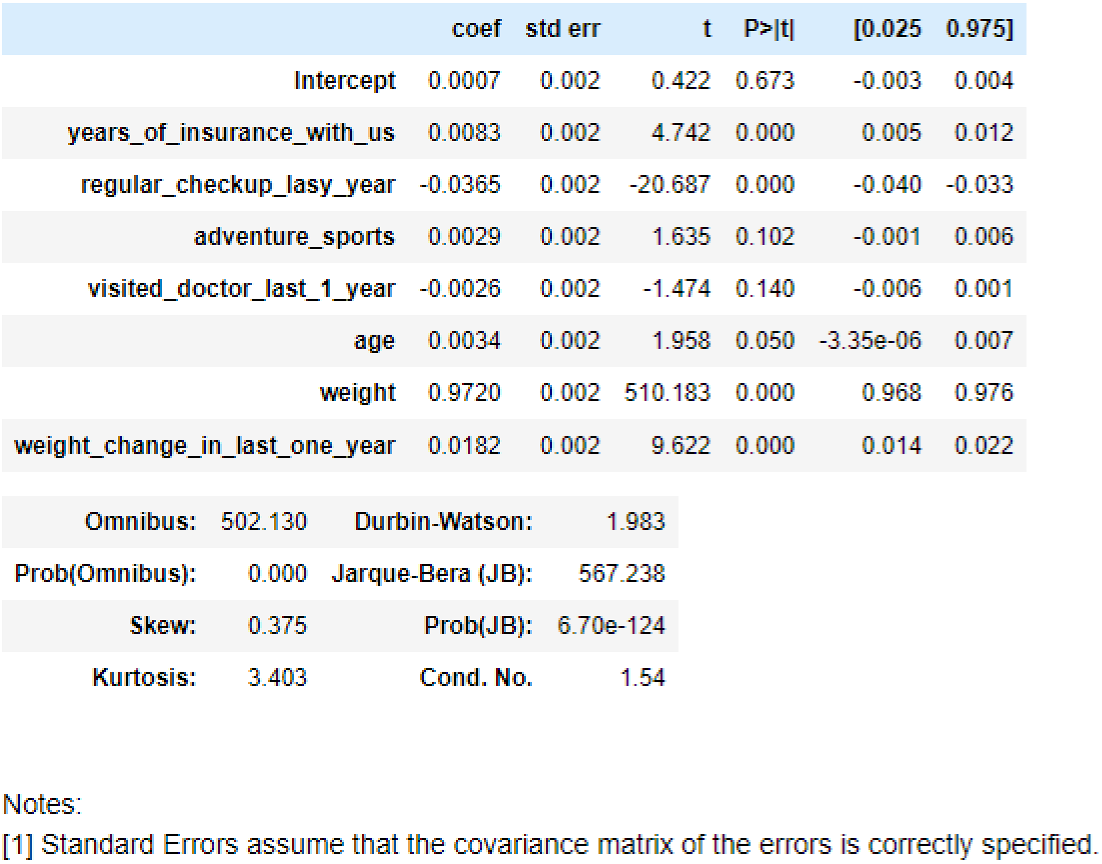
Model 7:-

OLS Summary:-



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	4.442e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:12:28	Log-Likelihood:	223.27
No. Observations:	18750	AIC:	-430.5
Df Residuals:	18742	BIC:	-367.8
Df Model:	7		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0007	0.002	0.422	0.673	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.742	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.687	0.000	-0.040	-0.033
adventure_sports	0.0029	0.002	1.635	0.102	-0.001	0.006
visited_doctor_last_1_year	-0.0026	0.002	-1.474	0.140	-0.006	0.001
age	0.0034	0.002	1.958	0.050	-3.35e-06	0.007
weight	0.9720	0.002	510.183	0.000	0.968	0.976
weight_change_in_last_one_year	0.0182	0.002	9.622	0.000	0.014	0.022

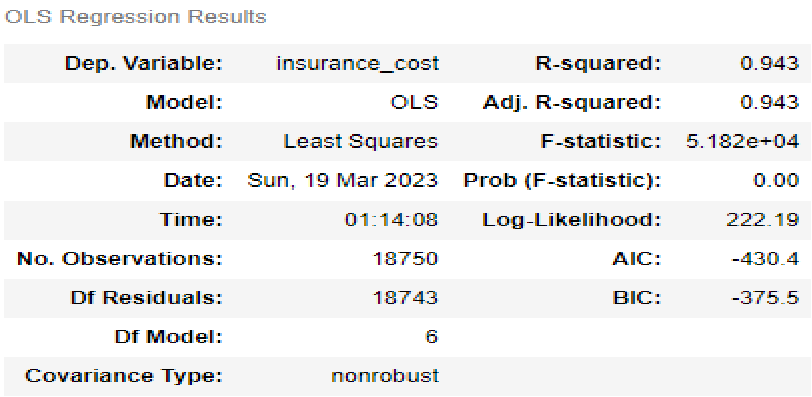
Omnibus:	502.130	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	567.238
Skew:	0.375	Prob(JB):	6.70e-124
Kurtosis:	3.403	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we will drop visited_doctor_last_1_year column

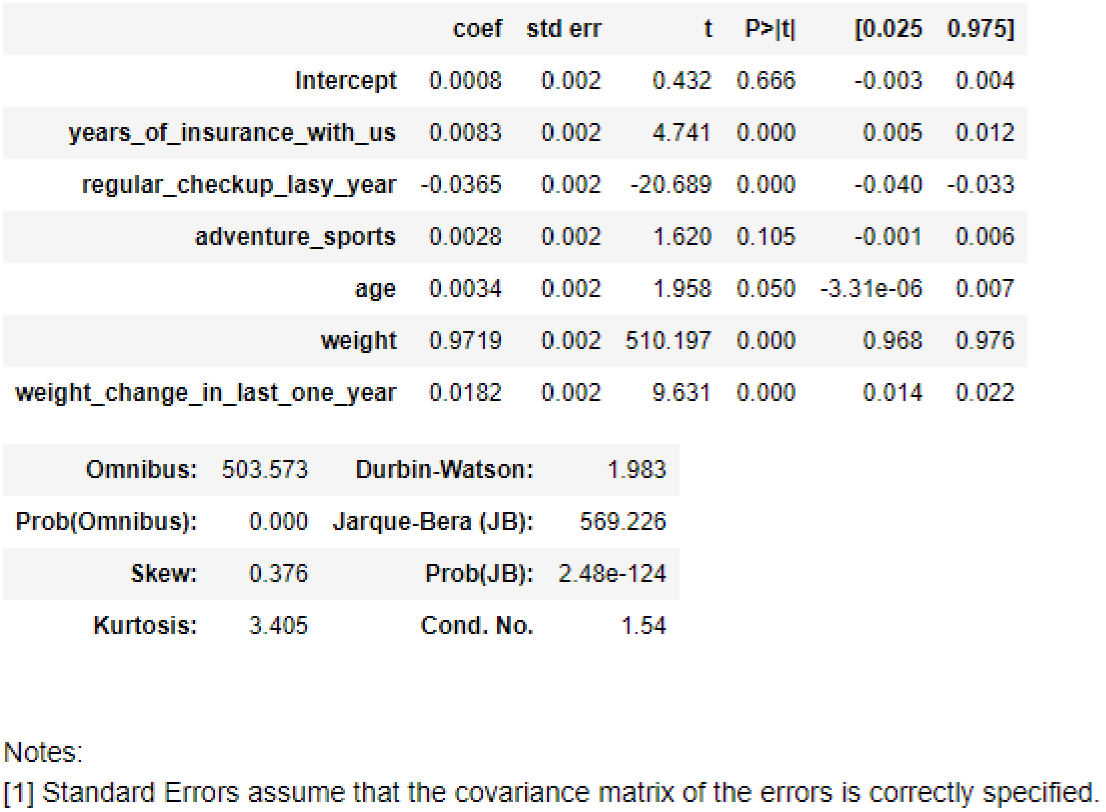
Model 8

OLS Summary:-



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	5.182e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:14:08	Log-Likelihood:	222.19
No. Observations:	18750	AIC:	-430.4
Df Residuals:	18743	BIC:	-375.5
Df Model:	6		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0008	0.002	0.432	0.666	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.741	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.689	0.000	-0.040	-0.033
adventure_sports	0.0028	0.002	1.620	0.105	-0.001	0.006
age	0.0034	0.002	1.958	0.050	-3.31e-06	0.007
weight	0.9719	0.002	510.197	0.000	0.968	0.976
weight_change_in_last_one_year	0.0182	0.002	9.631	0.000	0.014	0.022

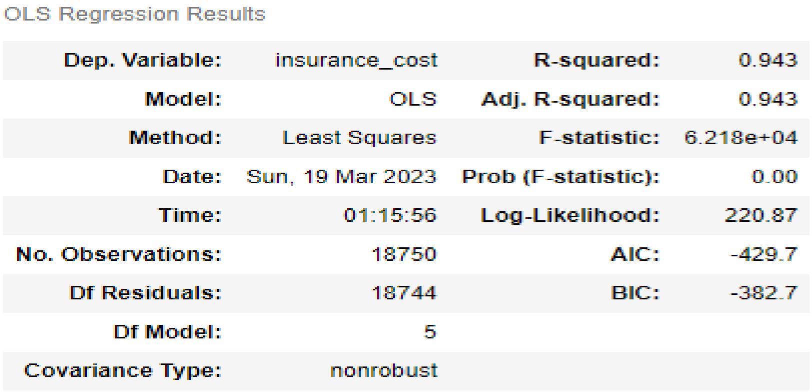
Omnibus:	503.573	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	569.226
Skew:	0.376	Prob(JB):	2.48e-124
Kurtosis:	3.405	Cond. No.	1.54

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Now we will drop adventure sports column

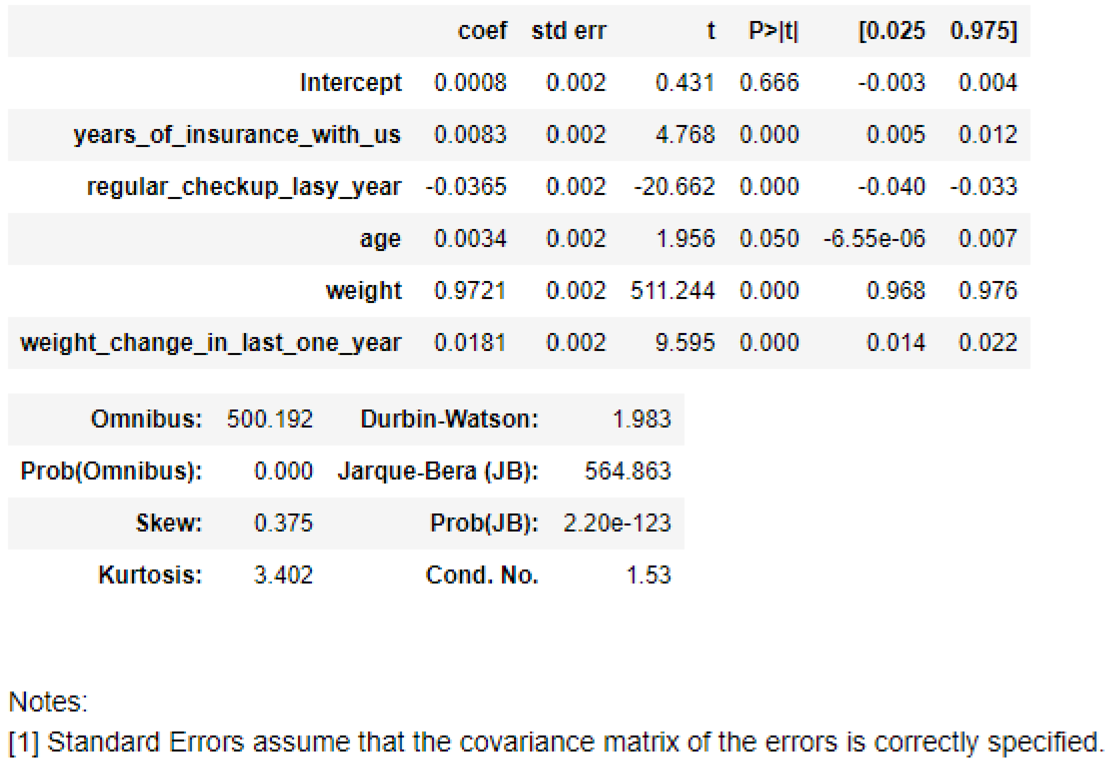
Model 9:-

OLS Summary:-



OLS Regression Results

Dep. Variable:	insurance_cost	R-squared:	0.943
Model:	OLS	Adj. R-squared:	0.943
Method:	Least Squares	F-statistic:	6.218e+04
Date:	Sun, 19 Mar 2023	Prob (F-statistic):	0.00
Time:	01:15:56	Log-Likelihood:	220.87
No. Observations:	18750	AIC:	-429.7
Df Residuals:	18744	BIC:	-382.7
Df Model:	5		
Covariance Type:	nonrobust		



	coef	std err	t	P>|t|	[0.025	0.975]
Intercept	0.0008	0.002	0.431	0.666	-0.003	0.004
years_of_insurance_with_us	0.0083	0.002	4.768	0.000	0.005	0.012
regular_checkup_lasy_year	-0.0365	0.002	-20.662	0.000	-0.040	-0.033
age	0.0034	0.002	1.956	0.050	-6.55e-06	0.007
weight	0.9721	0.002	511.244	0.000	0.968	0.976
weight_change_in_last_one_year	0.0181	0.002	9.595	0.000	0.014	0.022

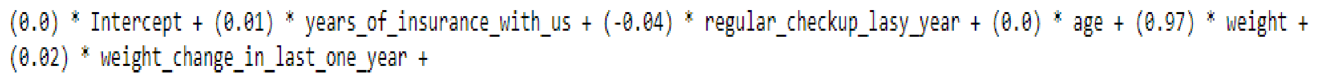
Omnibus:	500.192	Durbin-Watson:	1.983
Prob(Omnibus):	0.000	Jarque-Bera (JB):	564.863
Skew:	0.375	Prob(JB):	2.20e-123
Kurtosis:	3.402	Cond. No.	1.53

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

The overall P value is less than alpha, so rejecting H0 and accepting Ha that atleast 1 regression co-efficient is not 0. Here all regression co-efficients are not 0

Interpretation of the model:-

The final Linear Regression equation is



```
(0.0) * Intercept + (0.01) * years_of_insurance_with_us + (-0.04) * regular_checkup_lasy_year + (0.0) * age + (0.97) * weight +
(0.02) * weight_change_in_last_one_year +
```

Insurance_cost= b0 + b1 * years_of_insurance_with_us + b2 *regular_checkup_lasy_year + b3 * age + b4 * weight + b5 * weight_change_in_one_year+

Insurance_cost= (0.0) * Intercept + (0.01) * years_of_insurance_with_us + (-0.04) *regular_checkup_lasy_year + (0.0) * age + (0.97) * weight + (0.02) * weight_change_in_one_year+

When **heart_decs_history** increases by 1 unit, insurance_cost increases by 0.97 units, keeping all other predictors constant When weight_change_in_one_year increases by 1 unit, insurance_cost increases by 0.02 units, keeping all other predictors constant etcc....

There are also some negative co-efficient values, for instance, regular_checkup_lasy_year has its corresponding co-efficient as -0.04. This implies, when the customer go for regular checkup in years, the insurance_cost decreases by 0.04 units, keeping all other predictors constant. etc..

Regularization:-

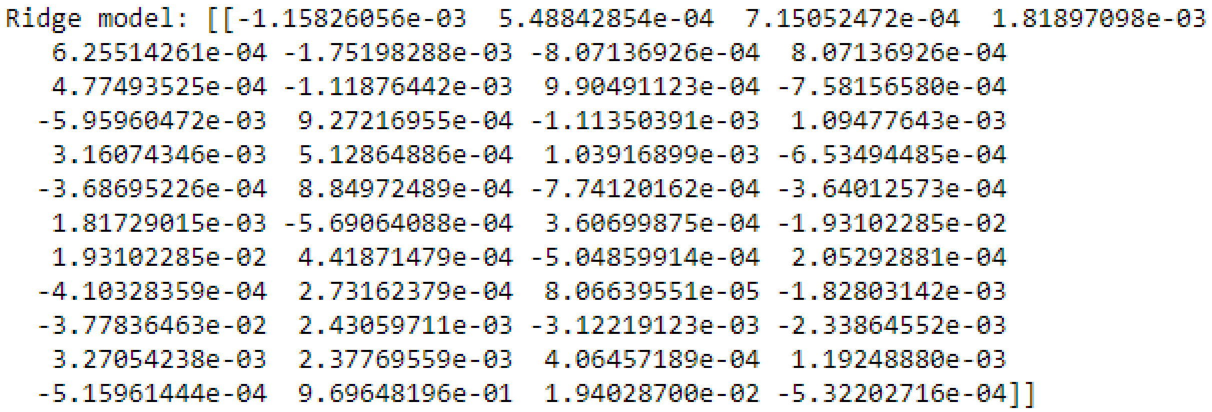
- It reduces the overfitting nature of the model. Even if the model works well, this is done in order to prevent the problem from occurring in the future.
- This is done by introducing more errors and making the model learn more.
- This will help the model to learn more. And as a result, even if more data is added in the later stage, the model will be able to process those without any issues.
- Now the model performance will increase and will be better than the unregularized model.

Types of Regularization:-

Ridge Regularization:-

- Ridge regression is a standard model tuning process used to analyse the data suffering from multicollinearity.
- 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

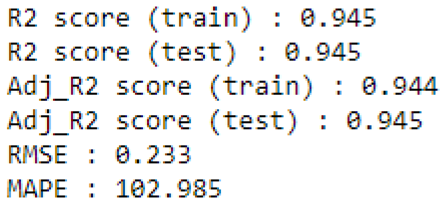
Let's create a regularized RIDGE model and note the coefficients



```
Ridge model: [[-1.15826056e-03  5.48842854e-04  7.15052472e-04  1.81897098e-03
   6.25514261e-04 -1.75198288e-03 -8.07136926e-04  8.07136926e-04
   4.77493525e-04 -1.11876442e-03  9.90491123e-04 -7.58156580e-04
  -5.95960472e-03  9.27216955e-04 -1.11350391e-03  1.09477643e-03
   3.16074346e-03  5.12864886e-04  1.03916899e-03 -6.53494485e-04
  -3.68695226e-04  8.84972489e-04 -7.74120162e-04 -3.64012573e-04
   1.81729015e-03 -5.69064088e-04  3.60699875e-04 -1.93102285e-02
   1.93102285e-02  4.41871479e-04 -5.04859914e-04  2.05292881e-04
  -4.10328359e-04  2.73162379e-04  8.06639551e-05 -1.82803142e-03
  -3.77836463e-02  2.43059711e-03 -3.12219123e-03 -2.33864552e-03
   3.27054238e-03  2.37769559e-03  4.06457189e-04  1.19248880e-03
  -5.15961444e-04  9.69648196e-01  1.94028700e-02 -5.32202716e-04]]
```

Image5.Ridge_Coefficient

Model Evaluation:-

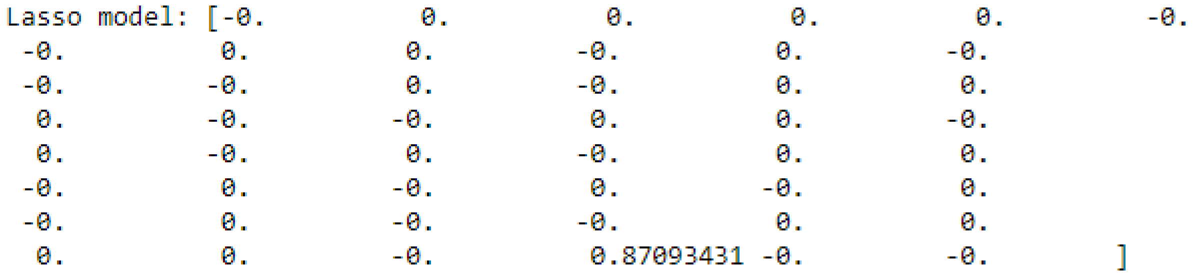


```
R2 score (train) : 0.945
R2 score (test) : 0.945
Adj_R2 score (train) : 0.944
Adj_R2 score (test) : 0.945
RMSE : 0.233
MAPE : 102.985
```

LASSO Regularization Model:

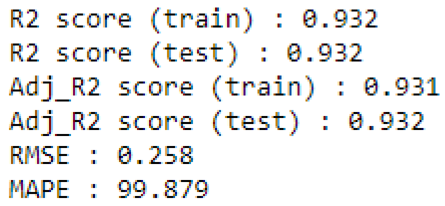
- The model is penalized for the sum of absolute values of the weights. Introduces a new hyper parameter, alpha, the coefficient to penalize weights.
- Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction.
- This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters)
- 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.

Let's Create a regularized LASSO model and note the coefficients



```
Lasso model: [-0.          0.          0.          0.          0.         -0.
 -0.          0.          0.         -0.          0.         -0.
 -0.         -0.          0.         -0.          0.          0.
  0.         -0.         -0.          0.          0.         -0.
  0.         -0.          0.         -0.          0.          0.
 -0.          0.         -0.          0.         -0.          0.
 -0.          0.         -0.         -0.          0.          0.
  0.          0.         -0.          0.87093431 -0.         -0.        ]
```

Model Evaluation:-



```
R2 score (train) : 0.932
R2 score (test) : 0.932
Adj_R2 score (train) : 0.931
Adj_R2 score (test) : 0.932
RMSE : 0.258
MAPE : 99.879
```

Key Observation:-

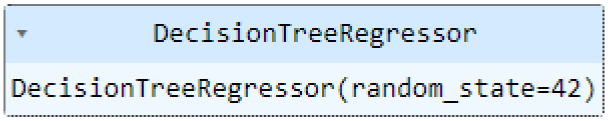
- Lasso regression usually works better under conditions where some predictors have high coefficients, and the rest have low coefficients.
- Ridge regression performs better when the result is a function of many predictors, all of which have coefficients of approximately the same size

Decision Tree Regressor:-

Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

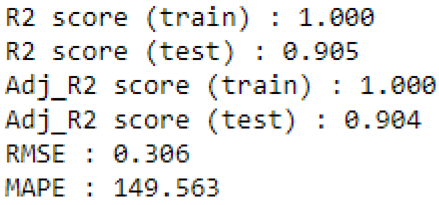
For continuous data prediction decision tree Regressor is used. It is used to fit a sine curve with addition noisy observation. As a result, it learns local linear regressions approximating the sine curve.

We will first fit the decision tree Regressor



```
▾        DecisionTreeRegressor
DecisionTreeRegressor(random_state=42)
```

Model Evaluation of Decision tree Regressor:-

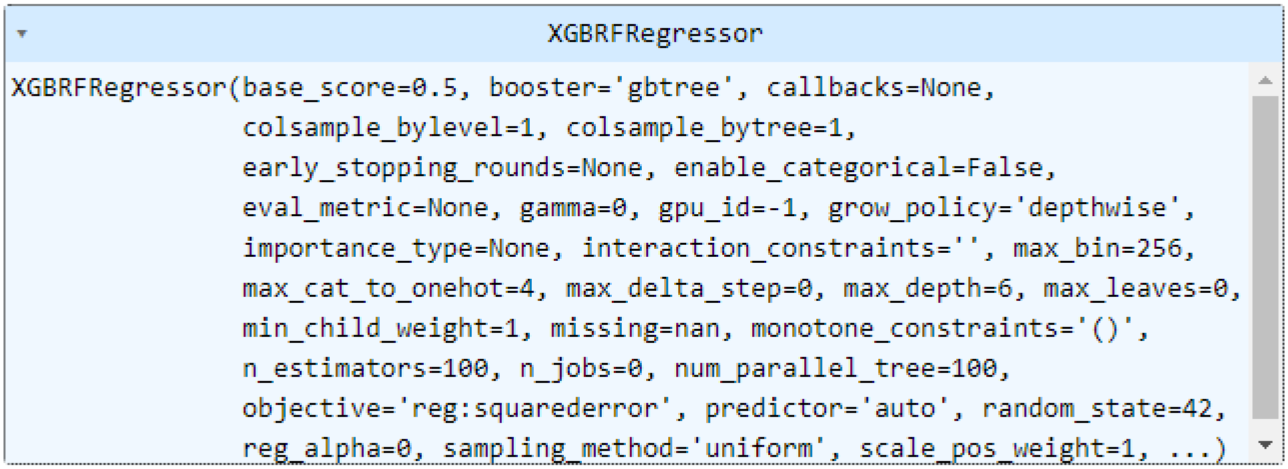


```
R2 score (train) : 1.000
R2 score (test) : 0.905
Adj_R2 score (train) : 1.000
Adj_R2 score (test) : 0.904
RMSE : 0.306
MAPE : 149.563
```

Here it can be seen that r2 score of train and test data which results in overfitting model as model is working fine on train data but not on test data. RMSE score is less but MAPE value is high

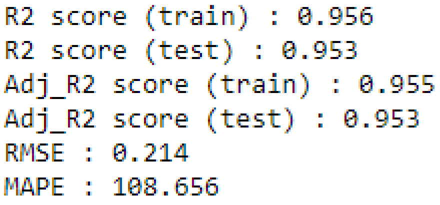
XGBoost Regressor:-

We will first fit the XGBoost Regressor



```
XGBRFRegressor(base_score=0.5, booster='gbtree', callbacks=None,
               colsample_bylevel=1, colsample_bytree=1,
               early_stopping_rounds=None, enable_categorical=False,
               eval_metric=None, gamma=0, gpu_id=-1, grow_policy='depthwise',
               importance_type=None, interaction_constraints='', max_bin=256,
               max_cat_to_onehot=4, max_delta_step=0, max_depth=6, max_leaves=0,
               min_child_weight=1, missing=nan, monotone_constraints='()',
               n_estimators=100, n_jobs=0, num_parallel_tree=100,
               objective='reg:squarederror', predictor='auto', random_state=42,
               reg_alpha=0, sampling_method='uniform', scale_pos_weight=1, ...)
```

Model Evaluation of XGBoost Regressor:-



```
R2 score (train) : 0.956
R2 score (test) : 0.953
Adj_R2 score (train) : 0.955
Adj_R2 score (test) : 0.953
RMSE : 0.214
MAPE : 108.656
```

Here it can be seen that r2 score of train and test data is almost same as it work on train and test data. RMSE and MAPE score is also less.

Effort to improve model performance.

Model Tuning: -

We tune the model to maximize model performances without overfitting and reduce the variance error in our model. We have to apply the appropriate Hyperparameter technique for our model

Types of error:-

Variance: - The variability of model prediction for a given data point which tells us spread of our data is called the variance of the model

Bias: - The bias is known as the difference between the prediction of the values by the ML model and the correct value. Being high in biasing gives a large error in training as well as testing data.

Underfitting: - A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data.

Overfitting: - A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance

Ensemble modelling, wherever applicable:-

It is very common that the individual model suffers from bias or variances and that's why we need the ensemble learning. Ensemble learning is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.

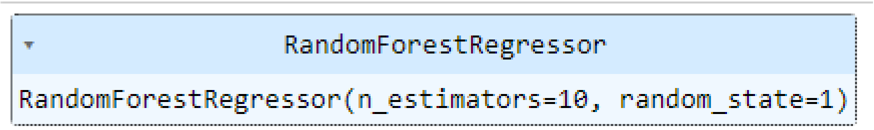
Two most popular ensemble methods are bagging and boosting.

Bagging: Training a bunch of individual models in a parallel way. Each model is trained by a random subset of the data
Boosting: Training a bunch of individual models in a sequential way. Each individual model learns from mistakes made by the previous model.

Ensemble Random Forest Regressor:-

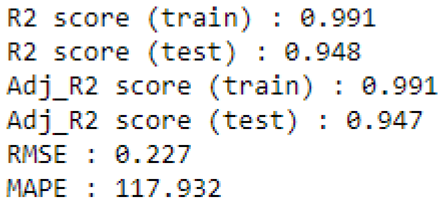
Random forest Regressor is an ensemble model using bagging as the ensemble method and decision tree as the individual model.

We will first fit the Random Forest Regressor



```
RandomForestRegressor
RandomForestRegressor(n_estimators=10, random_state=1)
```

Model Evaluation of Random Forest Regressor:-



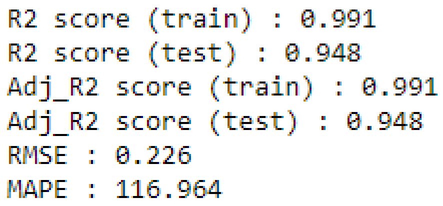
```
R2 score (train) : 0.991
R2 score (test) : 0.948
Adj_R2 score (train) : 0.991
Adj_R2 score (test) : 0.947
RMSE : 0.227
MAPE : 117.932
```

Here it can be seen that r2 score of train and test data which results in overfitting model as model is working fine on train data but not on test data. RMSE score is less but MAPE value is high.

Ensemble Learning – Bagging

We will first fit the Bagging Regressor

Model Evaluation of Bagging Regressor:-



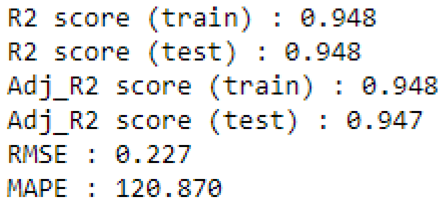
```
R2 score (train) : 0.991
R2 score (test) : 0.948
Adj_R2 score (train) : 0.991
Adj_R2 score (test) : 0.948
RMSE : 0.226
MAPE : 116.964
```

Here it can be seen that r2 score of train and test data which results in overfitting model as model is working fine on train data but not on test data. RMSE score is less but MAPE value is high.

Ensemble Learning – AdaBoosting

We will first fit the AdaBoosting Regressor

Model Evaluation of AdaBoosting Regressor:-



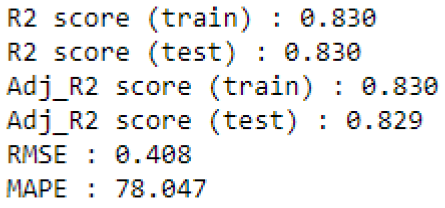
```
R2 score (train) : 0.948
R2 score (test) : 0.948
Adj_R2 score (train) : 0.948
Adj_R2 score (test) : 0.947
RMSE : 0.227
MAPE : 120.870
```

Here it can be seen that r2 score of train and test data same. RMSE score is less but MAPE value is high.

Ensemble Learning – GradientBoost

We will first fit the **GradientBoost** Regressor

Model Evaluation of GradientBoost Regressor:-



```
R2 score (train) : 0.830
R2 score (test) : 0.830
Adj_R2 score (train) : 0.830
Adj_R2 score (test) : 0.829
RMSE : 0.408
MAPE : 78.047
```

Here it can be seen that r2 score of train and test data same but less compared to other models. RMSE score is high compared to other models but MAPE value is less.

5. Model Validation

Evaluation helps to judge the performance of any machine learning model that would provide best results to our test data.
Fundamentally three types of evaluation metrics are used to evaluate linear regression model.
1. R-square/ Adjusted R-square Score
2. Mean Square Error (MSE) / Root Mean Square Error (RMSE)
3. Mean Absolute Error (MAE) / Mean Absolute percentage error (MAPE)

1. **R-square/ Adjusted R-square Score:-**

   R-squared and Adjusted R-squared are two evaluation metrics which is used to evaluate regression problems.

   - R-squared (R2) is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. R-squared explains to what extent the variance of one variable explains the variance of the second variable.

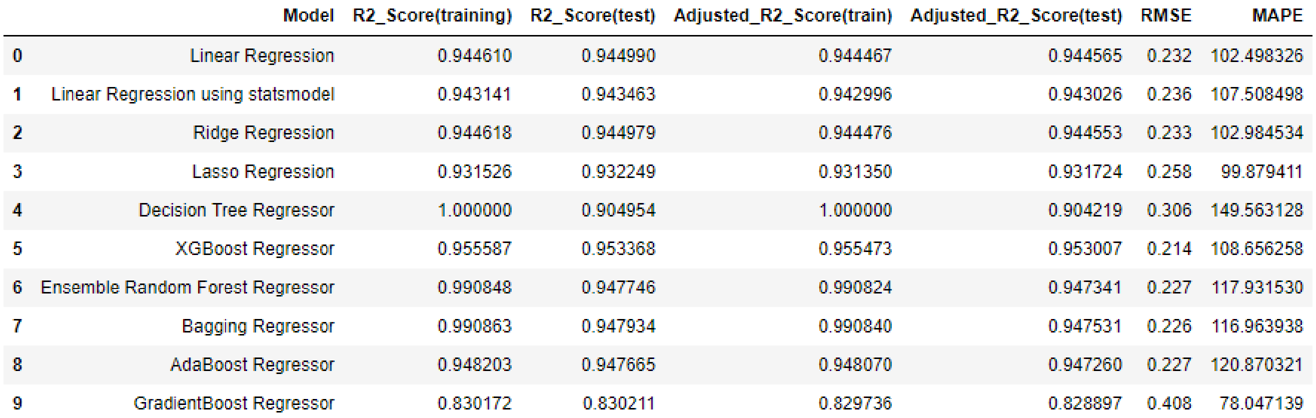
   - Adjusted R-squared is a modified version of R-squared that has been adjusted for the number of predictors in the model. The adjusted R-squared increases when the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected. Typically, the adjusted R-squared is positive, not negative. It is always lower than the R-squared. Adjusted R-squared can provide a more precise view of that correlation by also taking into account how many independent variables are added to a particular model.

2. **RMSE (Root mean square error)**
   It is one of the most commonly used measures for evaluating the quality of predictions. It shows how far predictions fall from measured true values using Euclidean distance.

3. **MAPE (Mean absolute percentage error)**
   It is the mean percentage error difference between the predicted and actual values. The model having less MAPE score is the best model. Less MAPE value means prediction made by the model are close to the real value.

So in our case we will use MAPE value and Adjusted R-squared to evaluate the model .We are using Adjusted R-squared as R2 may have noise in it as it starts increasing.

Evaluation table having all the models.



	Model	R2_Score(training)	R2_Score(test)	Adjusted_R2_Score(train)	Adjusted_R2_Score(test)	RMSE	MAPE
0	Linear Regression	0.944610	0.944990	0.944467	0.944565	0.232	102.498326
1	Linear Regression using statsmodel	0.943141	0.943463	0.942996	0.943026	0.236	107.508498
2	Ridge Regression	0.944618	0.944979	0.944476	0.944553	0.233	102.984534
3	Lasso Regression	0.931526	0.932249	0.931350	0.931724	0.258	99.879411
4	Decision Tree Regressor	1.000000	0.904954	1.000000	0.904219	0.306	149.563128
5	XGBoost Regressor	0.955587	0.953368	0.955473	0.953007	0.214	108.656258
6	Ensemble Random Forest Regressor	0.990848	0.947746	0.990824	0.947341	0.227	117.931530
7	Bagging Regressor	0.990863	0.947934	0.990840	0.947531	0.226	116.963938
8	AdaBoost Regressor	0.948203	0.947665	0.948070	0.947260	0.227	120.870321
9	GradientBoost Regressor	0.830172	0.830211	0.829736	0.828897	0.408	78.047139

Table7.Model_Evaluation

Out of all Model GradientBoost model has less MAPE score but it has very less Adjusted R-squared score compared to other model. And XGBoost Model has highest adjusted r square value but it also have high MAPE value.
Hence Linear Regression is the best model to predict the insurance cost of the customers.

6. Final interpretation / recommendation

Interpretation:-

- Dataset has 25000 rows and 24 Columns / variables
- Data is collected between the **age** groups of 16 to 74
- **Occupation** of the customers are ranging from students, Business and Salaried
  Student:-10169
  Business:-10020
  Salaried:-4811
- **Alcohol** intake values ranging from No, Rare and Daily.

Rare:-13752

No:-8541

Daily:-2707

- **Weight** is ranging from 52kgs – 96kgs.
- **Doing Exercise** values ranging from Daily, Moderate and Extreme
- **Smoking status** ranging from never smoked, unknown, formerly smoked, smokes

  Never smoked: - 9249

  Unknown: - 7555

  Formerly smoked: - 4329

  Smokes: - 3867
- **Insurance Cost** (Target Variable) is considered as Premium per Year; Insurance Cost is ranging from Rs 2468 to Rs 67870
- **Applicant_id** column is irrelevant in the above context and hence can be ignored.
- Mean **BMI** = 31 and Max = 100
- Mean **age** = 44 and Max = 74
- 16422 are **Male** (65%) and 8578 (35%) are **Female**

From the above insight we can say that

- Number of male customers are higher than female customers.
- Most of the customers who are taking health insurance are non-smoker as compared to who smokes.
- Students are the one who are taking more number of health insurance as compared to salaried customer.

Insights from Univariate analysis:-

- Most of the customer who have taken the health insurance visited the doctor 2, 3 and 4 times in last one year
- Most the number of customers doesn't have any past heart disease history.
- The customers insured with health insurance doesn't involved with adventure sports like climbing, trekking, cycling etc.
- Most of the customers haven't done the regular heath checkup in last one year. And there are very less number of customers who went for regular checkup.
- The weight change in last one year also affect the number of customers who are insured with us. As the weight changes the chances of getting ill which will also affect the insurance cost.

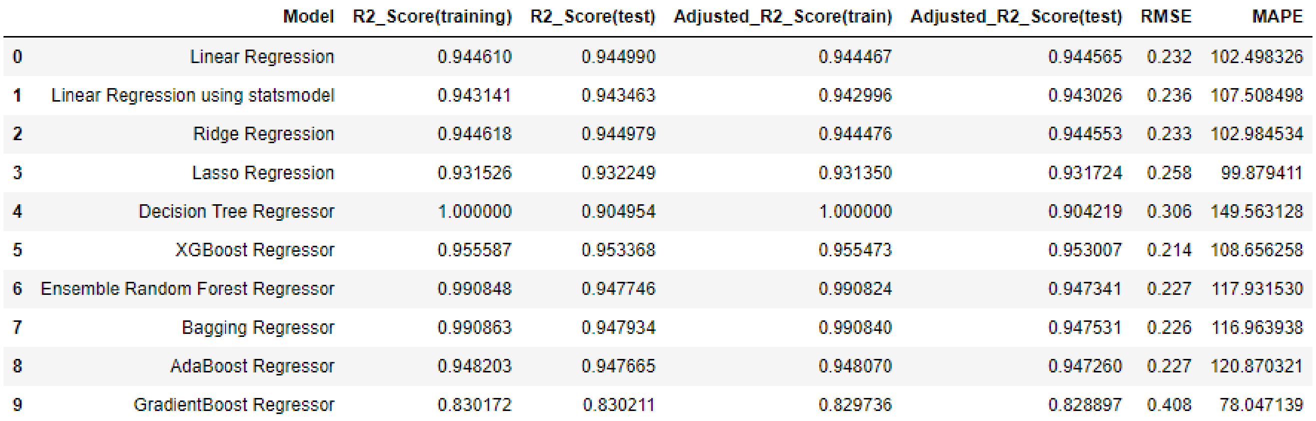
Insights from Bivariate analysis:-

- Among the categorical variable, none of the variable is showing a strong predictor as compared together. The range and median of all unique values of all variable with respect to insurance_cost are somewhat same.
- Only weight is forming a straight line when plotted, which means weight is showing a positive correlation. We can also say that it is a strong predictor of insurance cost.
- Years of insurance with us vs insurance cost- the customers who are insured with the company for more number of years. We can't see any variation in the plot. So it doesn't have correlation with insurance cost.
- Regular checkup lasy year vs insurance cost- the customers who haven't went for regular checkup have maximum health insurance cost. And customers who went for regular checkup 5 times have less insurance. Which means it has strong correlation with insurance cost. And it is a strong predictor.
- Heart disease history vs insurance cost - The insurance cost of customer who have heart disease history and customers who don't have heart disease is almost same.
- Visited doctor last year vs insurance cost – There are almost same insurance cost of all customer who have visited doctor in last one year. But there is slight difference that customer who visited doctor 13 times have min health insurance cost. And the customer who visited doctor 10 times have highest insurance cost
- Weight change in last one year vs insurance cost – the customer whose weight change 2kg in last one year have max insurance cost which means it is a strong predictor of insurance cost.

From the above bivariate insight we can say that years of insurance with us, age, weight, Regular checkup lasy year, and weight change in last one year are the strong predictors of insurance cost.

After preprocessing the data we build the model. There are so many machine learning models available but some fits better to some problem while some fits with other problems. So to make this decision performed several model and evaluate the model on the basis of adjusted r square score and MAPE score to do the prediction of the insurance cost.

Model evaluation of all types of models.

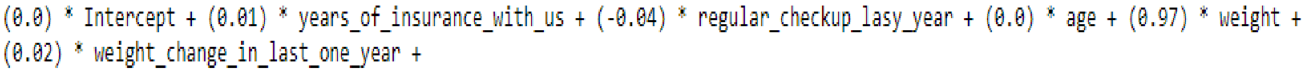


	Model	R2_Score(training)	R2_Score(test)	Adjusted_R2_Score(train)	Adjusted_R2_Score(test)	RMSE	MAPE
0	Linear Regression	0.944610	0.944990	0.944467	0.944565	0.232	102.498326
1	Linear Regression using statsmodel	0.943141	0.943463	0.942996	0.943026	0.236	107.508498
2	Ridge Regression	0.944618	0.944979	0.944476	0.944553	0.233	102.984534
3	Lasso Regression	0.931526	0.932249	0.931350	0.931724	0.258	99.879411
4	Decision Tree Regressor	1.000000	0.904954	1.000000	0.904219	0.306	149.563128
5	XGBoost Regressor	0.955587	0.953368	0.955473	0.953007	0.214	108.656258
6	Ensemble Random Forest Regressor	0.990848	0.947746	0.990824	0.947341	0.227	117.931530
7	Bagging Regressor	0.990863	0.947934	0.990840	0.947531	0.226	116.963938
8	AdaBoost Regressor	0.948203	0.947665	0.948070	0.947260	0.227	120.870321
9	GradientBoost Regressor	0.830172	0.830211	0.829736	0.828897	0.408	78.047139

Table8.Final_Model_Evaluation

Out of all Model GradientBoost model has less MAPE score but it has very less Adjusted R-squared score compared to other model. And XGBoost Model has highest adjusted r square value but it also have high MAPE value.
Hence Linear Regression is the best model to predict the insurance cost of the customers.

The final Linear Regression equation is



```
(0.0) * Intercept + (0.01) * years_of_insurance_with_us + (-0.04) * regular_checkup_lasy_year + (0.0) * age + (0.97) * weight +
(0.02) * weight_change_in_last_one_year +
```

Insurance_cost= b0 + b1 * years_of_insurance_with_us + b2 *regular_checkup_lasy_year + b3 * age + b4 * weight + b5 * weight_change_in_one_year+

Insurance_cost= (0.0) * Intercept + (0.01) * years_of_insurance_with_us + (-0.04) *regular_checkup_lasy_year + (0.0) * age + (0.97) * weight + (0.02) * weight_change_in_one_year+

When **weight** increases by 1 unit, insurance_cost increases by 0.97 units, keeping all other predictors constant When weight_change_in_one_year increases by 1 unit, insurance_cost increases by 0.02 units, keeping all other predictors constant etc.

There are also some negative co-efficient values, for instance, regular_checkup_lasy_year has its corresponding co-efficient as -0.04. This implies, when the customer go for regular checkup in years, the insurance_cost decreases by 0.04 units, keeping all other predictors constant. etc.

Recommendations:-

- Insurance company can tie up with hospitals or medical institution to provide awareness about various diseases.
- The insurance company should conduct medical awareness program in various corporate industries for the employees as per our analysis we found that salaried customers are taking less health insurance compared to others.
- With the help of advertisements insurance companies can educate customers to avoid junk food and do some daily to live the healthy lifestyle.
- Now the era has been changed, most of the female customers are working they can't focus on their diet and healthy habits. We can create an awareness program among them like "how" they can balance their life with some proper diet and exercise by taking more protein and balance sugar.
- As per our analysis younger generation don't focus more on their health and investments, hence insurance company can conduct awareness campaign to reduce any disease risk in early stage.

These days medical industries are very expensive which is significantly impact emotion and financial condition of an individual. So Health Insurance Company can help to reduce the future medical risk.

Reference:-

https://www.marketresearch.com/Netscribes-India-Pvt-Ltd-v3676/Healthcare-India-32528948/

https://www.ibef.org/industry/insurance-sector-india#:~:text=15%2C906.71%20crore%20(US%24%202.14%20billion,premium%20underwritten%20services%20in%20FY20

END