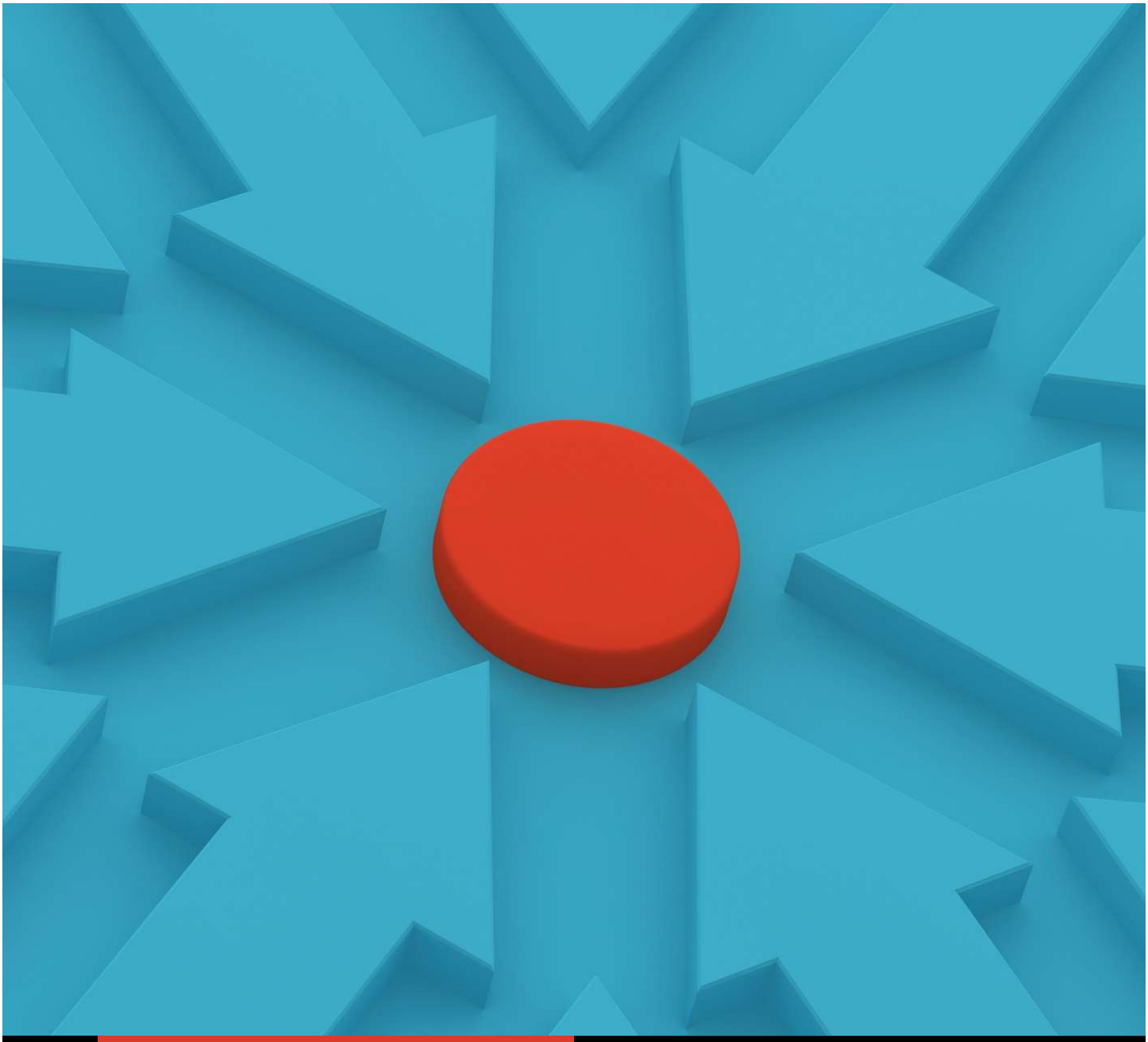


CAPSTONE PROJECT FINAL REPORT – DSBA

MOHIT VERMA

---------- INDEX ----------

* Data Dictionary (HealthCare Life Insurance Data):-

- **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_co mpany: 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**

1. Introduction:-

- Brief introduction about the problem statement and the need of solving it.

---

a) Defining problem statement:-

- The goal of this project is to build a model that predicts the best insurance cost for an individual based on their health and lifestyle habits. This involves analyzing factors like age, BMI, smoking habits, physical activity, and medical history to estimate personalized insurance premiums. The aim is to make insurance pricing fair and accurate, ensuring that people pay based on their actual health risks. For insurance companies, this helps create better pricing strategies and improves customer trust by offering policies tailored to each person. It also encourages people to adopt healthier habits by showing how their lifestyle affects their insurance costs. Overall, this approach makes insurance more accessible, helps people manage their health better, and creates a win-win for both customers and insurers.

- The data provided here it's related to users as with their medical related history where we have many features like 'years_of_insurance_with_us', 'age', 'gender', 'smoking_status' on the basis of that we have to predict insurance cost by using some Machine Learning Algorithm.

b) Need of the study/project:-

- We are working on this project just to study and address the challenges in accurately determining insurance costs that are fair and reflective of an individual's health and lifestyle. Traditional methods of pricing often rely on

generalized assumptions, which can lead to overcharging or undercharging certain individuals. This can result in customer dissatisfaction, reduced accessibility to insurance, and financial inefficiencies for insurance providers. By developing a model that uses health and habit-related parameters to predict insurance costs, the study aims to ensure a data-driven, transparent, and personalized approach to pricing.
This project is also crucial for promoting financial inclusion by making insurance more affordable and accessible to people from diverse backgrounds. Additionally, it can raise awareness about the impact of health and lifestyle choices on long-term costs, encouraging healthier behaviors. For insurance companies, the study enables better risk assessment, enhances customer trust, and supports the development of innovative and sustainable insurance products, ultimately benefiting both individuals and the industry.

c) Understanding business/social opportunity:-

- Predicting healthcare life insurance costs can benefit both companies and society. For businesses, it helps insurance companies set fair and accurate premiums based on individual health risks. This ensures they don't lose money by underestimating risks or lose customers by overcharging. It also allows companies to create personalized policies, improve their financial planning, and build trust with customers.

- For society, it makes life insurance more affordable and accessible, especially for people who might otherwise struggle to get coverage. It also encourages healthier lifestyles by identifying common health risks, helping

people take preventive actions. Overall, this improves the financial security and well-being of individuals and their families while supporting a stronger relationship between insurers and healthcare providers. It is going to help people to get better insurance as per their usage and coverage also it'll

help companies to provide better service to user as per their need and also it'll help companies to decide insurance cost on the basis of usage.

- Also this problem is related to Insurance Cost Prediction. So it'll be related to Regression problem. We are going to use few Machine Learning algorithm in order to predict it cost.

2. EDA and Business Implication:-

- Uni-variate / Bi-variate / Multi-variate analysis to understand relationship b/w variables. How your analysis is impacting the business? ……..
- Both visual and non-visual understanding of the data …….

---

a) Understanding how data was collected in terms of time, frequency and methodology:-

- So in this Project if we are talking about Data Collection process. We have already dataset available which was along with Business problem there.

b) Visual inspection of data (rows, columns, descriptive details):-

- Once we imported this dataset we get to know that we have '25000' rows and '24' columns in our dataset.

- As we have total '24' columns available where we have '16 Numerical' Column along with '14 int' and '2 float' data type. Also we have '8 Categorical' columns.

- If we are talking about available columns in our dataset which is already declared above in Data Dictionary part it is more related to applicants daily routine activity like 'smoking', 'cholesterol_level' etc. Below is the snap related to statistical summary of the available column in the dataset.

	applicant_id	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
count	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000	25000.000000
mean	17499.500000	4.089040	0.773680	0.081720	3.104200	5215.889320	44.918320	0.054640	0.098160
std	7217.022701	2.606612	1.199449	0.273943	1.141663	1053.179748	16.107492	0.227281	0.297537
min	5000.000000	0.000000	0.000000	0.000000	0.000000	2034.000000	16.000000	0.000000	0.000000
25%	11249.750000	2.000000	0.000000	0.000000	2.000000	4543.000000	31.000000	0.000000	0.000000
50%	17499.500000	4.000000	0.000000	0.000000	3.000000	5089.000000	45.000000	0.000000	0.000000
75%	23749.250000	6.000000	1.000000	0.000000	4.000000	5730.000000	59.000000	0.000000	0.000000
max	29999.000000	8.000000	5.000000	1.000000	12.000000	11255.000000	74.000000	1.000000	1.000000

- If we are talking about insurance duration with us we can see that we have applicants who has insurance with us max around 8 years.

- Also We can see that there's maximum 5 applicants who has done regular checkup last year.

- Apart from this we can also see that there's numbers of applicants or user who walk maxmimum 'daily_avg_steps' around '11,255 steps' and minimum '2034 steps' .

- One more thing in our dataset we have applicant who is maximum 74 and 16 years old.

c) Understanding of attributes (variable info, renaming if required):-

```
<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
```

- In above picture we can see information related to dataset and available columns there. As I mentioned above and we can see that there's we have 2 types numerical columns and categorical columns available which is in 'Object' data type.

- Also we found one discrepancy in the columns name available in our dataset like 'regular_checkup_lasy_year' and 'heart_decs_history'. So we are going to rename those in order to get proper understanding.

```
<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_last_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_disease_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
```

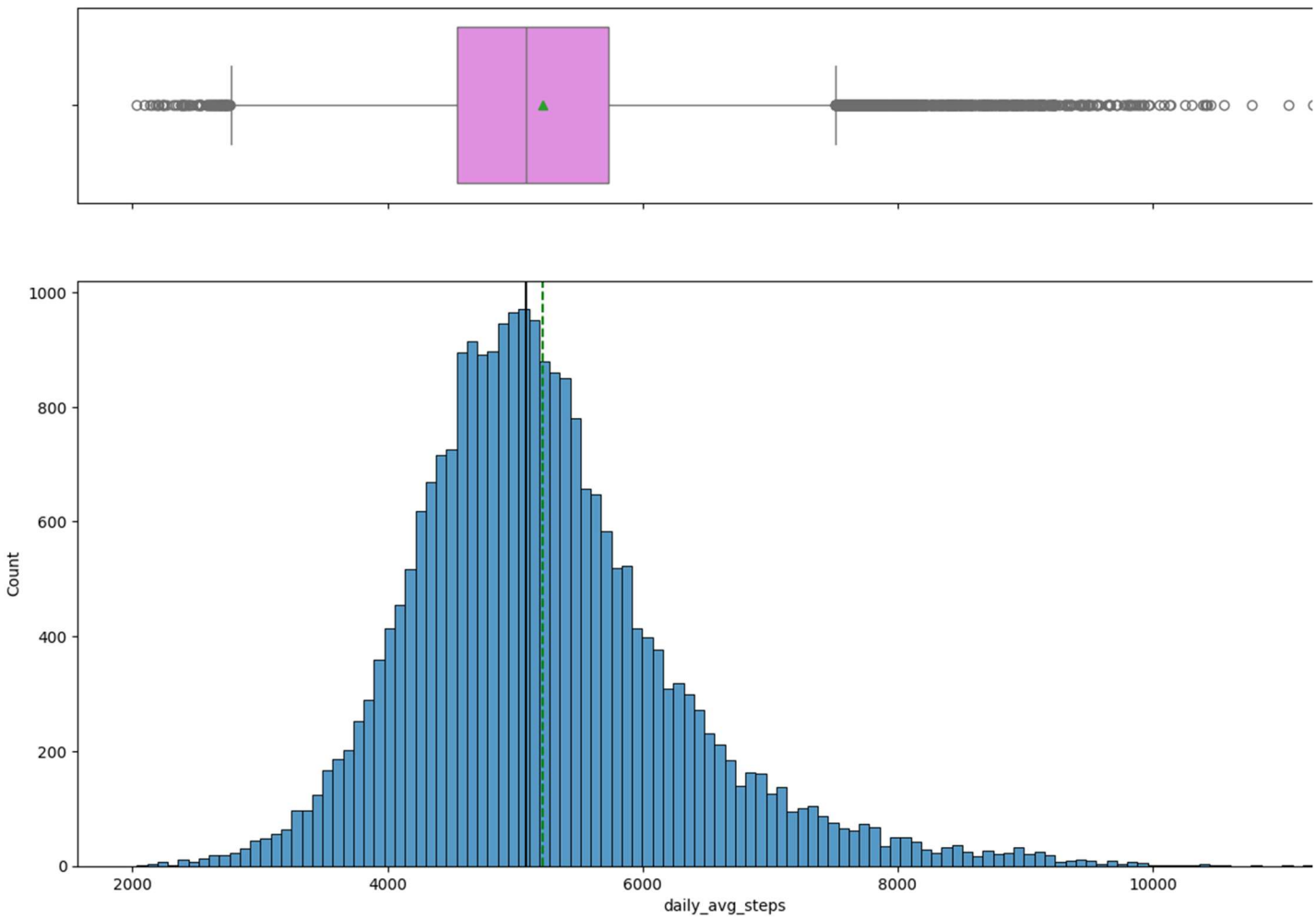
- As we can see that those columns name got changed and now it is correctly spelled.

- Also we found that there's no duplicate record available in our dataset.

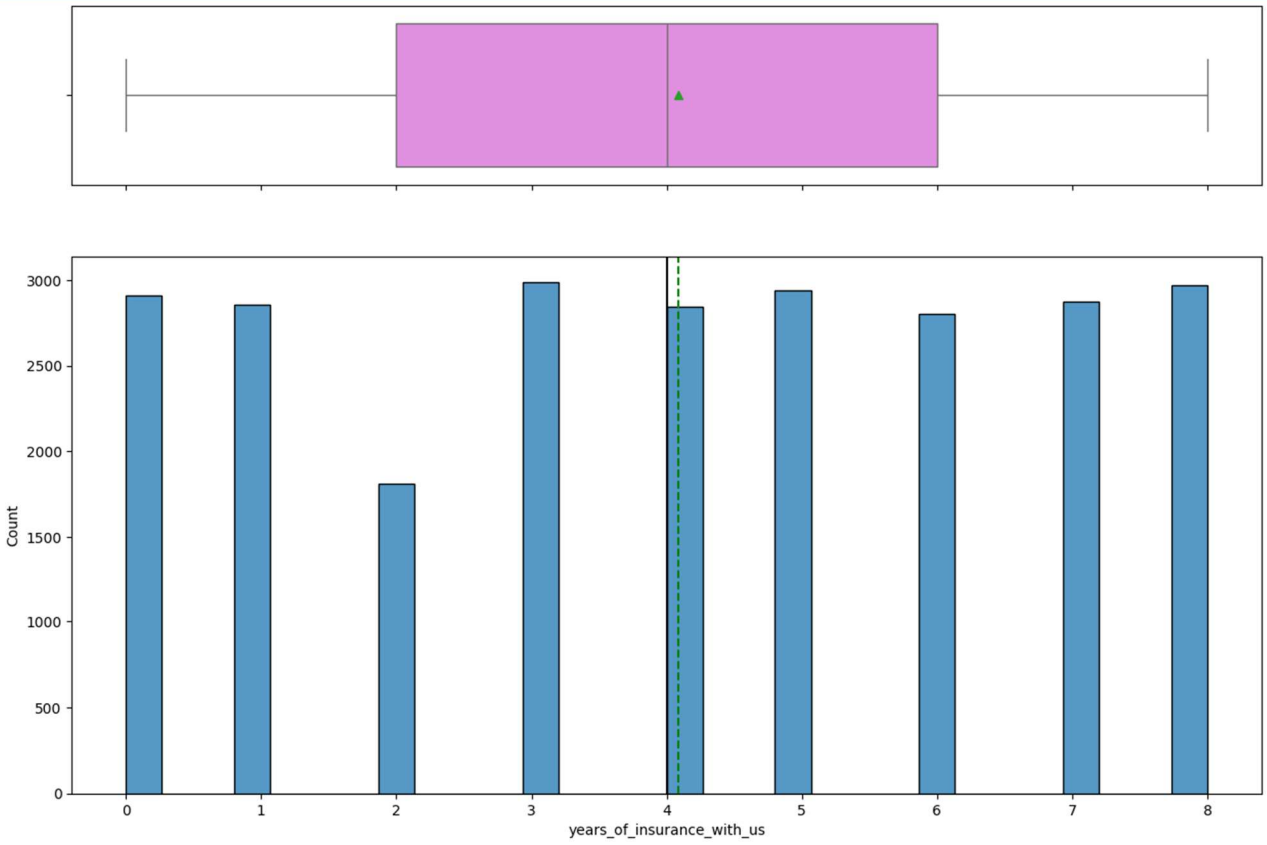
```
0
```

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

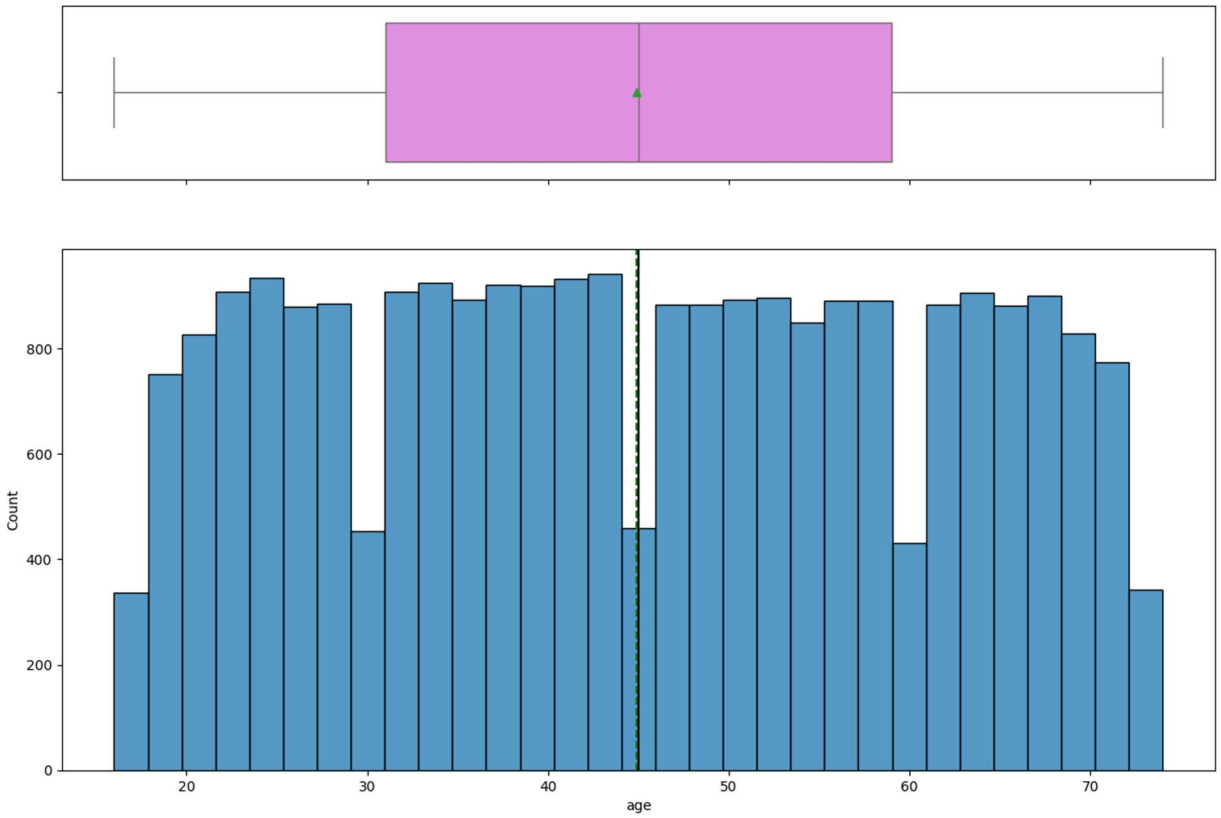
daily_avg_steps



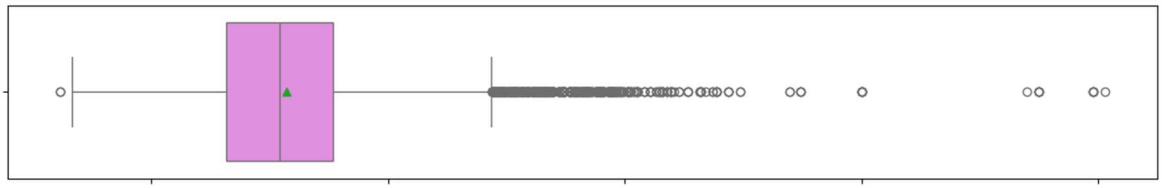
Years_of_insurance_with_us

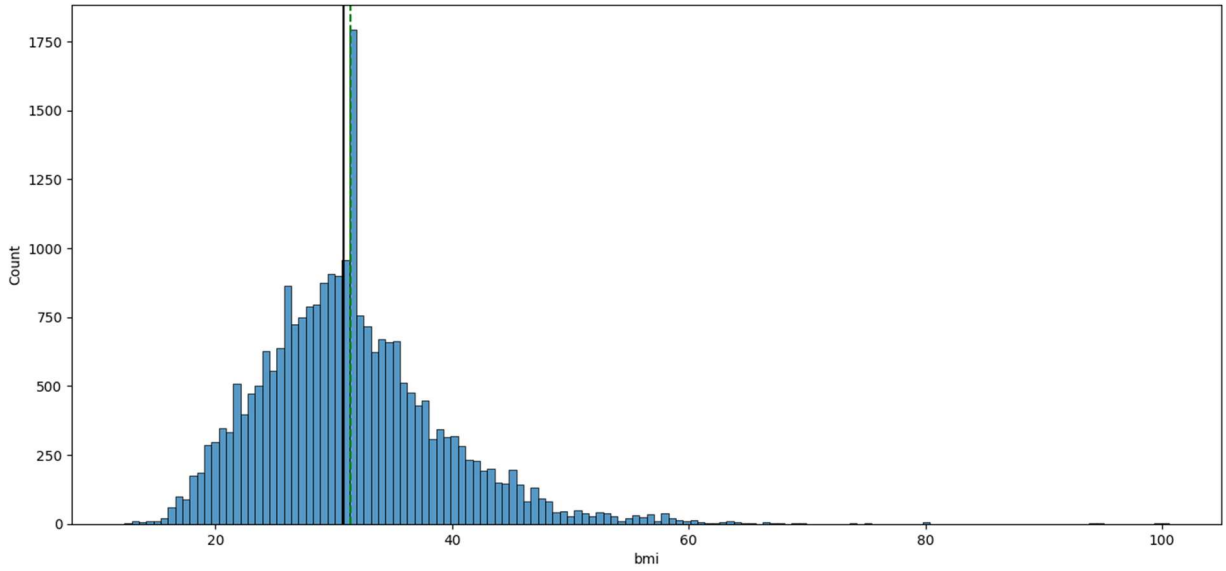

3000
2500
2000
1500
1000
500
0
Count
0
1
2
3
4
5
6
7
8
years_of_insurance_with_us

Age

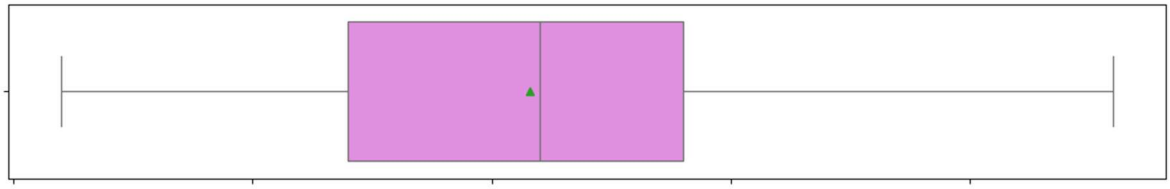

800
600
400
200
0
Count
20
30
40
50
60
70
age

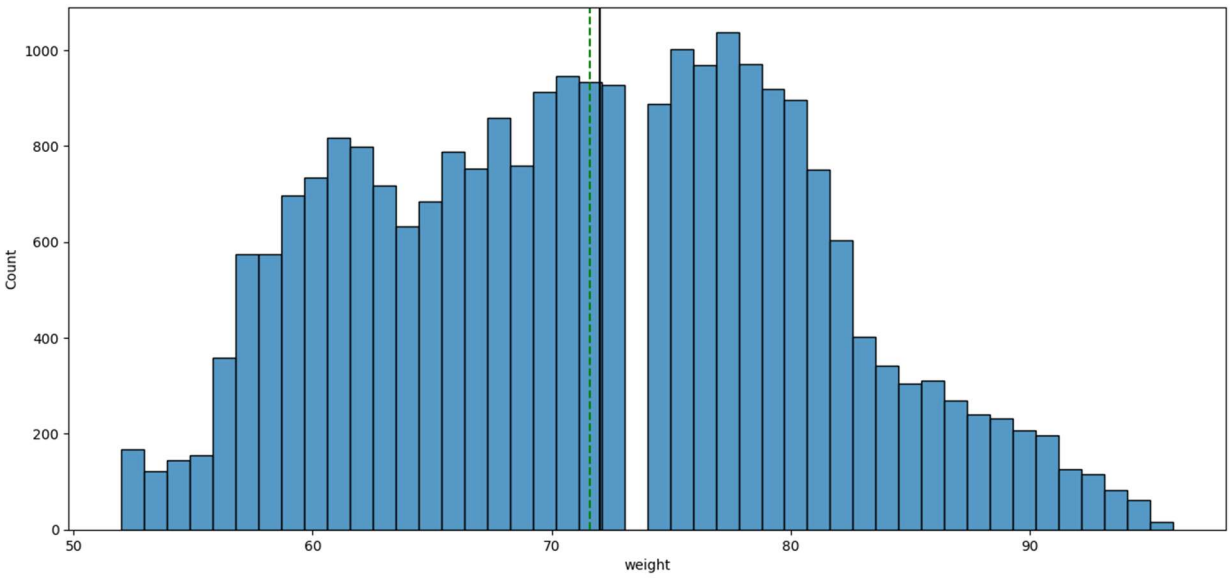
Bmi



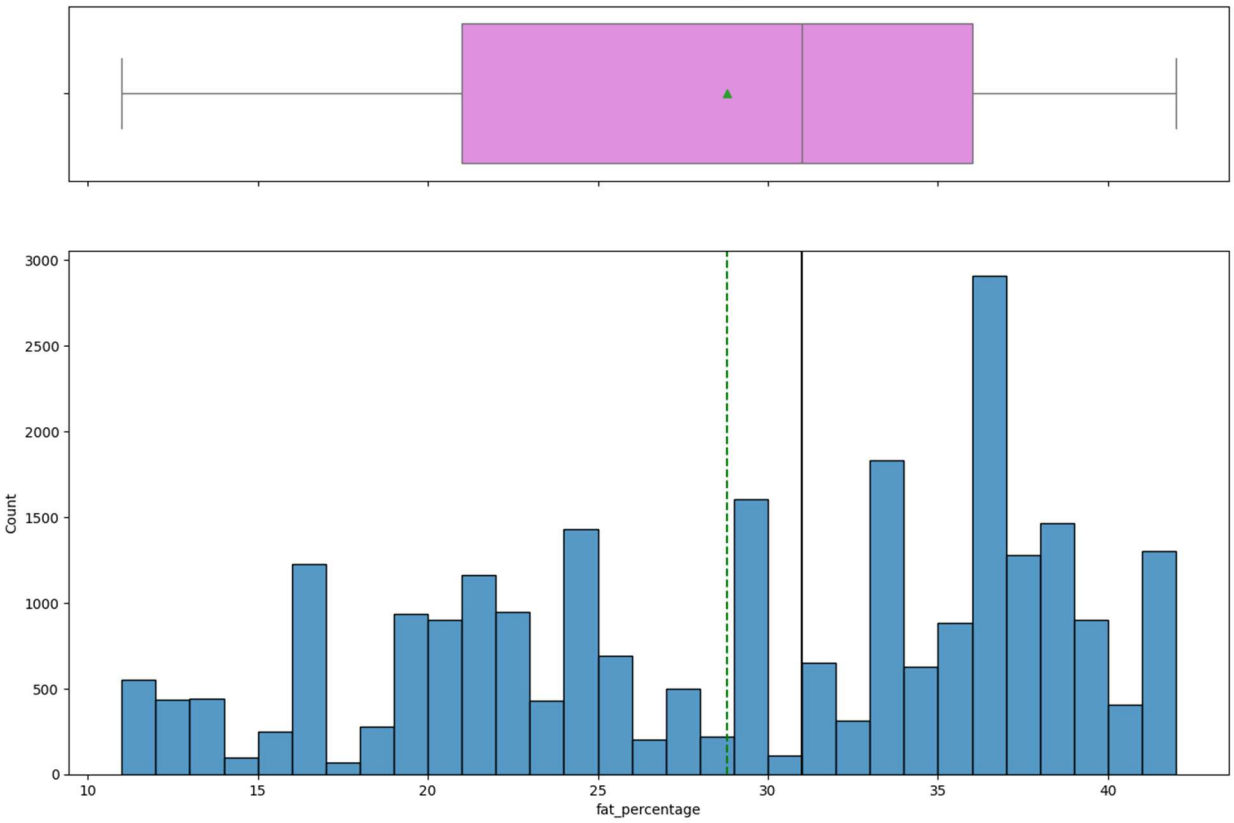

1750
1500
1250
1000
750
500
250
0
Count
20
40
60
80
100
bmi

Weight

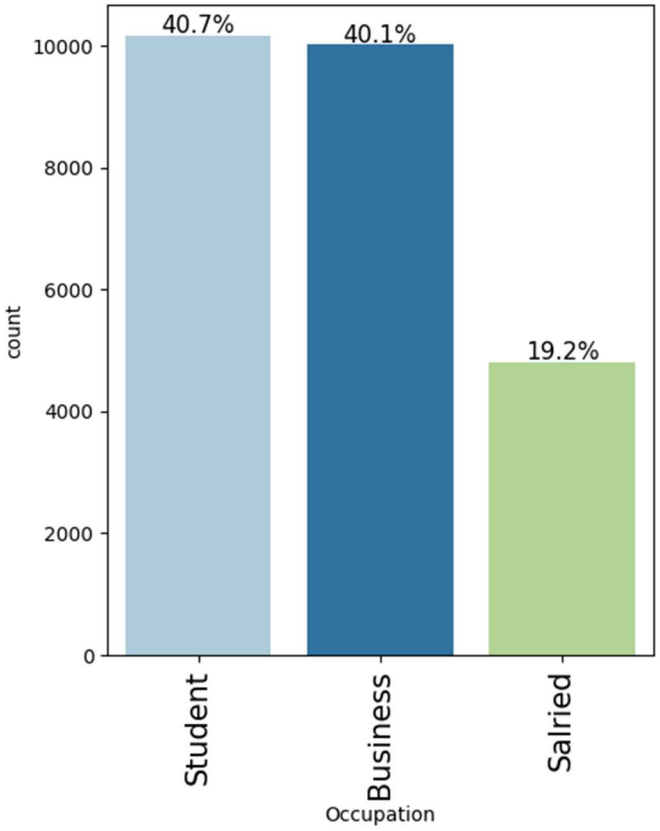



1000
800
600
400
200
0
Count
50
60
70
80
90
weight

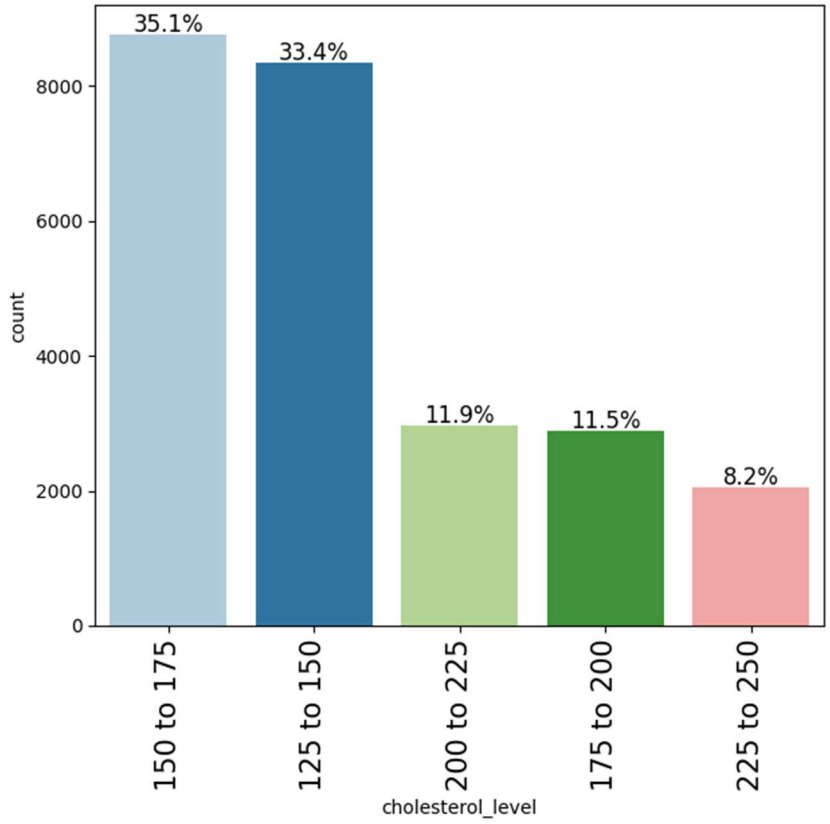
Fat_percentage



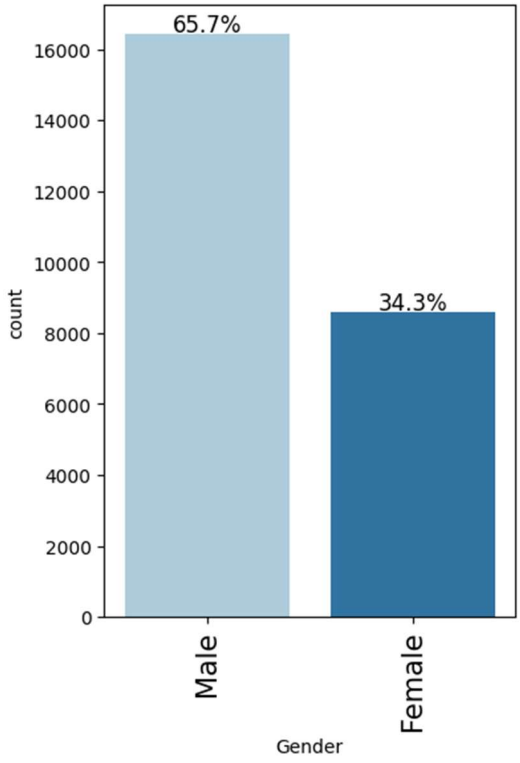
Occupation



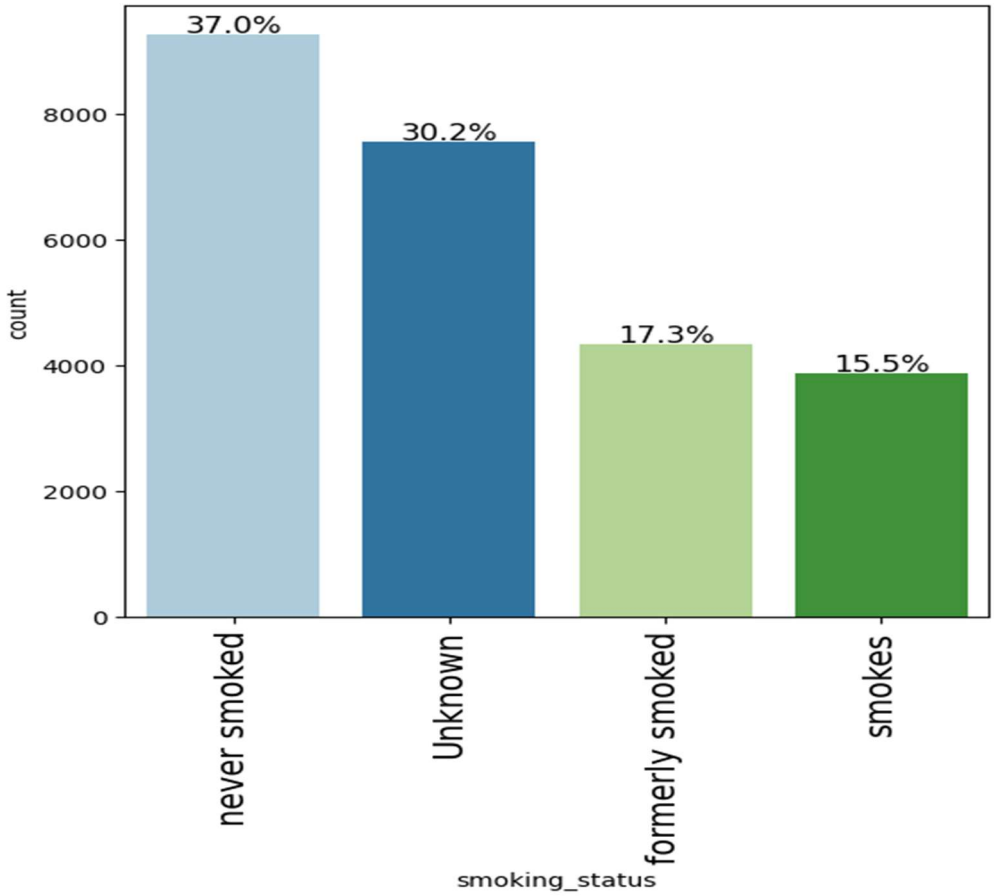
Cholesterol_level



Gender



Smoking_status



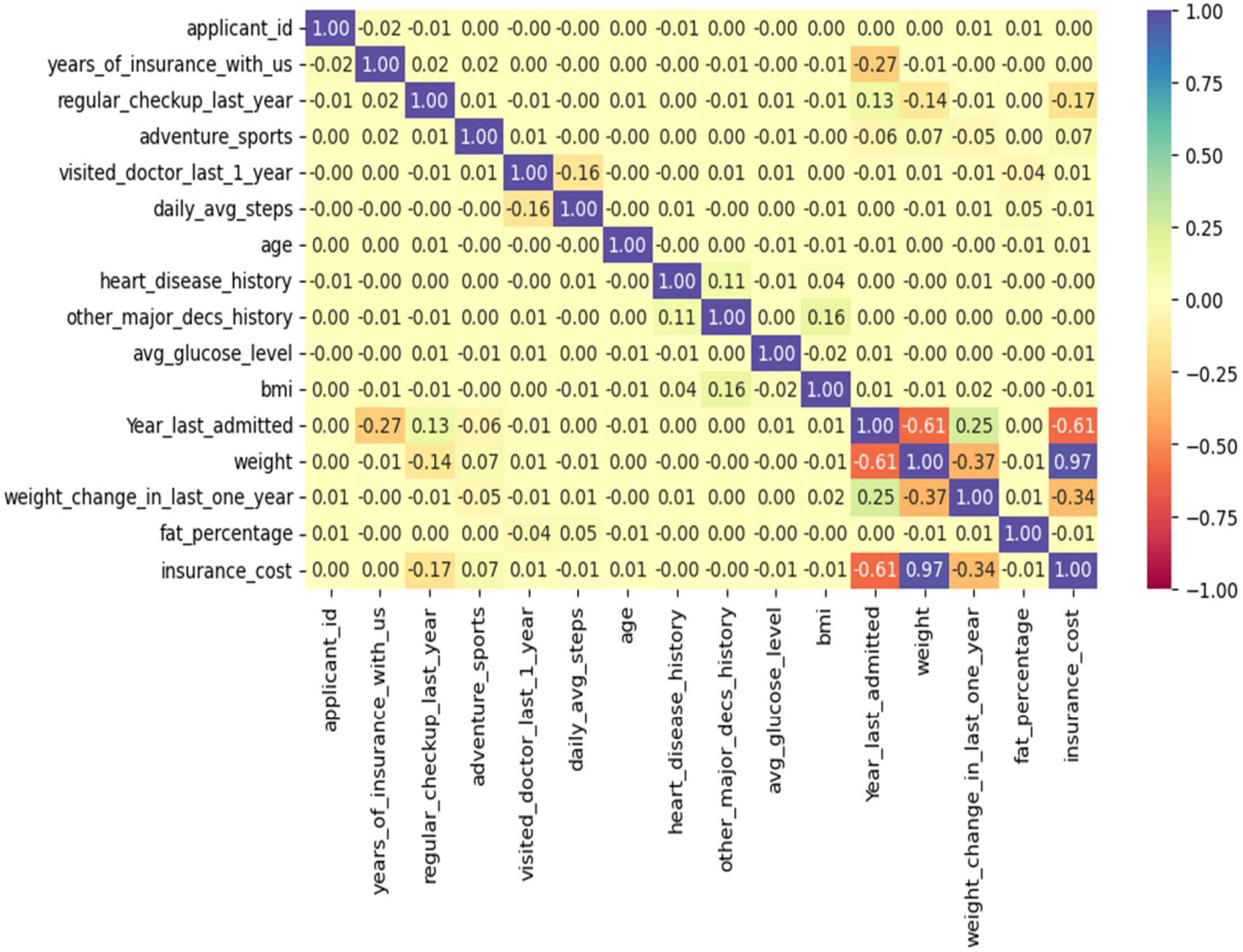
-- Observation --

- We can see by the above Boxplot diagram that we have many outliers available in columns ‘daily_avg_steps’ and ‘bmi’ which we are going to treat later.
- Also we have ‘bmi’ column which is Right Skewed apart from that we can see that ‘daily_avg_steps’ also little bit Right Skewed.
- Apart from that we can get to know that we have many applicants are available between the age of 16 to 74.
- Also we have maximum number of insurance applicants who are students which is around ’40.7%’ and less number of applicants whose occupation as a salaried person and ratio is around 19.2%.
- Also if we are checking in our dataset we have maximum number of people around 35.1% who has cholesterol_level between ‘150 to 175’ and 8.2% people who has cholesterol_level around ‘225 to 250’.

- In this dataset we have mostly “Male” candidate available in employees which is around 65.7% and “Female” around 34.3%.
- Also if we are checking the data distribution for the applicants we have many records available around ‘37.0%’ where people ‘never smoked’ and ’15.5%’ of people who generally smokes it’s lesser as compare to people who doesn’t smoke which is quite good.

b) Bivariate analysis (relationship between different variables, correlations):-

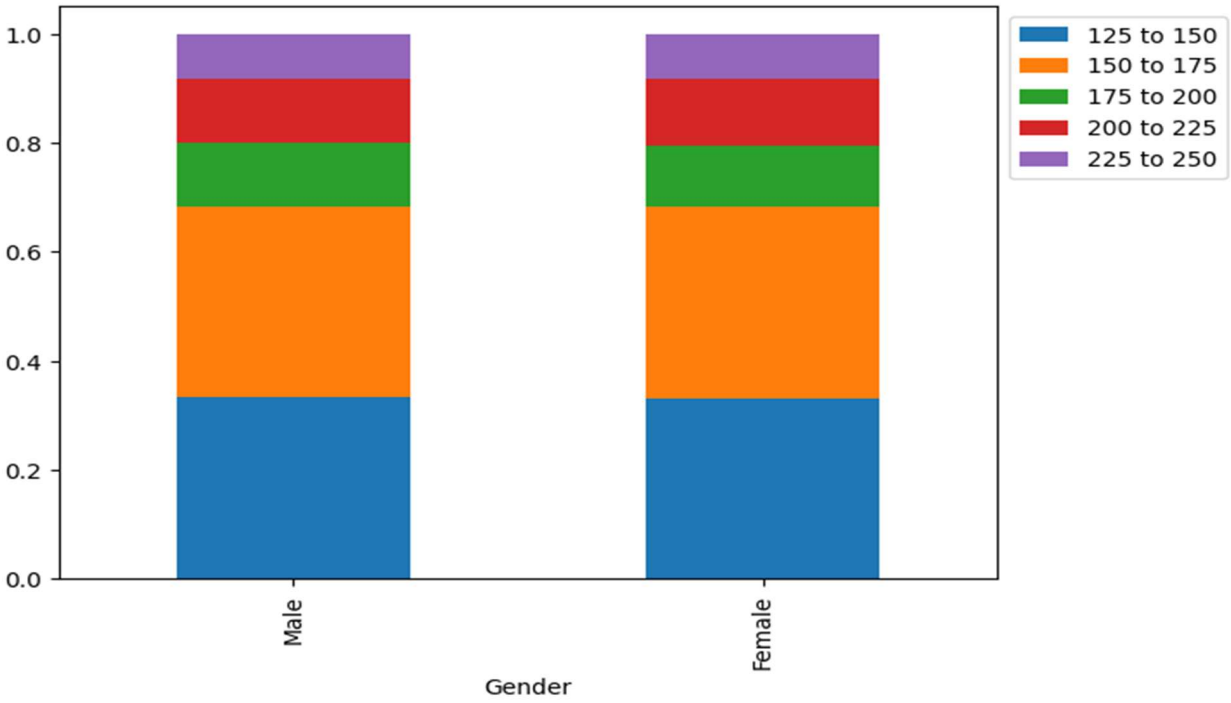
Creating a Heatmap to check relationship between each other



Gender wise ‘Cholesterol_level’

Gender	125 to 150	150 to 175	175 to 200	200 to 225	225 to 250
All	8339	8763	2881	2963	2054
Male	5498	5742	1925	1905	1352
Female	2841	3021	956	1058	702

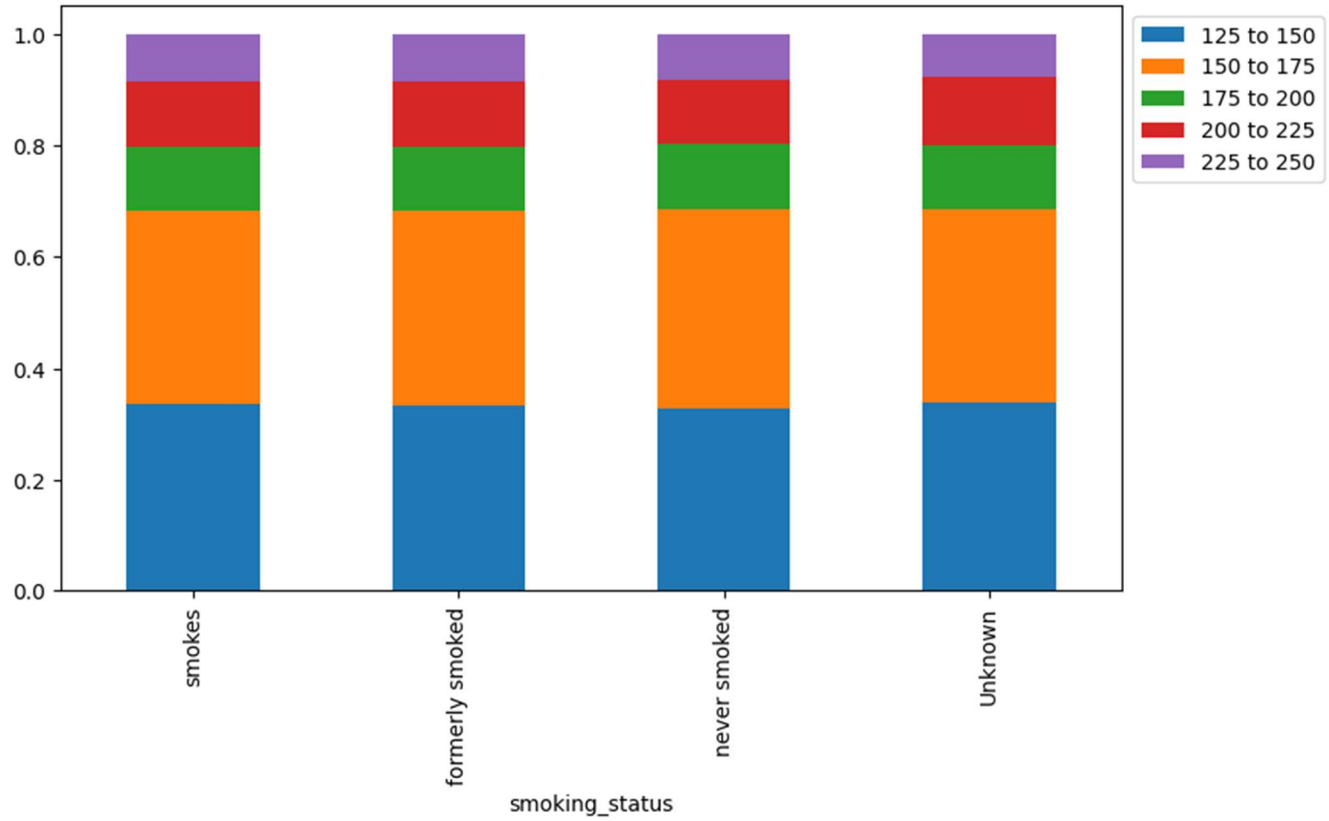
Gender	All
All	25000
Male	16422
Female	8578



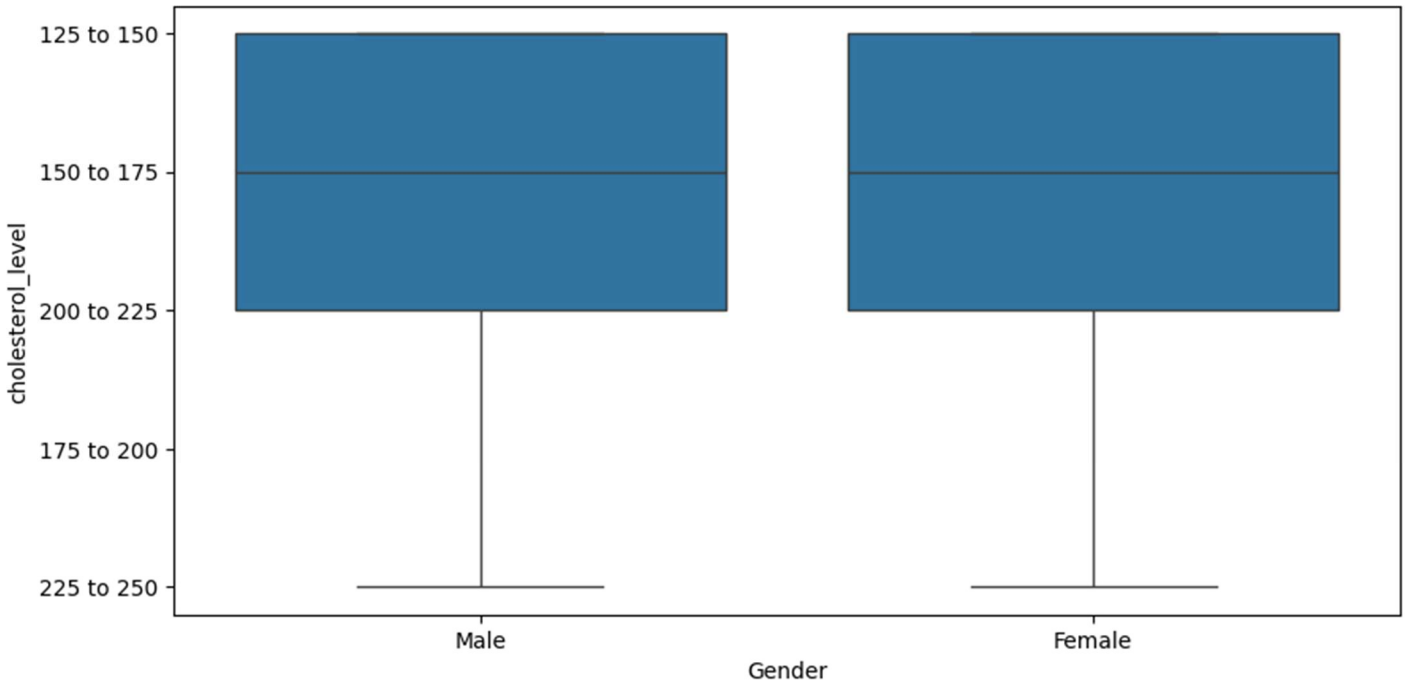
Cholesterol_level as per ‘Smoking_status

smoking_status	125 to 150	150 to 175	175 to 200	200 to 225	225 to 250
All	8339	8763	2881	2963	2054
never smoked	3038	3291	1086	1070	764
Unknown	2550	2631	852	927	595
formerly smoked	1448	1506	496	512	367
smokes	1303	1335	447	454	328

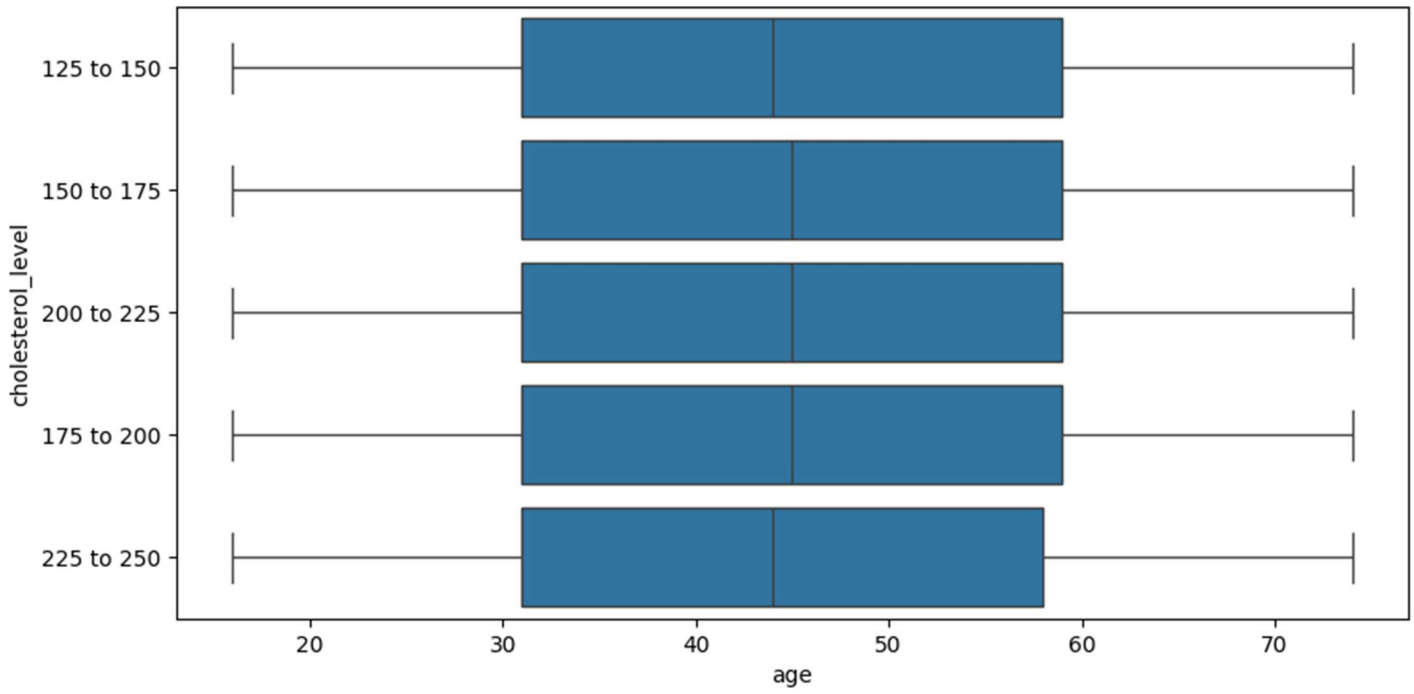
smoking_status	All
All	25000
never smoked	9249
Unknown	7555
formerly smoked	4329
smokes	3867



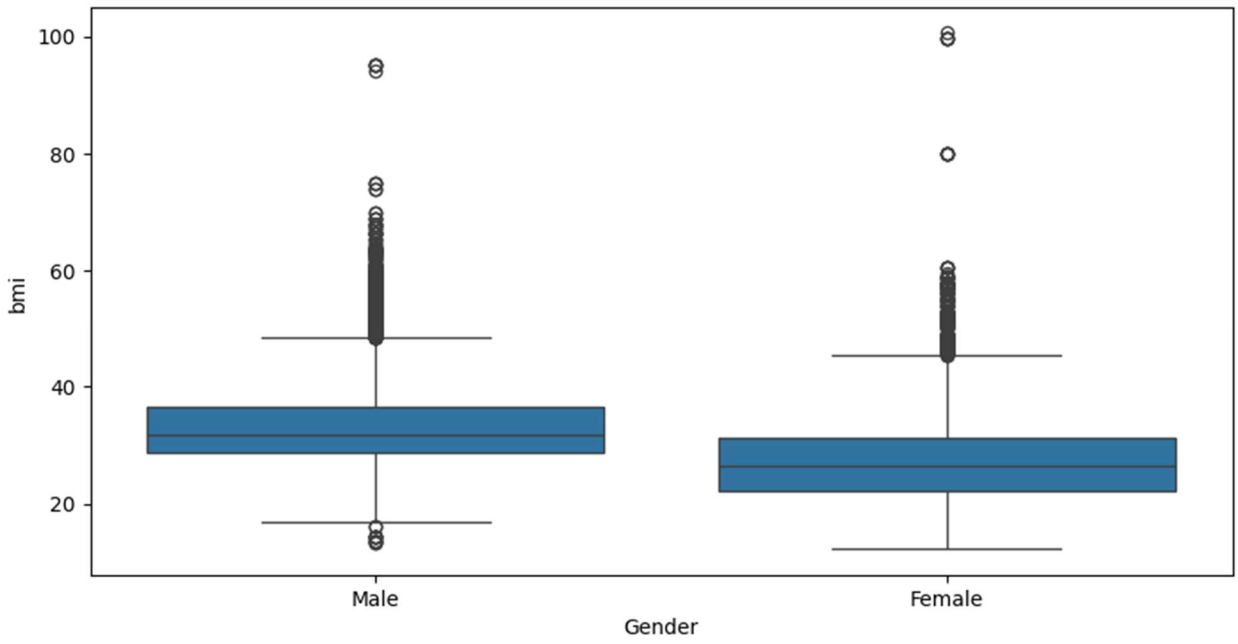
Plotting Boxplot for Cholesterol_level as per 'Gender'



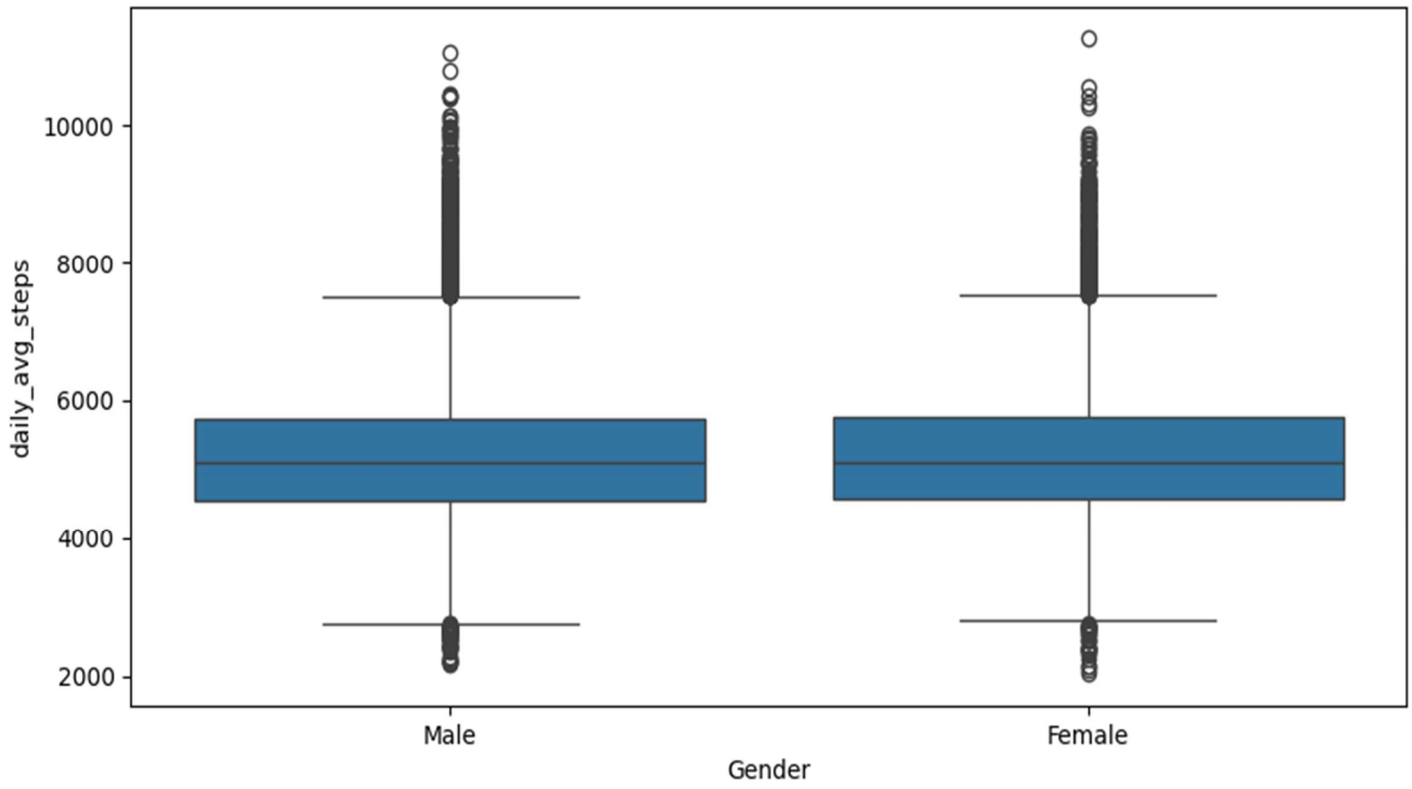
Plotting Boxplot for Cholesterol_level as per ‘Age’ Wise



Boxplot for BMI as per ‘Gender’



Boxplot for ‘daily_avg_steps’ as per ‘Gender’ Wise



-- Observation –

- We can observe that ‘Year_last_admitted’ column is somehow correlated with ‘Weight_change_in_last_one_year’ column which is around 25%. So we can say there's Positive relationship between them.

- Same goes for ‘Weight’ column it’s also highly correlated with Target column (insurance cost) which is around 97% and we can find Positive relationship between them.

- Also we can say that we have maximum number of people who fall under 125 to 150 cholesterol_level which is ‘8339’ where we have 5498 male and 2841 female applicants or user. If we are talking about we have more records available for cholesterol_level in Male gender as compare to female.

- If we are checking cholesterol_level as per ‘smoking_status’ here also we have maximum people fall under category of 125 to 150 choesterol_level. Also we have maximum number of people available who never_smoke at all. Apart from this if we are checking we have less amount of people who do smokes which is around 1303 and they fall under 125 to 150 cholesterol_level category.

- Also if we are checking the cholesterol_level as per the age wise we found that we have higher amount of people between the age of approx 32 to 56.

- While we are checking the ‘bmi’ status as per ‘gender’ we found that as compare to male we found lesser bmi available in Female category. Also we have few outliers available in both the list which we are going to take care later.

- Apart from this in ‘daily_avg_steps’ we can find that ‘Female’ Category is the one who covers maximum daily average steps which is approx 8000 steps after removing outliers.

3. Data Cleaning and Pre-processing:-

- Approach used for identifying and treating missing values and outlier treatment (and why) ……
- Need for variable transformation (if any) ……...
- Variables removed or added and why (if any) ……..

- Removal of unwanted variables (if applicable):-

- Before removing unwanted variables we are converting ‘Categorical columns’ to binary form and we found some categorical columns.

```
<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_last_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_disease_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                             25000 non-null  float64
 14  smoking_status                  25000 non-null  object
 15  Year_last_admitted              25000 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
```

- **Now we are converting those categorical column in binary form and also we are changing those data type from ‘obj’ to ‘int’.**

```
['Occupation',
 'cholesterol_level',
 'Gender',
 'smoking_status',
 'Location',
 'covered_by_any_other_company',
 'Alcohol',
 'exercise']
```

- We have changed those categorical columns using get_dummies function.

- Now we are using Chi2 method in order to get important columns out from our dataset. Because with the help of ‘p-value’ there we can get to know which columns are important for model building on the basis of ‘p-value’ threshold. If p-value is less than 0.05 we’ll keep that important else better to skip those columns.

- Below we have stored ‘p-value’ in array.

```
(array([1.17033476e+05, 2.83259400e+02, 1.84160823e+03, 2.28536063e+02,
        2.92472584e+01, 8.12155465e+03, 2.32466547e+02, 6.33625687e+01,
        5.92238186e+01, 1.40480508e+03, 8.70143146e+01, 1.93989967e+02,
        2.87068377e+04, 5.70612462e+03, 1.50550105e+02, 4.91001244e+01,
        2.66603433e+01, 3.88916162e+01, 5.95253917e+01, 5.58464859e+01,
        3.21920010e+01, 1.98694531e+01, 4.66731410e+01, 4.48430536e+01,
        4.92061366e+01, 4.72289435e+01, 3.99614523e+01, 5.24542440e+01,
        5.23699668e+01, 4.35324700e+01, 4.23517883e+01, 5.19731307e+01,
        5.96837430e+01, 6.05004454e+01, 3.72900543e+01, 4.00871188e+01,
        4.56888543e+01, 5.01192588e+01, 6.00352480e+01, 4.99635422e+02,
        2.16900844e+01, 1.79166593e+01, 3.12485913e+01, 5.40040719e+01]), array([0.00000000e+000, 3.42610852e-033, 0.00000000e+000, 1.15708349e-023,
        9.96709488e-001, 0.00000000e+000, 2.49270058e-024, 1.55932176e-001,
        2.58962150e-001, 4.55395425e-259, 2.23026511e-003, 5.91650362e-018,
        0.00000000e+000, 0.00000000e+000, 2.76982870e-011, 6.26760572e-001,
        9.99053353e-001, 9.26306149e-001, 2.50255792e-001, 3.68359523e-001,
        9.89351162e-001, 9.99990428e-001, 7.17473780e-001, 7.79744216e-001,
        6.22657563e-001, 6.97374006e-001, 9.06927196e-001, 4.95342810e-001,
        4.98629296e-001, 8.19909857e-001, 8.52451011e-001, 5.14149506e-001,
        2.45758590e-001, 2.23385914e-001, 9.49819950e-001, 9.04449948e-001,
        7.51778866e-001, 5.87037777e-001, 2.35960151e-001, 6.24361538e-074,
        9.99958658e-001, 9.99998444e-001, 9.92502994e-001, 4.35801018e-001]))
```

- Now we can see below it is showing us all the important features along with their ‘Chi-Square score’ and ‘p-value’ which are good to use for Model building to do prediction.

	Feature	Chi-Square Score	p-Value
0	applicant_id	117033.475783	0.000000e+00
12	weight	28706.837728	0.000000e+00
5	daily_avg_steps	8121.554645	0.000000e+00
13	weight_change_in_last_one_year	5706.124622	0.000000e+00
2	regular_checkup_last_year	1841.608226	0.000000e+00
9	avg_glucose_level	1404.805082	4.553954e-259
39	covered_by_any_other_company_Y	499.635422	6.243615e-74
1	years_of_insurance_with_us	283.259400	3.426109e-33
6	age	232.466547	2.492701e-24
3	adventure_sports	228.536063	1.157083e-23
11	Year_last_admitted	193.989967	5.916504e-18
14	fat_percentage	150.550105	2.769829e-11
10	bmi	87.014315	2.230265e-03

- After selecting important features and dropping unusual columns we have just 13 columns available in our dataset.

```
(25000, 13)
```

d) Missing Value treatment (if applicable):-

- Earlier at the beginning we have checked the ‘Null’ Values and we found that in 2 columns there which we can see it below which is in ‘bmi’ column we have around 990 and ‘year_last_admitted’ around 11881 Null values:-

	0
applicant_id	0
years_of_insurance_with_us	0
regular_checkup_last_year	0
adventure_sports	0
Occupation	0
visited_doctor_last_1_year	0
cholesterol_level	0
daily_avg_steps	0
age	0
heart_disease_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

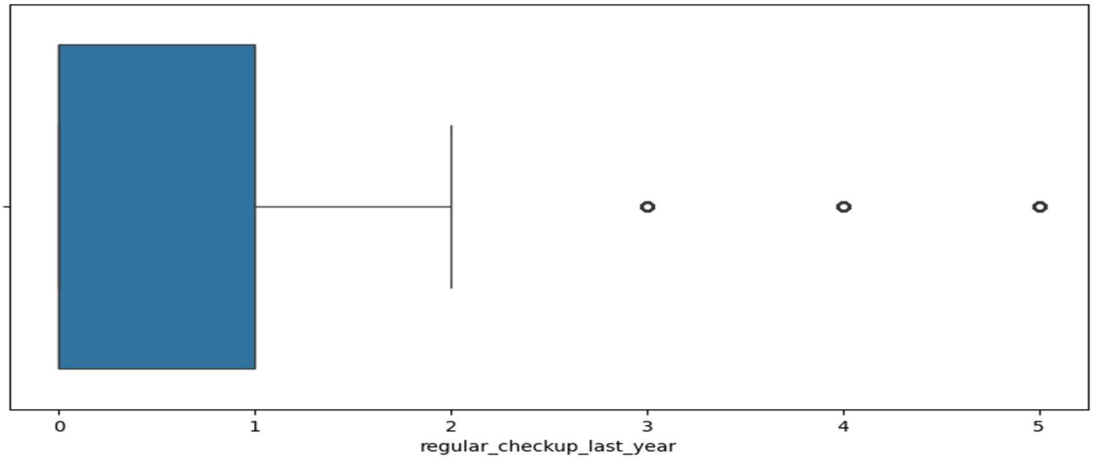
	0
applicant_id	0
years_of_insurance_with_us	0
regular_checkup_last_year	0
adventure_sports	0
Occupation	0
visited_doctor_last_1_year	0
cholesterol_level	0
daily_avg_steps	0
age	0
heart_disease_history	0
other_major_decs_history	0
Gender	0
avg_glucose_level	0
bmi	0
smoking_status	0
Year_last_admitted	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

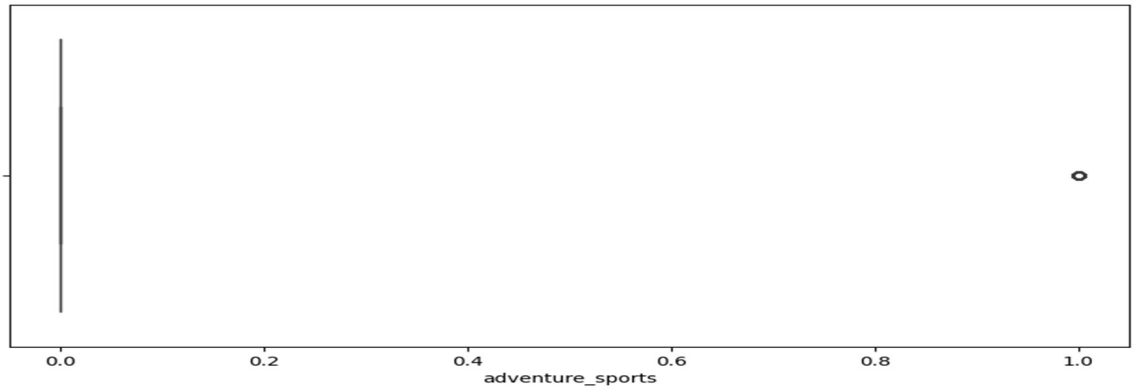
dtype: int64

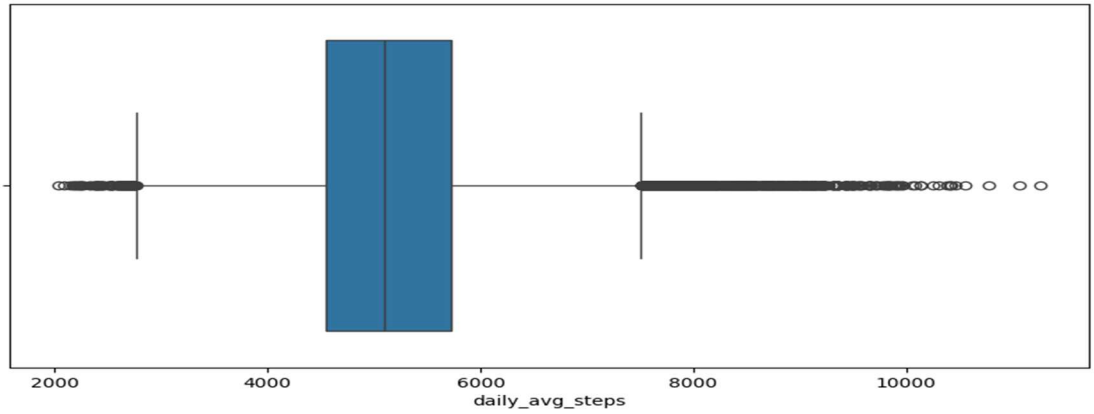
- So we have fill those ‘Null’ values using different method in ‘Year_last_admitted’ we have used mode method means which will occur most of the time are are filling with it and in ‘bmi’ column we have filled null value with the help of mean method.

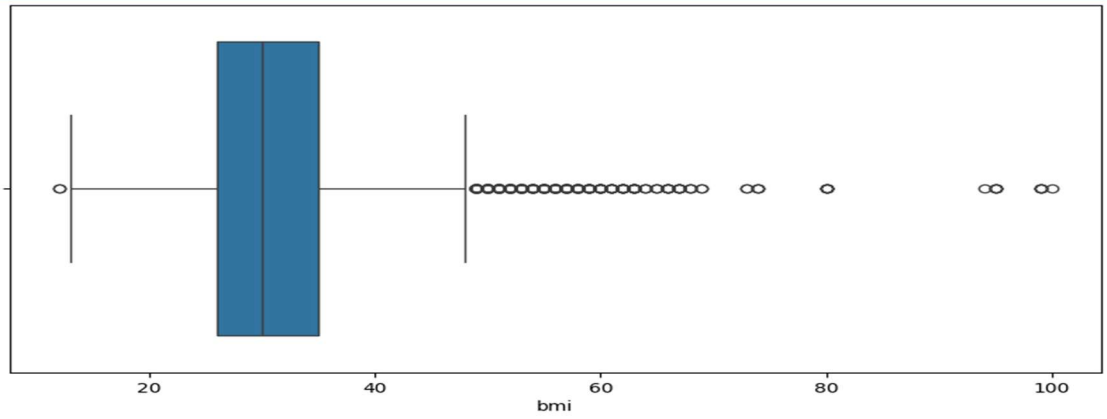
e) Outlier treatment (if required):-

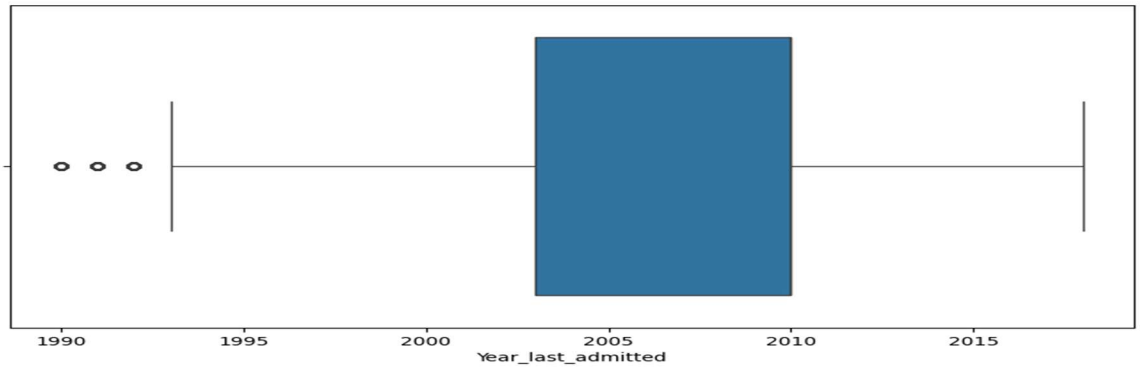
- We are checking for outliers in available columns by plotting Boxplot.











- As we found outliers in those Columns which we are going to treat using ‘IQR’ method.

```
Original dataset size: 25000 rows
Dataset size after removing outliers: 18413 rows
```

- Once it is done we can see that we have just 18413 rows available in our dataset earlier it was 25000.

f) Variable transformation (if applicable):-

- Now we are going to transformation part as our data records are in different units which is mentioned below.

	applicant_id	years_of_insurance_with_us	regular_checkup_last_year	adventure_sports	daily_avg_steps	age	avg_glucose_level	bmi	Year_last_admitted	weight	weight_change_in_last_one_year	fat_percentage	covered_by_any_other_cc
1	5001	0	0	0	6411	50	212	34	2010	58	3	27	
2	5002	1	0	0	4509	68	166	40	2010	73	0	32	
4	5004	3	1	0	4938	44	118	26	2004	74	0	34	
5	5005	8	0	0	5306	39	155	38	2003	78	3	13	
6	5006	8	0	0	4676	40	80	28	2004	81	3	16	

- So we have used ‘Standard_Scaler’ method in order to do scaling part in our dataset because it can scale them to mean of 0 and a standard deviation of 1. Once it is done we can see that in our dataset all the values are between 0 and 1.

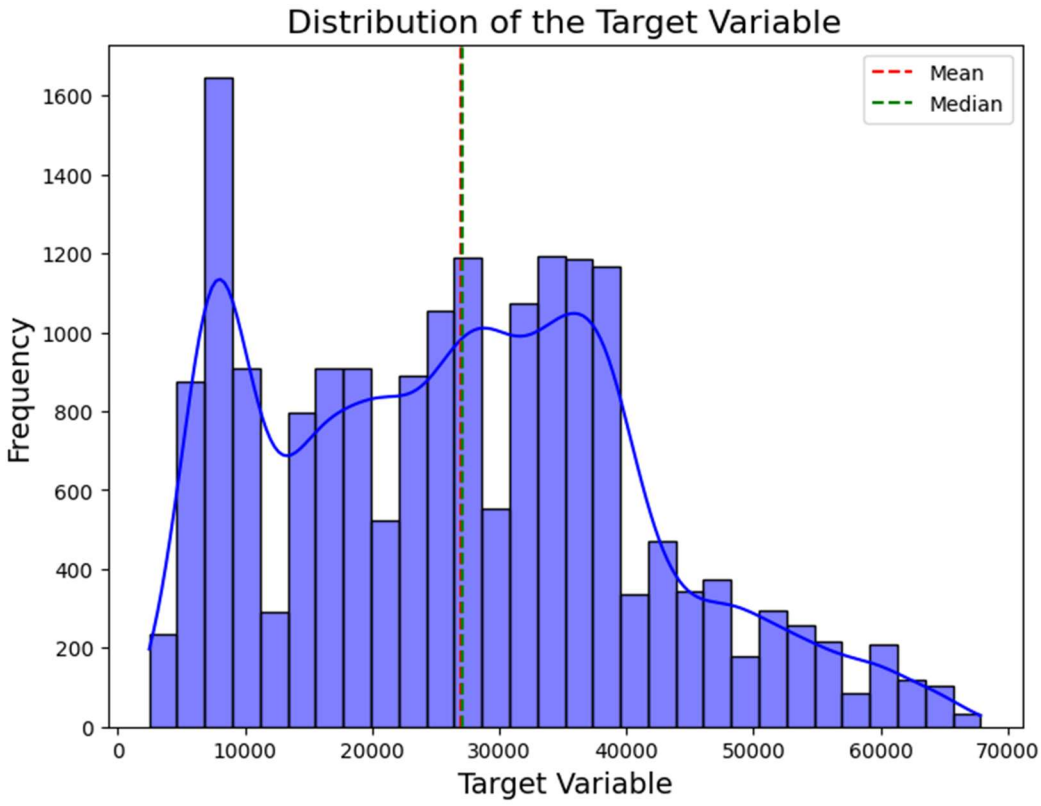
	applicant_id	years_of_insurance_with_us	regular_checkup_last_year	adventure_sports	daily_avg_steps	age	avg_glucose_level	bmi	Year_last_admitted	weight	weight_change_in_last_one_year	fat_percentage	covered_by
0	-1.732293	-1.526765	-0.618929	0.0	1.481307	0.317733	0.709873	0.545329	0.475370	-1.425882	0.252840	-0.211841	
1	-1.732155	-1.146020	-0.618929	0.0	-0.696183	1.436322	-0.024098	1.449055	0.475370	0.167982	-1.502641	0.368852	
2	-1.731878	-0.384529	0.882960	0.0	-0.205046	-0.055131	-0.789981	-0.659639	-0.545268	0.274240	-1.502641	0.601129	
3	-1.731740	1.519197	-0.618929	0.0	0.216256	-0.365850	-0.199613	1.147813	-0.715374	0.699270	0.252840	-1.837781	
4	-1.731602	1.519197	-0.618929	0.0	-0.504994	-0.303706	-1.396306	-0.358397	-0.545268	1.018043	0.252840	-1.489366	

g) Addition of new variables (if required):-

- As of now we don’t need to add new variables in our dataset it is ready to use for Model Building.

- **Is the data unbalanced? If so, what can be done? Please explain in the context of the business:-**

- We are checking here the distribution of ‘Target’ column in our dataset.



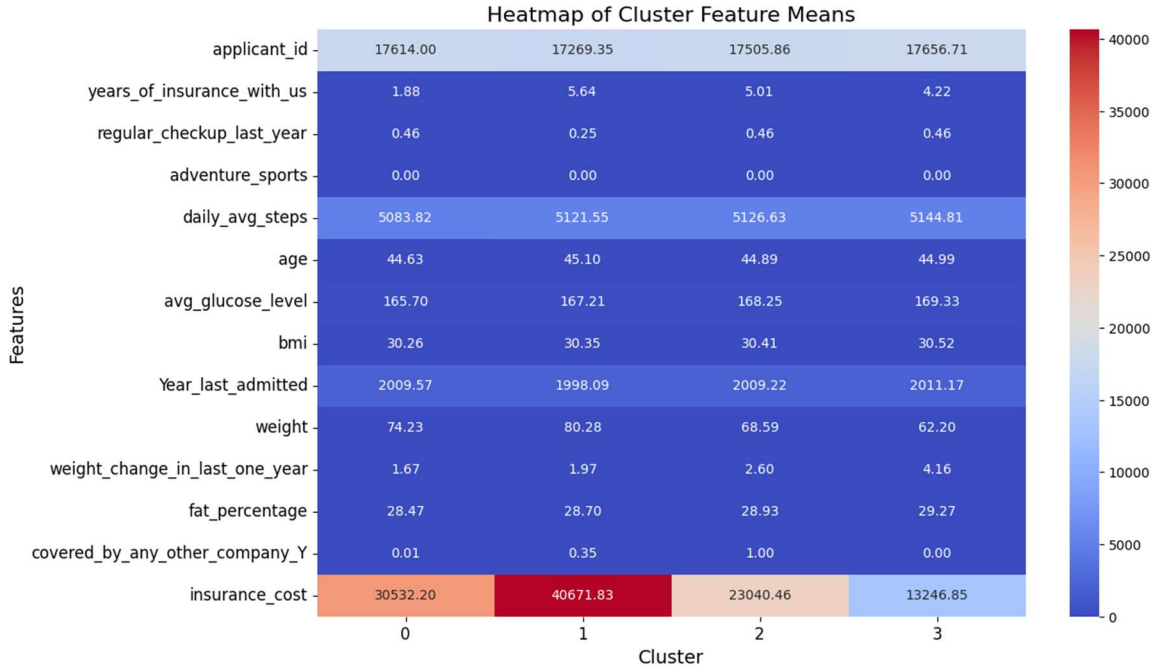
- The target variable appears to be slightly skewed to the right (positive skewness), as there are higher values extending further to the right.
- Also the mean (red dashed line) is slightly greater than the median (green dashed line), which is a characteristic of right-skewed distributions..

- We can apply transformation there like ‘Log Transformation’, ‘Square root transformation’ etc.
- Also we can evaluate model performance once we trained the model to reduce skeweness.

- Any business insights using clustering (if applicable):-

- We are doing clustering in order to understand data segmentation or user behavior.
- We have used K-Means clustering in order to segment our dataset in different cluster and you can see below snap for that.

cluster	applicant_id	years_of_insurance_with_us	regular_checkup_last_year	adventure_sports	daily_avg_steps	age	avg_glucose_level	bmi	Year_last_admitted	weight	weight_change_in_last_one_year	fat_percentage	cov
0	17613.995513	1.882807	0.459799	0.0	5083.822505	44.633525	165.704594	30.259512	2009.567660	74.227207	1.670136	28.467157	
1	17269.345345	5.642181	0.253176	0.0	5121.552553	45.100716	167.210903	30.352044	1998.090321	80.276276	1.973897	28.702010	
2	17505.863482	5.008092	0.460976	0.0	5126.629079	44.887497	168.246933	30.407204	2009.223962	68.589141	2.595406	28.933699	
3	17656.711600	4.215552	0.462294	0.0	5144.810938	44.991241	169.333689	30.524888	2011.171331	62.201453	4.163427	29.271950	



1. Key Observations:

- Years of Insurance with Us: Cluster 0 has a significantly lower mean, suggesting these customers might be newer to the company.Cluster 1 has higher loyalty (longer tenure) compared to other clusters.
- Year Last Admitted: Cluster 1 has a much earlier admission year (negative value indicating a distant past). Cluster 2 and 3 seem to have more recent admissions.
- Weight Change in Last One Year: Cluster 3 shows the highest positive weight change, suggesting possible health fluctuations. Cluster 0 has the most negative weight change, indicating customers might have lost weight.
- Covered by Any Other Company (Y): Cluster 2 has a very high value, indicating this group is significantly more likely to have other insurance providers. Cluster 0 is the least likely to have multiple insurance providers.

2. Suggestions for Business Action:

- Targeted Customer Retention: Focus on customers in Cluster 0 who are newer to the company. Design retention strategies like discounts, loyalty programs, or personalized offers.
- Marketing Campaigns: For customers in Cluster 2, emphasize your company's unique benefits to encourage them to prioritize your insurance over competitors.
- Health Improvement Programs: For customers in Cluster 3 with significant weight changes, introduce wellness programs, fitness benefits, or health monitoring perks to improve their health outcomes.
- Loyalty Rewards: For customers in Cluster 1 with longer tenure, offer loyalty-based incentives such as premium discounts or free add-ons to acknowledge their trust in your services.

c) Any other business insights:-

- Loyalty Analysis: Customers with longer tenure (e.g., Cluster 1) are less likely to switch providers. These customers can be prioritized for upselling or cross-selling additional products.Newer customers (e.g., Cluster 0) might require tailored marketing strategies to ensure retention.

- Health Patterns: Weight fluctuations and high BMI levels can indicate potential health risks. Offering targeted wellness programs could reduce long-term claims while improving customer satisfaction. Regular checkups might indicate a proactive approach to health. Customers with low checkup rates may benefit from reminders or incentives to schedule health evaluations.

---------- Key Recommendations for Business ----------

- Invest in data-driven pricing models that adjust premiums based on customer profiles.
- Enhance customer satisfaction through proactive health management programs.
- Leverage cluster insights to refine marketing and retention strategies.

4. Model building:-

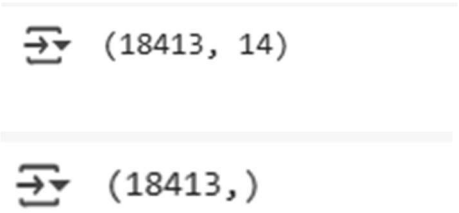
- Clear on why was a particular model(s) chosen.
- Effort to improve model performance.

---

a. Build various models (You can choose to build models for either or all of descriptive, predictive or prescriptive purposes)
b. Test your predictive model against the test set using various appropriate performance metrics

c. Interpretation of the model(s).

- Before building the Model we are checking the scaled dataset features (variables) “X” and target “y” shape.

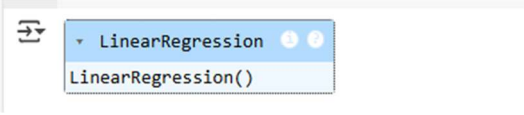


- Now we are splitting dataset in training and test part so we can build the Model.

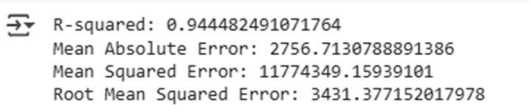
- So we have set 75% of data ‘13809 records’ for Training and rest 25% ‘4604 records’ for test purpose.

- **Linear Regression Model:-**

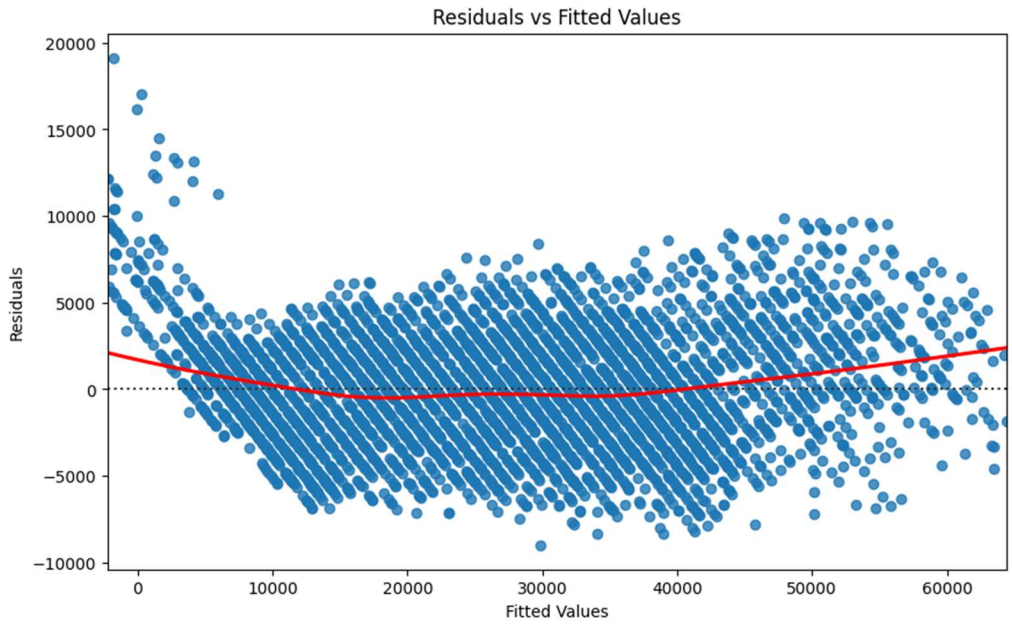
- We are Building regression model and fitting our dataset with ‘LinearRegression Model’.



- Once we fit our model then we are checking accuracy and error rate to evaluate the model performance.



- Also we are checking the linearity of this model below:-



-- Observation –

- ***Non-linear Pattern:***

The residuals show a distinct curve, indicating that the model does not fully capture the relationship between the predictors and the response variable. A linear regression might not be the best fit for this data.

- ***Heteroscedasticity (Variance of Residuals):***

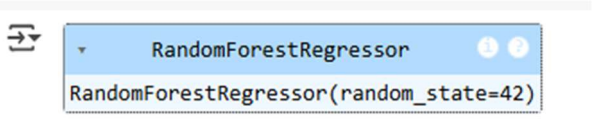
The spread of the residuals appears to change across the range of fitted values. At smaller fitted values, the variance seems larger, and as the fitted values increase, the spread reduces and then widens again slightly.

- ***Bias in the Model:***

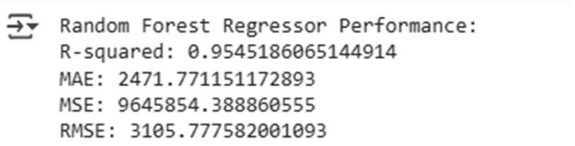
The red trend line (smoother) deviates significantly from zero across the range of fitted values, suggesting systematic bias in the model's predictions.

- **Random Forest Regressor Model:-**

- We are Building regressior model and fitting our dataset with ‘Random Forest Regressor Model’.



- After fitting our model then we are checking accuracy and error rate to evaluate the model performance.



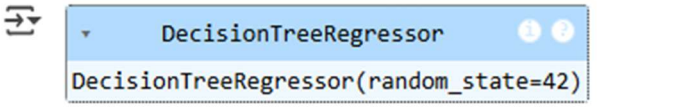
- As we can see here that “Random Forest Regressor” giving us R-Squared value of 95% and error as per MAE, MSE and RMSE.

- Random Forest's ensemble approach (averaging multiple trees) reduces overfitting and improves performance compared to a single Decision Tree.

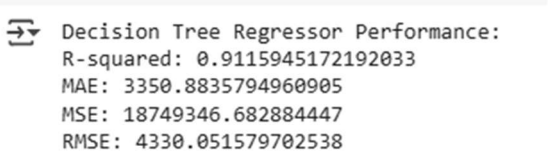
- The lower error metrics (MAE, MSE, RMSE) indicate that Random Forest provides better predictive stability and generalization.

- **Decision Tree Regressor Model:-**

- We are Building regressor model and fitting our dataset with ‘Decision Tree Regressor Model’.



- After fitting our model then we are checking accuracy and error rate to evaluate the model performance.



- As we can see here that “Random Forest Regressor” giving us R-Squared value of 91% and error as per MAE, MSE and RMSE.

- The high R^2 value indicates that the Decision Tree Regressor captures most of the variation in the data.

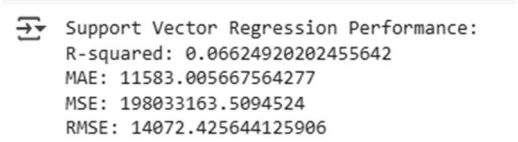
- The relatively low MAE and RMSE suggest good predictive performance.

- However, with the help of residuals we plot earlier it indicates that non-linear patterns and heteroscedasticity, which could mean the Decision Tree Regressor might still have room for improvement.

- **Support Vector Regressor (SVR) Model:-**

- We are Building regressor model and fitting our dataset with ‘Support Vector Regressor Model’.

- After fitting our model then we are checking accuracy and error rate to evaluate the model performance.



- R-squared (0.0662): The SVR explains only 6.62% of the variance in the target variable, which is very low and indicates poor model performance.

- Mean Absolute Error (11,583.01): The average prediction error is much larger than those of the Decision Tree and Random Forest models, indicating that the predictions are far off from the actual values.

- Mean Squared Error (198,033,163.51): The MSE is extremely high, suggesting the presence of large prediction errors.

- Root Mean Squared Error (14,072.43): The RMSE confirms that the SVR's predictions are far less accurate compared to the other models.

* Descriptive, Predictive, and Prescriptive Models:-

- **Descriptive Models:** These models help understand past data trends and relationships. In this study, exploratory data analysis (EDA) provided insights into variable distributions, correlations, and important features impacting insurance costs.
- **Predictive Models:** Various regression models, including Linear Regression, Decision Tree, Random Forest, and Support Vector Regressor (SVR), were built to predict insurance costs based on health and lifestyle data.
- **Prescriptive Models:** Although not explicitly implemented, prescriptive analytics could involve optimization algorithms suggesting the best insurance plan based on customer behavior and health parameters.

* Interpretation of Models:-

- **Linear Regression:** Showed bias and failed to capture non-linear relationships.
- **Decision Tree:** Improved accuracy but risked overfitting.
- **Random Forest:** Provided the best balance between bias and variance, achieving 95.6% R^2.
- **Support Vector Regressor:** Initially underperformed but improved with tuning, though still inferior to Random Forest.

5. Model validation:-

- How was the model validated? Just accuracy, or anything else too?

---------- Model Tuning ----------

a. Ensemble modelling, wherever applicable

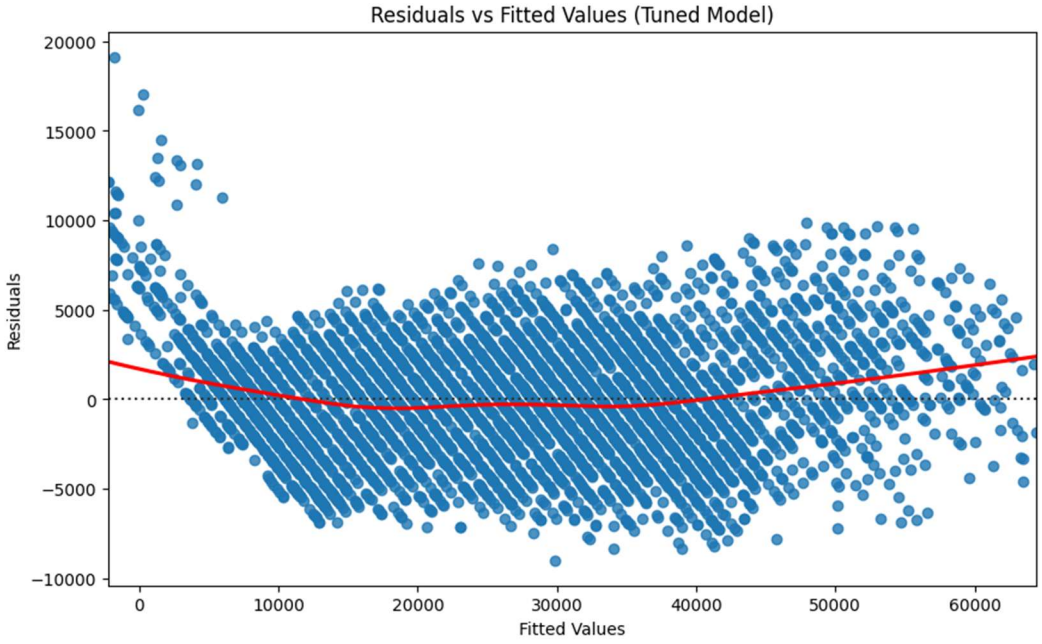
b. Any other model tuning measures(if applicable)
c. Interpretation of the most optimum model and its implication on the business.

- **Tuned Linear Regression Model:-**

- We have Tuned our 'LinearRegression Model' with the help of 'GridSearchCV' and gave parameters like cv=5 and fitted again our train model.

- Once we Tuned our model then we are checking accuracy and error rate to evaluate the model performance.

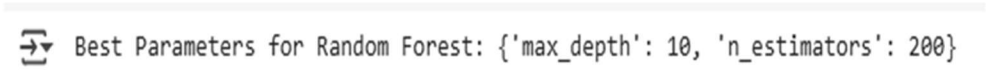
```
R-squared: 0.944482491071764
Mean Absolute Error: 2756.7130788891386
Mean Squared Error: 11774349.15939101
Root Mean Squared Error: 3431.377152017978
```



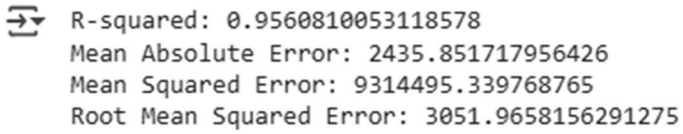
- As we can see above almost we are getting same accuracy and error rate which we got before tuning our model.

- **Tuned Random Forest Regressor Model:-**

- We are Tuning our existing “Random Forest Regressor Model’ with the help of few parameters like ‘n_estimators’, ‘max_depth’ etc.



- After Tuning our model then we are checking accuracy and error rate to evaluate the model performance.



- R-squared (0.9561): The current model shows 95.61% of the variance in the target variable, slightly better than the previous Random Forest model.

- Mean Absolute Error (2435.85): The average error has decreased slightly, indicating slightly better accuracy for most predictions.

- Mean Squared Error and RMSE: Both MSE and RMSE are slightly reduced, suggesting fewer and smaller large prediction errors.

- These incremental improvements suggest that the model has been further optimized. If the current model is still a Random Forest, the improvements might result from hyperparameter tuning.

- The performance is excellent, as R^2 is very close to 1, and error metrics (MAE, RMSE) are reasonably low.

- **Tuned Decision Tree Regressor Model:-**

- We are Tuning our existing “Decision Tree Regressor Model’ with the help of few parameters like ‘max_depth’, ‘min_samples_split’ etc.

- After Tuning our model then we are checking accuracy and error rate to evaluate the model performance.

```
R-squared (R2): 0.9525592206882104
Mean Absolute Error (MAE): 2531.3866737859807
Mean Squared Error (MSE): 10061407.847615624
Root Mean Squared Error (RMSE): 3171.9722331091775
```

- R-squared (0.9526): The tuned Decision Tree now shows 95.26% of the variance in the target variable, a substantial improvement over the untuned version (91.16%) but slightly below Random Forest (95.61%).

- Mean Absolute Error (2531.39): MAE is significantly reduced compared to the untuned version, suggesting better overall prediction accuracy, though it's slightly higher than the Random Forest model (2435.85).

- Mean Squared Error (10,061,407.85): The MSE has reduced significantly, indicating fewer large errors, but it is still higher than Random Forest.

- Root Mean Squared Error (3171.97): RMSE has also decreased substantially from the untuned version, but again, Random Forest performs slightly better.
- The tuned Decision Tree is now much closer to Random Forest in terms of performance, but it still falls short, especially in error metrics (MAE, RMSE).

- Overfitting Risk: Decision Trees are prone to overfitting, especially when the depth is high. Random Forest mitigates this risk by averaging predictions over multiple trees.

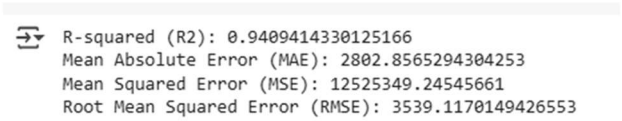
- Random Forest Superiority: The ensemble nature of Random Forest inherently provides better generalization and stability compared to a single Decision Tree.

- **Tuned Support Vector Regressor (SVR) Model:-**

- We are Tuning our existing "Support Vector Regressor Model' with the help of few parameters like 'C', 'kernel', 'gamma' etc.

```
Best Parameters for SVR: {'C': 10, 'gamma': 'scale', 'kernel': 'linear'}
```

- After Tuning our model then we are checking accuracy and error rate to evaluate the model performance.



- R-squared (0.9409): The tuned SVR explains ~94.09% of the variance in the target variable, a huge improvement over the untuned SVR (6.62%). However, it is slightly behind the Decision Tree (95.26%) and Random Forest (95.61%).

- Mean Absolute Error (2802.86): The MAE has improved drastically from the untuned model, but it is higher than both the tuned Decision Tree (2531.39) and Random Forest (2435.85).

- Mean Squared Error (12,525,349.25): The MSE is significantly better than the untuned SVR, but still higher than the Decision Tree and Random Forest.

- Root Mean Squared Error (3539.12): While the RMSE shows major improvement from the untuned version, it remains higher than the other two models.

- Improvement: The SVR model has been successfully tuned, showing drastic improvements across all metrics.
- Performance Gap: Despite improvements, SVR still underperforms compared to Random Forest and the tuned Decision Tree in terms of R^2 and error metrics.
- Computational Cost: SVR tends to be computationally expensive, especially with large datasets or complex hyperparameter tuning.

* Ensemble Modeling and Tuning:-

- **Random Forest Tuning:**
  - n_estimators and max_depth were optimized to improve predictive power.
  - Achieved R^2 of **95.6%**, MAE of **2435.85**, and RMSE of **3051.97**.

* Feature Importance:-

- **Top Features Influencing Insurance Cost:**
  1. Weight (97% correlation with insurance cost)
  2. BMI
  3. Years of insurance with the company
  4. Cholesterol level
  5. Smoking status

* Business Implications:-

- **Personalized Insurance Pricing:** Adjust premium calculations based on key health indicators.
- **Customer Engagement:** Offer wellness programs to high-risk individuals to reduce insurance claims.
- **Competitive Advantage:** Use predictive insights to tailor insurance plans and attract more customers.

* Recommendations for Business Growth:-

1. **Loyalty Programs:** Reward long-term customers with discounts.
2. **Health Improvement Plans:** Encourage fitness programs to improve policyholder health.
3. **Dynamic Pricing Models:** Implement AI-driven pricing adjustments to reflect real-time risk assessments.
4. **Preventive Health Checkups:** Encourage early detection programs, reducing long-term insurance claims.

6. Final interpretation / recommendation:-

- Detailed recommendations for the management/client based on the analysis done.

- **Best Model Selection -**

- **R-squared ($R2$):** Tuned Random Forest has the highest $R2$ (0.9561), indicating it explains the most variance in the target variable.

- **MAE (Mean Absolute Error):** Tuned Random Forest has the lowest MAE (2435.85), showing it has the smallest average error.
- **RMSE (Root Mean Squared Error):** Tuned Random Forest also has the lowest RMSE (3051.97), indicating it has the smallest overall error.
- **SVR:** Both the untuned and tuned SVR models perform poorly compared to the others. This suggests SVR is not the best model for this data.

---------- Conclusion ----------

- Based on the metrics, Tuned Random Forest Regressor is the best model because it has:

- The highest $R2$,
- The lowest MAE, and
- The lowest RMSE.

---------- End ----------