INSURANCE AMOUNT PREDICTION

The insurance dataset contains information about individuals' age, gender, BMI, number of children, smoking status, and region,Medical history,Family medical history,Exercise frequently,occupation,coverage level as well as the associated insurance charges.

- **Age: The age of the customer. (Float):**
- **Gender : Gender of the person (object):**
- **bmi : Body Mass Index(Float):**
- **Children: The number of children the customer has. (Integer)**
- **Smoker: Whether or not the customer is a smoker. (Object) ▯ ▯**
- **Region: The region the customer lives in. (Object)**
- **Medical history : Medical history of the person(object)**
- **Family Medical History : Medical History of the family of the person (object)**
- **exercise_frequency : Wheather the person exercise frequently or not (object)**
- **Occupation : Job of the person (object)**
- **Coverage level : Which level of coverage has he owns like Premium,Standard,Basic (object)**
- **Charges: The insurance charges for the customer. (Float)**

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
df=pd.read_csv('/content/Insurance_Final.csv')
df
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupa
0	46.0	male	21.45	5	yes	southeast	Diabetes	NaN	Never	Blue c
1	25.0	female	25.38	2	yes	northwest	Diabetes	High blood pressure	Occasionally	White c
2	38.0	male	44.88	2	yes	southwest	NaN	High blood pressure	Occasionally	Blue c

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupa
3	25.0	male	19.89	0	no	northwest	NaN	Diabetes	Rarely	White c
4	49.0	male	38.21	3	yes	northwest	Diabetes	High blood pressure	Rarely	White c
...	...	...	...	...	...	...	...	...	...	
49995	29.0	male	37.91	4	no	northeast	Heart disease	Diabetes	Frequently	Stu
49996	39.0	female	20.57	1	no	northeast	High blood pressure	High blood pressure	Frequently	Blue c
49997	23.0	female	37.22	4	yes	northeast	Heart disease	High blood pressure	Occasionally	Blue c
49998	65.0	male	45.35	0	yes	southwest	NaN	NaN	Occasionally	Unempl
49999	22.0	female	27.26	1	no	southeast	High blood pressure	Diabetes	Frequently	White c

50000 rows × 12 columns

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   age                     49999 non-null  float64
 1   gender                  49998 non-null  object 
 2   bmi                     49999 non-null  float64
 3   children                50000 non-null  int64  
 4   smoker                  49998 non-null  object 
 5   region                  49999 non-null  object 
 6   medical_history         37498 non-null  object 
 7   family_medical_history  37482 non-null  object 
 8   exercise_frequency      50000 non-null  object 
 9   occupation              49999 non-null  object 
 10  coverage_level          49996 non-null  object 
 11  charges                 49999 non-null  float64
dtypes: float64(3), int64(1), object(8)
memory usage: 4.6+ MB
```

OBSERVATION ON DATASET

- **To find the person with highest insurance amount**

In [ ]:

```
print(df['charges'].max())
df_max=df.loc[df['charges']==32087.05668627213]
df_max
```

```
32087.05668627213
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupati
31763	36.0	male	42.89	4	yes	northeast	Heart disease	Heart disease	Frequently	Whi coll

- **To find person with lowest insurance amount**

In [ ]:

```
print(df['charges'].min())
```

```
df_max=df.loc[df['charges']==4472.317058132149]
df_max
```

4472.317058132149

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupatio
254	34.0	female	22.69	1	no	northwest	NaN	NaN	Never	Unemploye

In [ ]:

```
df.head()
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupation
0	46.0	male	21.45	5	yes	southeast	Diabetes	NaN	Never	Blue collar
1	25.0	female	25.38	2	yes	northwest	Diabetes	High blood pressure	Occasionally	White collar
2	38.0	male	44.88	2	yes	southwest	NaN	High blood pressure	Occasionally	Blue collar
3	25.0	male	19.89	0	no	northwest	NaN	Diabetes	Rarely	White collar
4	49.0	male	38.21	3	yes	northwest	Diabetes	High blood pressure	Rarely	White collar

In [ ]:

```
df.tail()
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupa
49995	29.0	male	37.91	4	no	northeast	Heart disease	Diabetes	Frequently	Stu
49996	39.0	female	20.57	1	no	northeast	High blood pressure	High blood pressure	Frequently	Blue c
49997	23.0	female	37.22	4	yes	northeast	Heart disease	High blood pressure	Occasionally	Blue c
49998	65.0	male	45.35	0	yes	southwest	NaN	NaN	Occasionally	Unempl
49999	22.0	female	27.26	1	no	southeast	High blood pressure	Diabetes	Frequently	White c

SCATTER PLOT

In [ ]:

```
x=df.iloc[:,:-1]
x=x.head(50)
x
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupatio
0	46.0	male	21.45	5	yes	southeast	Diabetes	NaN	Never	Blue colla
1	25.0	female	25.38	2	yes	northwest	Diabetes	High blood pressure	Occasionally	White colla
2	38.0	male	44.88	2	yes	southwest	NaN	High blood pressure	Occasionally	Blue colla
3	25.0	male	19.89	0	no	northwest	NaN	Diabetes	Rarely	White colla

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupatio
4	49.0	male	38.21	3	yes	northwest	Diabetes	High blood pressure	Rarely	White colla
5	55.0	female	36.41	0	yes	northeast	NaN	NaN	Never	Studen
6	64.0	female	20.12	2	no	northeast	High blood pressure	High blood pressure	Never	Blue colla
7	53.0	male	30.51	4	no	southeast	Heart disease	High blood pressure	Rarely	Studen
8	40.0	female	44.93	2	yes	northeast	NaN	Diabetes	Occasionally	Unemploye
9	22.0	female	32.13	5	yes	northeast	Diabetes	NaN	Never	Studen
10	21.0	male	42.08	1	yes	northwest	NaN	Diabetes	Rarely	Studen
11	45.0	female	39.68	1	no	northwest	High blood pressure	High blood pressure	Occasionally	Blue colla
12	56.0	female	44.86	0	yes	northwest	NaN	Heart disease	Rarely	Unemploye
13	55.0	male	39.95	5	yes	southeast	High blood pressure	High blood pressure	Frequently	Studen
14	24.0	female	48.98	5	yes	southwest	Diabetes	High blood pressure	Rarely	White colla
15	36.0	male	39.17	2	yes	southwest	High blood pressure	NaN	Occasionally	Studen
16	45.0	female	43.77	4	yes	northwest	Diabetes	Heart disease	Never	Unemploye
17	32.0	female	45.46	2	yes	southeast	NaN	High blood pressure	Occasionally	Blue colla
18	30.0	male	24.69	0	yes	southeast	NaN	Diabetes	Frequently	White colla
19	46.0	female	31.27	1	no	southwest	NaN	Diabetes	Frequently	Unemploye
20	25.0	male	39.82	3	no	northwest	Diabetes	NaN	Occasionally	White colla
21	64.0	female	29.31	2	no	northeast	High blood pressure	NaN	Frequently	Unemploye
22	65.0	female	31.11	5	yes	southeast	High blood pressure	Diabetes	Never	White colla
23	25.0	male	22.15	4	no	northwest	Diabetes	NaN	Occasionally	Unemploye
24	35.0	male	46.83	1	yes	southwest	Diabetes	High blood pressure	Frequently	White colla
25	60.0	female	23.83	2	no	northwest	Heart disease	NaN	Rarely	Studen
26	65.0	female	39.70	4	yes	southwest	Heart disease	High blood pressure	Never	Studen
27	26.0	male	27.74	0	yes	southwest	NaN	NaN	Never	White colla
28	43.0	female	26.46	0	yes	southwest	High blood pressure	Heart disease	Occasionally	White colla
29	33.0	female	30.75	1	no	southwest	High blood pressure	Diabetes	Never	White colla
30	44.0	male	41.15	4	no	northwest	Heart disease	High blood pressure	Occasionally	White colla
31	19.0	male	30.97	4	no	northeast	High blood pressure	Diabetes	Frequently	Studen
32	55.0	male	47.62	3	yes	northeast	Diabetes	Diabetes	Occasionally	White colla
33	41.0	male	48.97	4	yes	northeast	High blood pressure	Heart disease	Never	White colla
34	41.0	female	43.83	2	no	northeast	Diabetes	Heart disease	Frequently	White colla
35	38.0	female	25.90	4	yes	southwest	NaN	Diabetes	Frequently	Studen
36	22.0	male	35.34	5	no	southwest	High blood pressure	Heart disease	Occasionally	Blue colla
37	26.0	female	45.06	2	no	southeast	NaN	Diabetes	Frequently	Unemploye
38	26.0	female	35.78	4	no	southwest	High blood pressure	High blood pressure	Rarely	Unemploye
39	42.0	male	40.54	0	yes	northwest	High blood pressure	High blood pressure	Occasionally	Blue colla
40	44.0	female	24.01	1	no	northwest	Diabetes	Diabetes	Never	Studen

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	occupatio
41	[illegible]	female	44.74	0	no	southeast	High blood pressure	Heart disease	Occasionally	White colla
42	38.0	male	46.70	5	no	southeast	NaN	High blood pressure	Frequently	Blue colla
43	65.0	male	23.19	1	no	southeast	High blood pressure	NaN	Frequently	Studen
44	19.0	male	22.52	1	no	southeast	Diabetes	High blood pressure	Rarely	Studen
45	35.0	male	49.01	4	no	northwest	Heart disease	NaN	Never	Unemploye
46	23.0	male	38.53	2	yes	southeast	Heart disease	Heart disease	Never	Unemploye
47	31.0	male	30.40	3	no	southeast	High blood pressure	Heart disease	Frequently	Blue colla
48	46.0	female	49.42	1	yes	northeast	High blood pressure	Diabetes	Rarely	Blue colla
49	56.0	male	40.31	2	yes	northwest	Heart disease	High blood pressure	Occasionally	Unemploye

In [ ]:

```
y=df.iloc[:,-1]
y=y.head(50)
y
```

Out[ ]:

```
0     20460.307669
1     20390.899218
2     20204.476302
3     11789.029843
4     19268.309838
5     11896.836613
6      9563.655011
7     15845.293730
8     14036.544129
9     13669.577830
10    18996.131561
11    14892.145930
12    17740.278300
13    16972.489611
14    16243.133212
15    13683.049130
16    18334.599389
17    14174.328774
18    18455.147694
19    13775.765035
20    11014.284341
21     9414.800786
22    21821.940055
23     8327.544962
24    20364.433860
25    16140.478462
26    23176.908664
27    11621.388689
28    18725.811332
29    14536.912975
30    16161.137819
31    12623.384578
32    21609.957520
33    23529.766655
34    17183.601696
35    17034.559108
36    14743.157943
37    14922.198081
38     9943.371682
39    17637.797647
40     8713.333376
41    15075.217324
42    14399.173901
43    11050.255459
```

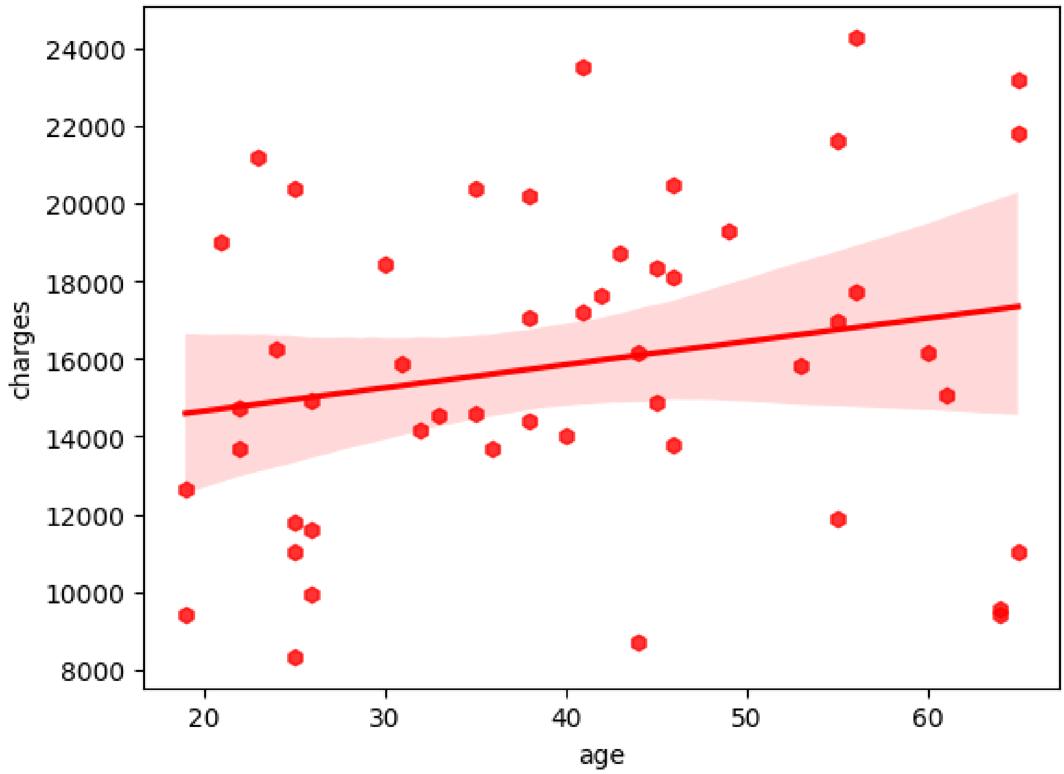
```
44     9421.413112
45    14612.231337
46    21195.320075
47    15853.048501
48    18121.971216
49    24275.473776
Name: charges, dtype: float64
```

In [ ]:

```
sns.regplot(x=x['age'],y=y,marker='h',color='r')
```

Out[ ]:

```
<Axes: xlabel='age', ylabel='charges'>
```

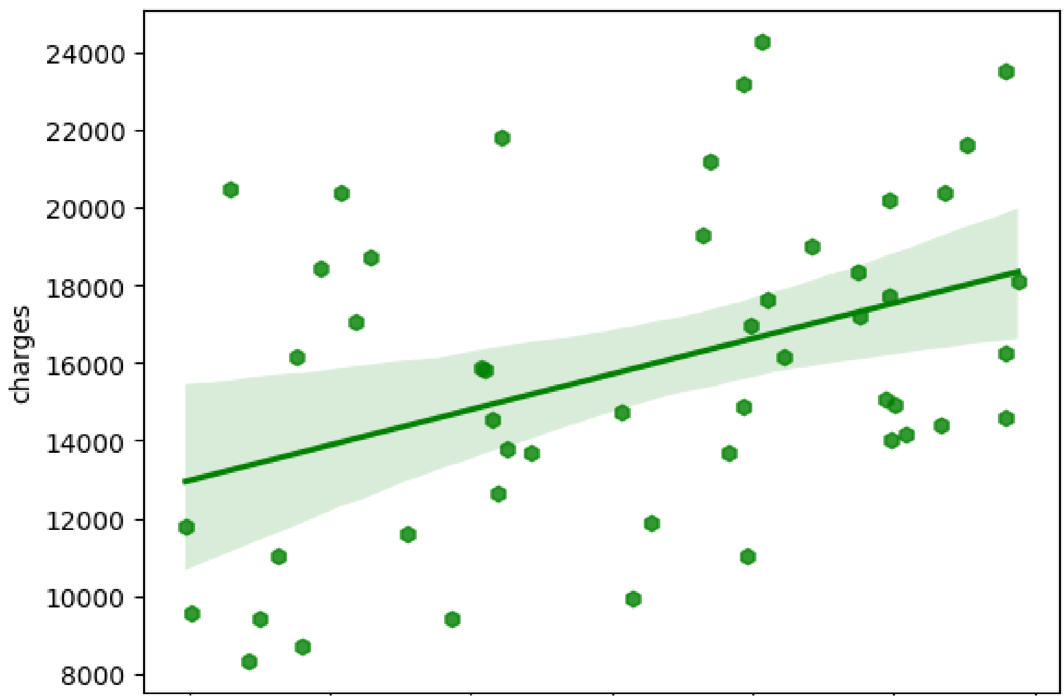


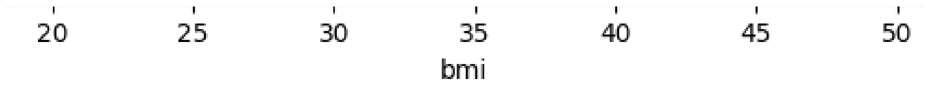
In [ ]:

```
sns.regplot(x=x['bmi'],y=y,color='g',marker='h')
```

Out[ ]:

```
<Axes: xlabel='bmi', ylabel='charges'>
```



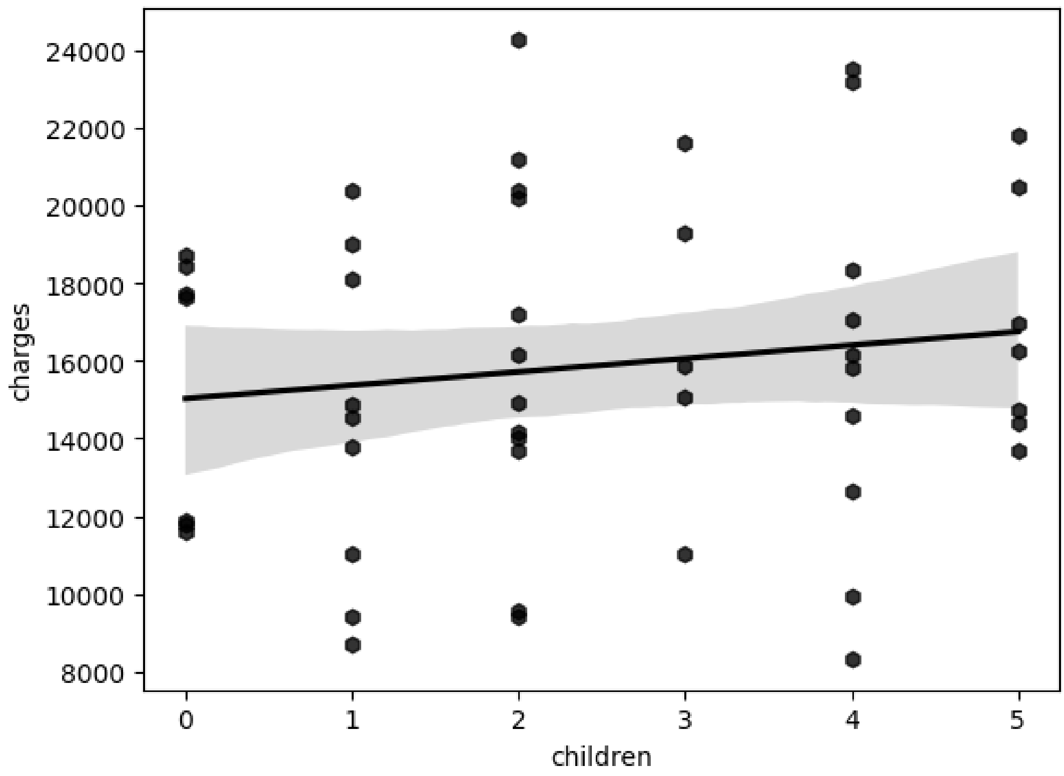


In [ ]:

```
sns.regplot(x=x['children'],y=y,color='k',marker='h')
```

Out[ ]:

```
<Axes: xlabel='children', ylabel='charges'>
```



In [ ]:

```
df.isna().sum()
```

Out[ ]:

```
age                          1
gender                       2
bmi                          1
children                     0
smoker                       2
region                       1
medical_history          12502
family_medical_history   12518
exercise_frequency           0
occupation                   1
coverage_level               4
charges                      1
dtype: int64
```

FILLING MISSING VALUES

In [ ]:

```
df['age']=df['age'].fillna(df['age'].mean())
df['gender']=df['gender'].fillna(df['gender'].mode() [0])
df['bmi']=df['bmi'].fillna(df['bmi'].mean())
df['smoker']=df['smoker'].fillna(df['smoker'].mode() [0])
df['region']=df['region'].fillna(df['region'].mode() [0])
df['medical_history']=df['medical_history'].fillna(df['medical_history'].mode() [0])
df['family_medical_history']=df['family_medical_history'].fillna(df['family_medical_hist
ory'].mode() [0])
df['occupation']=df['occupation'].fillna(df['occupation'].mode() [0])
```

```
df['coverage_level']=df['coverage_level'].fillna(df['coverage_level'].mode() [0])
df['charges']=df['charges'].fillna(df['charges'].mode() [0])
```

VISUALIZATION

In [ ]:

```
lst=['gender','smoker','region','medical_history','family_medical_history','exercise_frequency','occupation','coverage_level']
for col in lst:
    fig=px.pie(df,names=col,title=f'Pie chart of {col}')
    fig.show()
```

```
Output hidden; open in https://colab.research.google.com to view.
```

In [ ]:

```
lst1=['gender','smoker','region','medical_history','family_medical_history','exercise_frequency','occupation','coverage_level']
for i in lst1:
  print(i,df[i].unique())
```

```
gender ['male' 'female']
smoker ['yes' 'no']
region ['southeast' 'northwest' 'southwest' 'northeast']
medical_history ['Diabetes' 'High blood pressure' 'Heart disease']
family_medical_history ['Diabetes' 'High blood pressure' 'Heart disease']
exercise_frequency ['Never' 'Occasionally' 'Rarely' 'Frequently']
occupation ['Blue collar' 'White collar' 'Student' 'Unemployed']
coverage_level ['Premium' 'Standard' 'Basic']
```

In [ ]:

```
df.dtypes
```

Out[ ]:

```
age                          float64
gender                        object
bmi                          float64
children                       int64
smoker                        object
region                        object
medical_history               object
family_medical_history        object
exercise_frequency            object
occupation                    object
coverage_level                object
charges                      float64
dtype: object
```

In [ ]:

```
df.dtypes
```

Out[ ]:

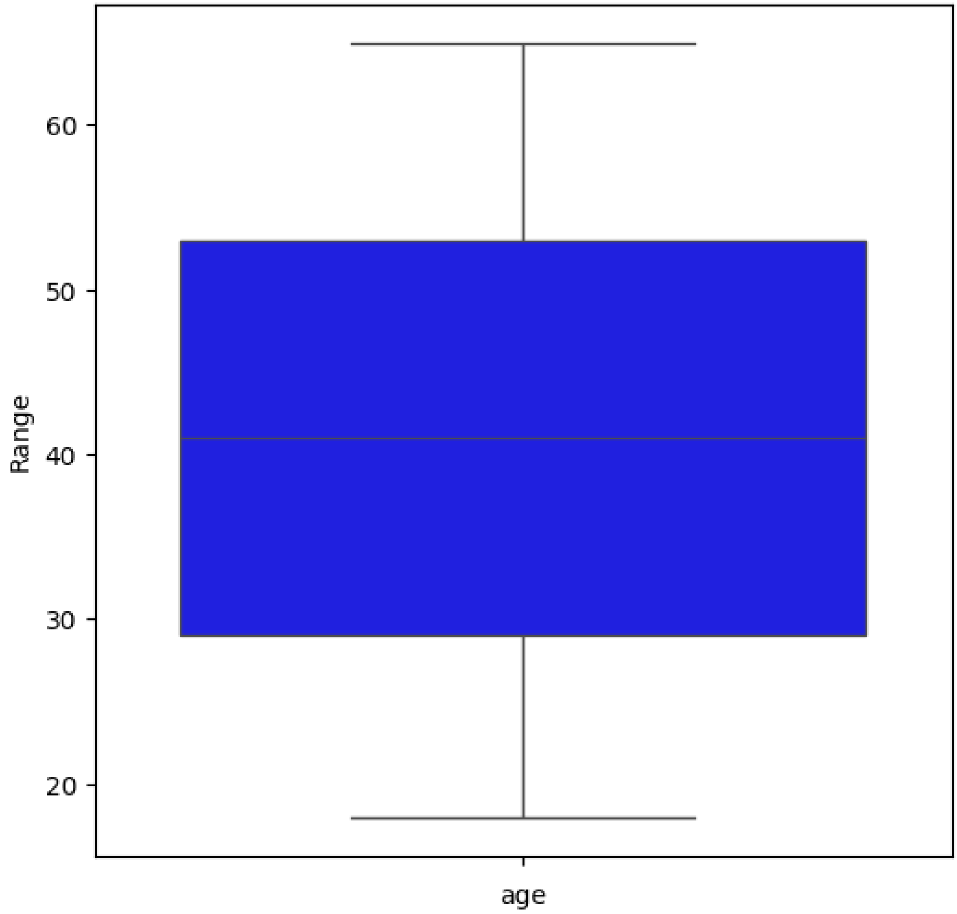
```
age                          float64
gender                        object
bmi                          float64
children                       int64
smoker                        object
region                        object
medical_history               object
family_medical_history        object
exercise_frequency            object
occupation                    object
coverage_level                object
charges                      float64
dtype: object
```

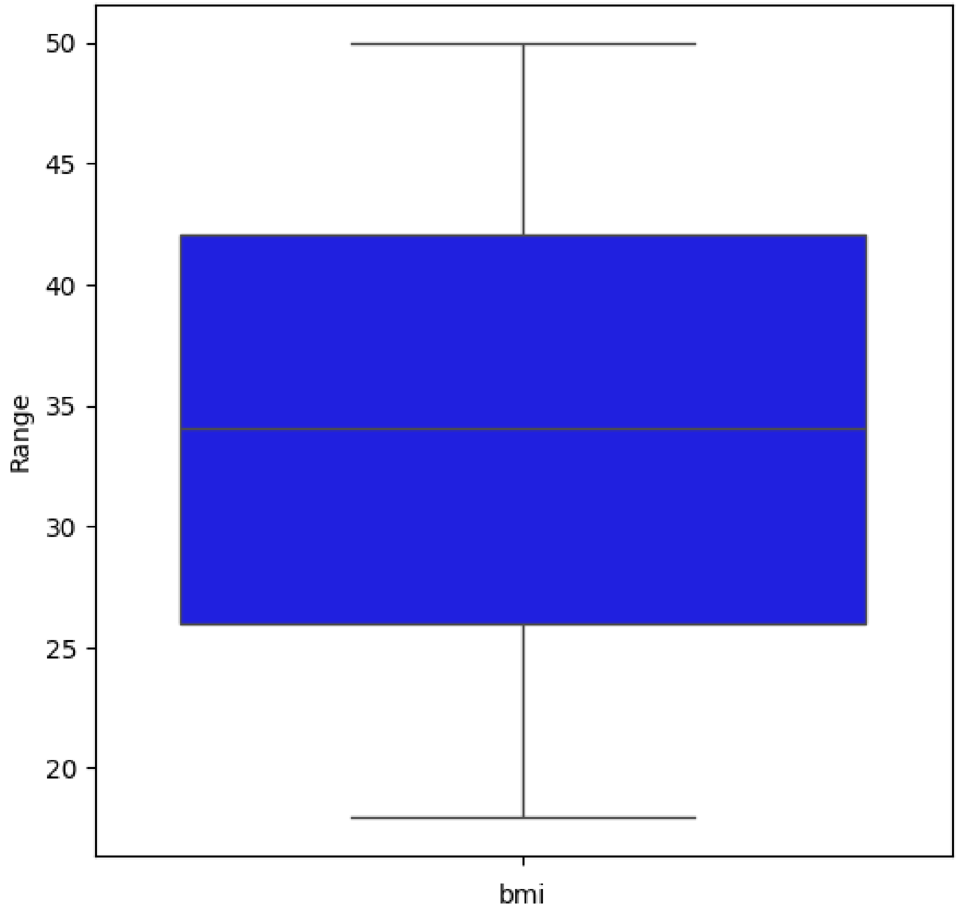
CHECKING FOR OUTLAYERS

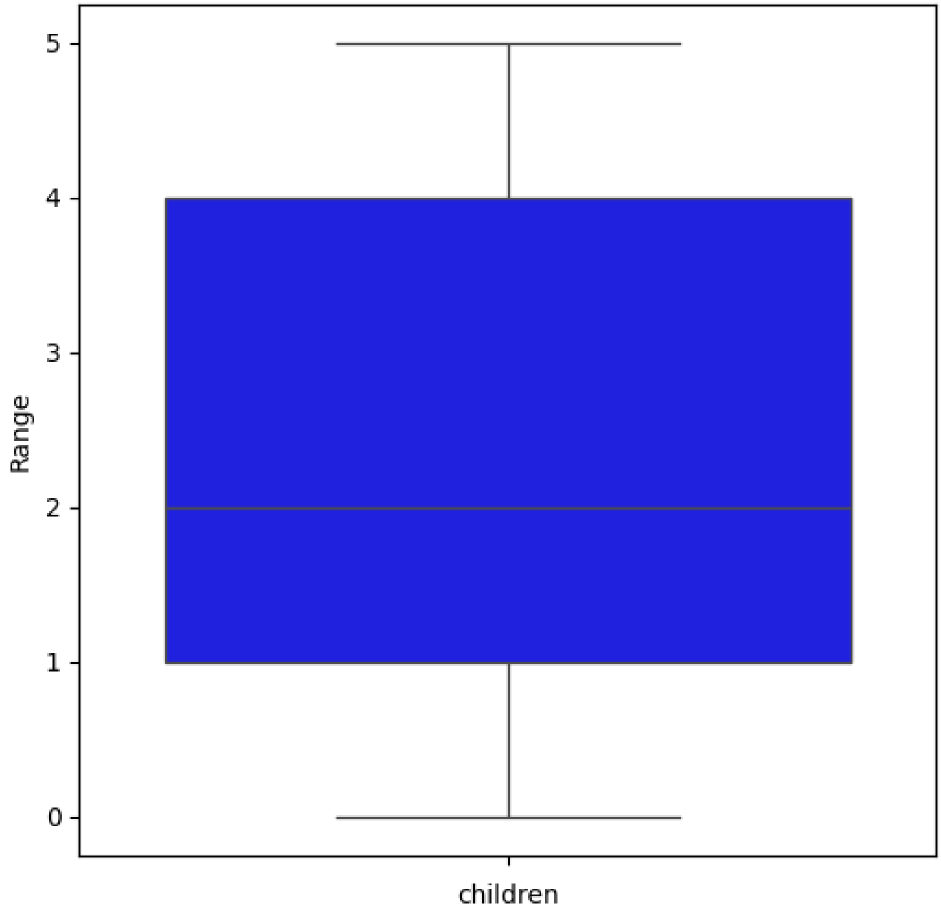
CHECKING FOR OUTLAYERS

In [ ]:

```
for i in['age','bmi','children']:
  plt.figure(figsize=(6,6))
  sns.boxplot(df[i],color='blue')
  plt.xlabel(i)
  plt.ylabel('Range')
```







LABEL ENCODING

In [ ]:

```
from sklearn.preprocessing import LabelEncoder
lb=LabelEncoder()
df['gender']=lb.fit_transform(df['gender'])
df['smoker']=lb.fit_transform(df['smoker'])
df['region']=lb.fit_transform(df['region'])
df['medical_history']=lb.fit_transform(df['medical_history'])
df['family_medical_history']=lb.fit_transform(df['family_medical_history'])
df['exercise_frequency']=lb.fit_transform(df['exercise_frequency'])
df['occupation']=lb.fit_transform(df['occupation'])
df['coverage_level']=lb.fit_transform(df['coverage_level'])
```

In [ ]:

```
df.dtypes
```

Out[ ]:

```
age                         float64
gender                        int64
bmi                         float64
children                      int64
smoker                        int64
region                        int64
medical_history               int64
family_medical_history        int64
exercise_frequency            int64
occupation                    int64
coverage_level                int64
charges                     float64
dtype: object
```

In [ ]:

```
df.corr()
```

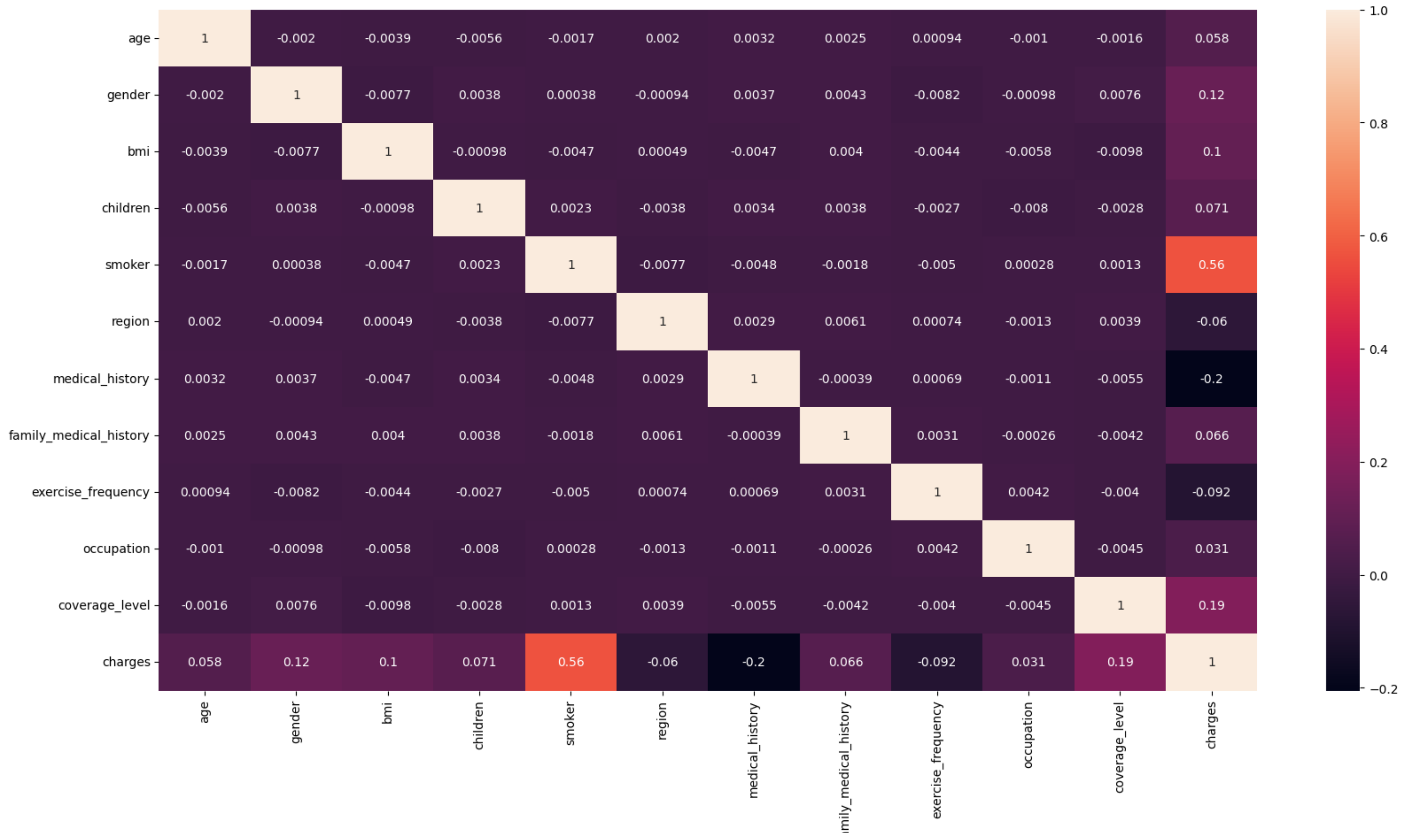
Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	ex
age	1.000000	-0.002018	-0.003922	-0.005598	-0.001661	0.001994	0.003179	0.002511	
gender	-0.002018	1.000000	-0.007656	0.003792	0.000379	-0.000945	0.003729	0.004335	
bmi	-0.003922	-0.007656	1.000000	-0.000985	-0.004712	0.000494	-0.004723	0.003979	
children	-0.005598	0.003792	-0.000985	1.000000	0.002281	-0.003815	0.003419	0.003825	
smoker	-0.001661	0.000379	-0.004712	0.002281	1.000000	-0.007702	-0.004768	-0.001839	
region	0.001994	-0.000945	0.000494	-0.003815	-0.007702	1.000000	0.002917	0.006080	
medical_history	0.003179	0.003729	-0.004723	0.003419	-0.004768	0.002917	1.000000	-0.000393	
family_medical_history	0.002511	0.004335	0.003979	0.003825	-0.001839	0.006080	-0.000393	1.000000	
exercise_frequency	0.000935	-0.008181	-0.004406	-0.002725	-0.005001	0.000743	0.000687	0.003076	
occupation	-0.001015	-0.000977	-0.005779	-0.007962	0.000281	-0.001285	-0.001140	-0.000263	
coverage_level	-0.001558	0.007594	-0.009787	-0.002808	0.001280	0.003920	-0.005529	-0.004174	
charges	0.057656	0.116691	0.101623	0.070588	0.564261	-0.059501	-0.204960	0.065676	

CORELATION

In [ ]:

```
plt.figure(figsize=(20,10))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



In [ ]:

```
df_corr=df.corr()
df_corr
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	ex
age	1.000000	-0.002018	-0.003922	-0.005598	-0.001661	0.001994	0.003179	0.002511	
gender	-0.002018	1.000000	-0.007656	0.003792	0.000379	-0.000945	0.003729	0.004335	
bmi	-0.003922	-0.007656	1.000000	-0.000985	-0.004712	0.000494	-0.004723	0.003979	
children	-0.005598	0.003792	-0.000985	1.000000	0.002281	-0.003815	0.003419	0.003825	
smoker	-0.001661	0.000379	-0.004712	0.002281	1.000000	-0.007702	-0.004768	-0.001839	
region	0.001994	-0.000945	0.000494	-0.003815	-0.007702	1.000000	0.002917	0.006080	
medical_history	0.003179	0.003729	-0.004723	0.003419	-0.004768	0.002917	1.000000	-0.000393	
family_medical_history	0.002511	0.004335	0.003979	0.003825	-0.001839	0.006080	-0.000393	1.000000	
exercise_frequency	0.000935	-0.008181	-0.004406	-0.002725	-0.005001	0.000743	0.000687	0.003076	
occupation	-0.001015	-0.000977	-0.005779	-0.007962	0.000281	-0.001285	-0.001140	-0.000263	
coverage_level	-0.001558	0.007594	-0.009787	-0.002808	0.001280	0.003920	-0.005529	-0.004174	
charges	0.057656	0.116691	0.101623	0.070588	0.564261	-0.059501	-0.204960	0.065676	

In [ ]:

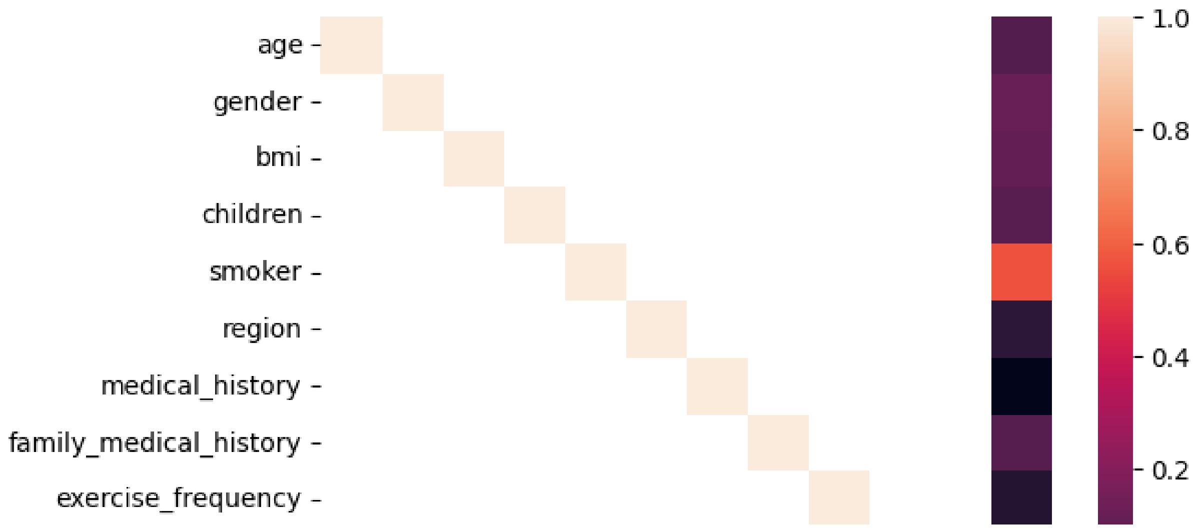
```
df_corrr=df_corr[abs(df_corr)>0.05]
```

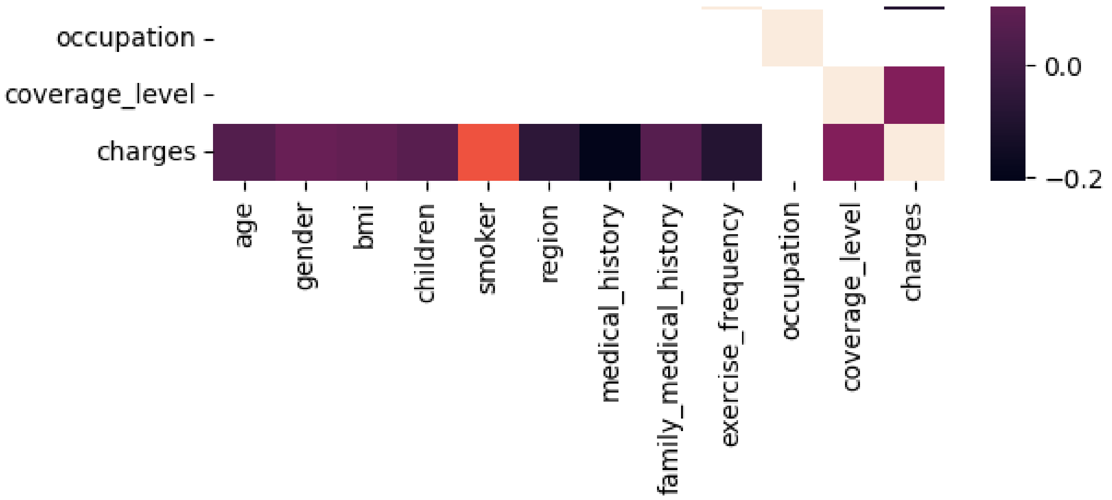
In [ ]:

```
sns.heatmap(df_corrr)
```

Out[ ]:

<Axes: >





In [ ]:

```
df.columns
```

Out[ ]:

```
Index(['age', 'gender', 'bmi', 'children', 'smoker', 'region',
       'medical_history', 'family_medical_history', 'exercise_frequency',
       'occupation', 'coverage_level', 'charges'],
      dtype='object')
```

In [ ]:

```
input=['age','gender','bmi','children','smoker','region','medical_history','family_medica
l_history','exercise_frequency','coverage_level']
x1=df[input]
x1
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	coverage_le
0	46.0	1	21.45	5	1	2	0	0	1	
1	25.0	0	25.38	2	1	1	0	2	2	
2	38.0	1	44.88	2	1	3	2	2	2	
3	25.0	1	19.89	0	0	1	2	0	3	
4	49.0	1	38.21	3	1	1	0	2	3	
...	...	...	...	...	...	...	...	...	...	
49995	29.0	1	37.91	4	0	0	1	0	0	
49996	39.0	0	20.57	1	0	0	2	2	0	
49997	23.0	0	37.22	4	1	0	1	2	2	
49998	65.0	1	45.35	0	1	3	2	0	2	
49999	22.0	0	27.26	1	0	2	2	0	0	

50000 rows × 10 columns

In [ ]:

```
x1.ndim
```

Out[ ]:

2

In [ ]:

```
y1=df.iloc[:,-1]
```

```
y1
```

Out[ ]:

```
0        20460.307669
1        20390.899218
2        20204.476302
3        11789.029843
4        19268.309838
             ...     
49995    18515.139201
49996    15486.399455
49997    19650.342574
49998    12956.013072
49999    16700.823932
Name: charges, Length: 50000, dtype: float64
```

In [ ]:

```
y1.ndim
```

Out[ ]:

```
1
```

SPLITING INTO TRAIN & TEST DATA

In [ ]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x1,y1,test_size=0.30,random_state=42)
x_train
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	coverage_le
38094	54.0	0	35.06	1	1	0	1	1	0	
40624	41.0	0	44.71	0	0	0	2	0	3	
49425	43.0	1	30.91	1	0	3	2	0	3	
35734	41.0	0	43.80	0	0	3	2	2	3	
41708	34.0	1	47.84	5	0	1	2	2	3	
...	...	...	...	...	...	...	...	...	...	
11284	37.0	0	24.42	3	0	0	2	1	2	
44732	27.0	1	40.23	1	1	2	1	0	2	
38158	55.0	1	26.03	4	0	3	2	2	2	
860	30.0	1	36.11	5	1	2	1	0	2	
15795	22.0	0	21.54	2	1	1	0	0	3	

35000 rows × 10 columns

In [ ]:

```
x_test
```

Out[ ]:

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	coverage_le
33553	36.0	1	37.17	3	0	3	2	0	3	
9427	54.0	0	20.32	1	1	2	1	1	3	
199	21.0	0	27.96	2	0	3	1	1	3	

	age	gender	bmi	children	smoker	region	medical_history	family_medical_history	exercise_frequency	coverage_le
12447	23.0	0	48.12	0	1	3	2	0	0	
39489	50.0	1	30.44	5	0	0	0	2	2	
...	...	...	...	...	...	...	...	...	...	
15168	22.0	1	26.53	4	1	1	0	2	0	
49241	29.0	0	26.34	4	1	1	1	0	2	
39317	28.0	1	38.29	3	0	3	2	2	2	
42191	20.0	1	37.53	5	1	2	0	2	1	
15109	50.0	1	38.30	1	0	2	2	0	2	

15000 rows × 10 columns

In [ ]:

```
y_train
```

Out[ ]:

```
38094    25523.927518
40624     8972.521103
49425    13699.538103
35734     9393.003535
41708    11730.434150
             ...
11284    14812.736306
44732    21948.215336
38158    11573.889809
860      21589.649719
15795    19940.878646
Name: charges, Length: 35000, dtype: float64
```

In [ ]:

```
y_test
```

Out[ ]:

```
33553     8951.667599
9427     20492.267121
199      18299.928568
12447    19579.316635
39489    15213.447970
             ...
15168    17760.646489
49241    16215.908584
39317    14781.926595
42191    18590.543582
15109    10994.410156
Name: charges, Length: 15000, dtype: float64
```

In [ ]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 50000 entries, 0 to 49999
Data columns (total 12 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   age                     50000 non-null  float64
 1   gender                  50000 non-null  int64
 2   bmi                     50000 non-null  float64
 3   children                50000 non-null  int64
 4   smoker                  50000 non-null  int64
 5   region                  50000 non-null  int64
 6   medical_history         50000 non-null  int64
 7   family_medical_history  50000 non-null  int64
 8   exercise_frequency      50000 non-null  int64
 9   occupation              50000 non-null  int64
```

```
 9   occupation             50000 non-null  int64
 10  coverage_level         50000 non-null  int64
 11  charges                50000 non-null  float64
dtypes: float64(3), int64(9)
memory usage: 4.6 MB
```

MODEL CREATION

In [ ]:

```
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.metrics import mean_absolute_error,mean_absolute_percentage_error,mean_squar
ed_error,r2_score
dt=DecisionTreeRegressor(criterion='squared_error',random_state=42)
rf=RandomForestRegressor(n_estimators=100,random_state=42)
reg=LinearRegression()
knn=KNeighborsRegressor(n_neighbors=7)
sv=SVR()
lst=[dt,rf,reg,knn,sv]
```

PERFORMNACE EVALUATION

In [ ]:

```
for i in lst:
  i.fit(x_train,y_train)
  y_pred=i.predict(x_test)
  print('MODEL IS',i)
  print('MAE :',mean_absolute_error(y_test,y_pred))
  print('MAPE :',mean_absolute_percentage_error(y_test,y_pred))
  print('MSE :',mean_squared_error(y_test,y_pred))
  print('r2_score :',r2_score(y_test,y_pred))
  mse=mean_squared_error(y_test,y_pred)
  print('RMSE :',np.sqrt(mse))
  print('*'*200)
```

```
MODEL IS DecisionTreeRegressor(random_state=42)
MAE : 1381.2068962241317
MAPE : 0.09130470163495656
MSE : 3012878.463874638
r2_score : 0.8477232490017621
RMSE : 1735.7645185550482
*****************************************************************************************
*****************************************************************************************
**********************
MODEL IS RandomForestRegressor(random_state=42)
MAE : 1034.0758563952268
MAPE : 0.06855222870440443
MSE : 1630026.392143114
r2_score : 0.9176152884979893
RMSE : 1276.724869399478
*****************************************************************************************
*****************************************************************************************
**********************
MODEL IS LinearRegression()
MAE : 2681.0255422384844
MAPE : 0.17481470834479643
MSE : 10898786.270720255
r2_score : 0.44915409531815165
RMSE : 3301.330984727259
*****************************************************************************************
*****************************************************************************************
**********************
MODEL IS KNeighborsRegressor(n_neighbors=7)
MAE : 3121.6996603739162
MAPE : 0.21550945923665693
MSE : 14826123.415064
```

```
r2_score : 0.2506588199241798
RMSE : 3850.4705446300973
*******************************************************************************************
*******************************************************************************************
**********************
MODEL IS SVR()
MAE : 3601.982302222389
MAPE : 0.24471642376000224
MSE : 19647690.027430084
r2_score : 0.006967443967224707
RMSE : 4432.571491519351
*******************************************************************************************
*******************************************************************************************
**********************
```

PREDICTION USING RANDOM FOREST

In [ ]:

```
rf.predict([[38.0,0,37.17,3,0,3,2,0,3,0]])
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning:

X does not have valid feature names, but RandomForestRegressor was fitted with feature na
mes
```

Out[ ]:

```
array([9271.28049867])
```

In [ ]:

```
df1=pd.DataFrame({'Actual value':y_test,'Predicted Value':y_pred,'Difference':y_test-y_p
red})
df1
```

Out[ ]:

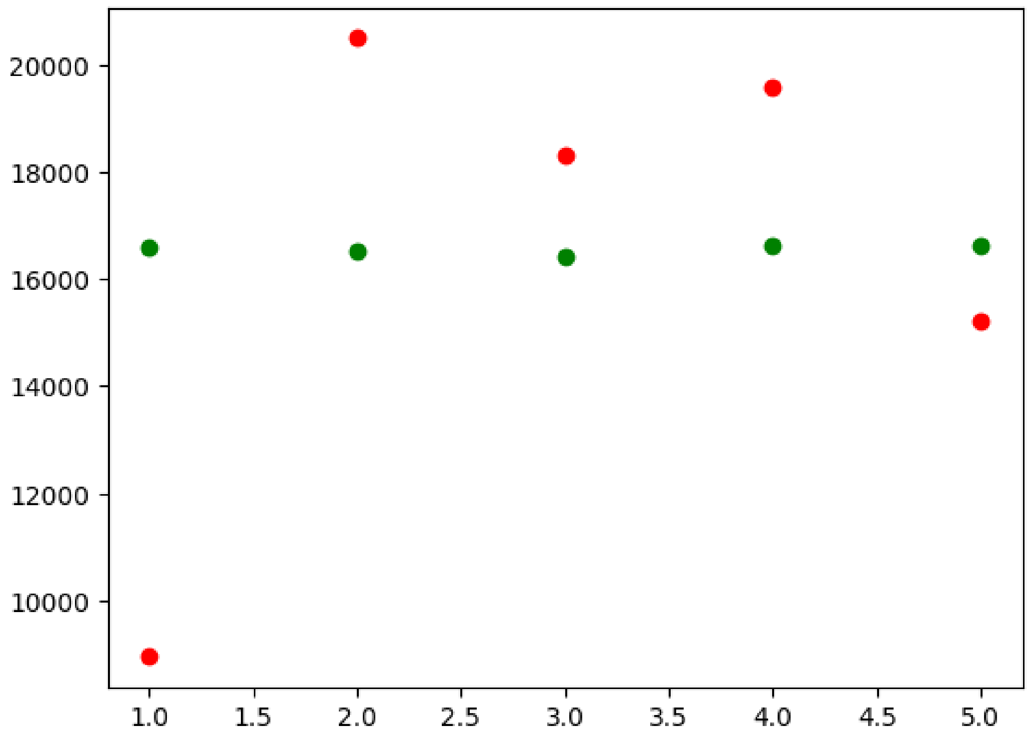
	Actual value	Predicted Value	Difference
33553	8951.667599	16587.535369	-7635.867770
9427	20492.267121	16526.897022	3965.370099
199	18299.928568	16413.685903	1886.242665
12447	19579.316635	16629.066195	2950.250440
39489	15213.447970	16628.942117	-1415.494148
...	...	...	...
15168	17760.646489	16420.009138	1340.637351
49241	16215.908584	16438.202428	-222.293844
39317	14781.926595	16556.357901	-1774.431306
42191	18590.543582	16527.240178	2063.303404
15109	10994.410156	16691.893930	-5697.483773

15000 rows × 3 columns

In [ ]:

```
import plotly.graph_objects as go
x_num=list(range(1,6))
y_num=list(range(1,6))
y_pred_df=pd.DataFrame(y_pred)
y_test_df=pd.DataFrame(y_test)
y_pred_plot=y_pred_df.head(5)
y_test_plot=y_test_df.head(5)
```

```
plt.scatter(x_num,y_pred_plot,color='g')
plt.scatter(y_num,y_test_plot,color='r')
plt.show()
```



HYPER PARAMETER TUNING

In [ ]:

```
dt1 = DecisionTreeRegressor(random_state=42)
```

In [ ]:

```
from sklearn.model_selection import GridSearchCV
```

In [ ]:

```
param_grid = {'max_depth': [None, 10, 20, 30],'min_samples_split': [2, 5, 10],'min_sampl
es_leaf': [1, 2, 4]}
grid_search = GridSearchCV(estimator=dt1, param_grid=param_grid, cv=3, n_jobs=-1, verbos
e=2)
grid_search.fit(x_train, y_train)
print("Best parameters found: ", grid_search.best_params_)
```

```
Fitting 3 folds for each of 36 candidates, totalling 108 fits
Best parameters found:  {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 10}
```

In [ ]:

```
dt2=DecisionTreeRegressor(max_depth=10,min_samples_leaf=4,min_samples_split=10)
dt2.fit(x_train,y_train)
y_pred1=dt2.predict(x_test)
y_pred1
```

Out[ ]:

```
array([ 9375.25296609, 21126.87411411, 18287.77587474, ...,
       15142.77877829, 17369.69820069, 10076.746368  ])
```

IMPROVED r2_score

In [ ]:

```
print('r2_score :',r2_score(y_test,y_pred1))
```

```
r2_score : 0.9069959179421905
```

In conclusion, the model i have developed has shown impressive results, achieving an accuracy of 91%. This high level of accuracy is due to choosing the right features, using advanced algorithms, and thoroughly testing the model. ***THANK YOU ALL***