In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OrdinalEncoder
import plotly.express as px
import scipy.stats as stats
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import Ridge,LinearRegression
from sklearn.model_selection import KFold,StratifiedKFold,train_test_split,GridSearchCV
from sklearn.ensemble import GradientBoostingRegressor
import datetime as dt
from sklearn.metrics import mean_squared_error

import warnings
warnings.filterwarnings('ignore')
```

DATA SCIENCE

In [2]:
```
df_hos = pd.read_csv("Hospitalisation details.csv")
df_hos
```

Out[2]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013
...	...	...	...	...	...	...	...	...	...
2338	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012
2339	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024
2340	Id3	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012
2341	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013
2342	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013

2343 rows × 9 columns

In [3]:
```
df_hos.shape
```

Out[3]:
```
(2343, 9)
```

In [4]:
```
df_med = pd.read_csv("Medical Examinations.csv")
df_med
```

Out[4]:

	Customer ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeries	smoker
0	Id1	47.410	7.47	No	No	No	No major surgery	yes
1	Id2	30.360	5.77	No	No	No	No major surgery	yes
2	Id3	34.485	11.87	yes	No	No	2	yes
3	Id4	38.095	6.05	No	No	No	No major surgery	yes
4	Id5	35.530	5.45	No	No	No	No major surgery	yes
...	...	...	...	...	...	...	...	...
2330	Id2331	22.340	5.57	No	No	No	1	No
2331	Id2332	17.700	6.28	No	No	No	1	No
2332	Id2333	16.470	6.35	No	No	Yes	1	No
2333	Id2334	17.600	4.39	No	No	No	1	No
2334	Id2335	17.580	4.51	No	No	No	1	No

2335 rows × 8 columns

In [5]:
```
df_med.shape
```

Out[5]:
```
(2335, 8)
```

In [6]:
```
df_names = pd.read_excel("Names.xlsx")
df_names
```

Out[6]:

	Customer ID	name
0	Id1	Hawks, Ms. Kelly
1	Id2	Lehner, Mr. Matthew D
2	Id3	Lu, Mr. Phil
3	Id4	Osborne, Ms. Kelsey
4	Id5	Kadala, Ms. Kristyn
...	...	...
2330	Id2331	Brietzke, Mr. Jordan
2331	Id2332	Riveros Gonzalez, Mr. Juan D. Sr.
2332	Id2333	Albano, Ms. Julie
2333	Id2334	Rosendahl, Mr. Evan P
2334	Id2335	German, Mr. Aaron K

2335 rows × 2 columns

Merging all datasets

In [7]:
```
df = pd.merge(df_hos,df_med,on=['Customer ID'],how='inner')
df
```

Out[7]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeı
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	No	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	No	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	No	No	Yes	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	No	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	No	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	No	No	No	No major surg
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	No	No	No	No major surg
2332	Id3	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012	34.485	11.87	yes	No	No	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	No	No	No	No major surg
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	No	No	No	No major surg

2335 rows × 16 columns

In [8]:
```
df = pd.merge(df,df_names,on='Customer ID',how='left')
df
```

Out[8]:

	Customer ID	year	month	date	children	charges	Hospital tier	City tier	State ID	BMI	HBA1C	Heart Issues	Any Transplants	Cancer history	NumberOfMajorSurgeı
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	No	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	No	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	No	No	Yes	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	No	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	No	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	No	No	No	No major surg
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	No	No	No	No major surg
2332	Id3	1970	?	11	3	60021.40	tier - 1	tier - 1	R1012	34.485	11.87	yes	No	No	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	No	No	No	No major surg
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	No	No	No	No major surg

2335 rows × 17 columns

In [9]:
```
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df.columns
```

Out[9]:
```
Index(['customer_id', 'year', 'month', 'date', 'children', 'charges',
       'hospital_tier', 'city_tier', 'state_id', 'bmi', 'hba1c',
       'heart_issues', 'any_transplants', 'cancer_history',
       'numberofmajorsurgeries', 'smoker', 'name'],
      dtype='object')
```

- **All the data is imported and merged in one dataset.**

Checking for missing values

In [10]:
```
df.isnull().sum()
```

Out[10]:
```
customer_id              0
year                     0
month                    0
date                     0
children                 0
charges                  0
hospital_tier            0
city_tier                0
state_id                 0
bmi                      0
hba1c                    0
heart_issues             0
any_transplants          0
cancer_history           0
numberofmajorsurgeries   0
smoker                   0
name                     0
dtype: int64
```

- **There is no missing values in the data.**

Deleting rows with trivial values

In [11]:
```
(df == '?').sum()
```

Out[11]:
```
customer_id              0
year                     2
month                    3
date                     0
children                 0
charges                  0
hospital_tier            1
city_tier                1
state_id                 2
bmi                      0
hba1c                    0
heart_issues             0
any_transplants          0
cancer_history           0
numberofmajorsurgeries   0
smoker                   2
name                     0
dtype: int64
```

In [12]:
```
percentage = (df == '?').sum(axis=1)/df.shape[1]*100
percentage[percentage>0]
```

Out[12]:
```
11       5.882353
13       5.882353
17      11.764706
542      5.882353
1046     5.882353
1049     5.882353
1700     5.882353
1775     5.882353
2165     5.882353
2332     5.882353
dtype: float64
```

In [13]:
```
percentage[percentage>0].index
```

Out[13]:
```
Index([11, 13, 17, 542, 1046, 1049, 1700, 1775, 2165, 2332], dtype='int64')
```

In [14]:
```
main_df = df.drop(index = percentage[percentage>0].index)
main_df
```

Out[14]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	hba1c	heart_issues	any_transplants	cancer_h
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	No	No	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	No	No	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	No	No	
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	No	No	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	No	No	
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	No	No	

2325 rows × 17 columns

In [15]:
```
main_df.shape
```

Out[15]:
```
(2325, 17)
```

In [16]:
```
main_df.isnull().sum()
```

Out[16]:
```
customer_id              0
year                     0
month                    0
date                     0
children                 0
charges                  0
hospital_tier            0
city_tier                0
state_id                 0
bmi                      0
hba1c                    0
heart_issues             0
any_transplants          0
cancer_history           0
numberofmajorsurgeries   0
smoker                   0
name                     0
dtype: int64
```

- **All the rows containing trivial values have been deleted.**

Variables

- **heart_issues,any_transplants,cancer_history,smoker,state_id are nominal categorical whereas city tier and hospital tier are ordinal categorical. For nominal categorical, dummy variable are created while number are assigned to ordinal categorical based on rank.**

In [17]:
```
ordinal = OrdinalEncoder(categories= [['tier - 1', 'tier - 2', 'tier - 3'],['tier - 1', 'tier - 2', 'tier - 3']])
main_df[['hospital_tier_ord','city_tier_ord']] = ordinal.fit_transform(main_df[['hospital_tier','city_tier']])
```

In [18]:
```
main_df
```

Out[18]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	hba1c	heart_issues	any_transplants	cancer_h
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	4.51	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	4.39	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	6.35	No	No	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	6.28	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	5.57	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	6.59	No	No	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	5.45	No	No	
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	6.05	No	No	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	5.77	No	No	
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	7.47	No	No	

2325 rows × 19 columns

Conditional dummy variables

In [19]:
```
states = main_df.state_id.value_counts()
states
```

Out[19]:
```
state_id
R1013    609
R1011    574
R1012    572
R1024    159
R1026     84
R1021     70
R1016     64
R1025     40
R1023     38
R1017     36
R1019     26
R1022     14
R1014     13
R1015     11
R1018      9
R1020      6
Name: count, dtype: int64
```

In [20]:
```
states[:3].index
```

Out[20]:
```
Index(['R1013', 'R1011', 'R1012'], dtype='object', name='state_id')
```

In [21]:
```
for i in states[:3].index:
    var = 'state_id_' +i
    print(var)
    main_df[var] = 0
    main_df.loc[main_df.state_id == i,var] = 1
```

```
state_id_R1013
state_id_R1011
state_id_R1012
```

In [22]:
```
main_df
```

Out[22]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	any_transplants	cancer_history	numberofm
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	No	Yes	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	No	No	No
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	No	No	No
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	No	No	No
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	No	No	No
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	No	No	No

2325 rows × 22 columns

Variable cleanup

In [23]:
```
main_df.numberofmajorsurgeries.unique()
```

Out[23]:
```
array(['1', 'No major surgery', '2', '3'], dtype=object)
```

In [24]:
```
main_df['numberofmajorsurgeries'] = np.where(main_df['numberofmajorsurgeries'] == 'No major surgery',0,main_df['number
main_df['numberofmajorsurgeries'] = main_df['numberofmajorsurgeries'].astype(int)
```

In [25]:
```
main_df
```

Out[25]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	any_transplants	cancer_history	numberofm
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	No	No	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	No	No	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	No	Yes	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	No	No	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	No	No	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	No	No	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	No	No	
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	No	No	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	No	No	
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	No	No	

2325 rows × 22 columns

In [26]:
```
main_df.numberofmajorsurgeries.unique()
```

Out[26]:
```
array([1, 0, 2, 3])
```

Calculating age

In [27]:
```
main_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2325 entries, 0 to 2334
Data columns (total 22 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   customer_id             2325 non-null   object 
 1   year                    2325 non-null   object 
 2   month                   2325 non-null   object 
 3   date                    2325 non-null   int64  
 4   children                2325 non-null   int64  
 5   charges                 2325 non-null   float64
 6   hospital_tier           2325 non-null   object 
 7   city_tier               2325 non-null   object 
 8   state_id                2325 non-null   object 
 9   bmi                     2325 non-null   float64
 10  hba1c                   2325 non-null   float64
 11  heart_issues            2325 non-null   object 
 12  any_transplants         2325 non-null   object 
 13  cancer_history          2325 non-null   object 
 14  numberofmajorsurgeries  2325 non-null   int32  
 15  smoker                  2325 non-null   object 
 16  name                    2325 non-null   object 
 17  hospital_tier_ord       2325 non-null   float64
 18  city_tier_ord           2325 non-null   float64
 19  state_id_R1013          2325 non-null   int64  
 20  state_id_R1011          2325 non-null   int64  
 21  state_id_R1012          2325 non-null   int64  
dtypes: float64(5), int32(1), int64(5), object(11)
memory usage: 408.7+ KB
```

- **Since year is given in string it needs to be changed to int to calculate the age.**

In [28]:
```
main_df['year'] = main_df['year'].astype(int)
```

In [29]:
```
main_df['Age'] = 2024 - main_df['year']
```

In [30]:
```
main_df
```

Out[30]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	cancer_history	numberofmajorsurgeries	sm
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	No	1	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	No	1	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	Yes	1	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	No	1	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	No	1	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	No	0	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	No	0	
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	No	0	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	No	0	
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	No	0	

2325 rows × 23 columns

Gender

In [31]:
```
main_df.name
```

Out[31]:
```
0                     German, Mr.  Aaron K
1                   Rosendahl, Mr.  Evan P
2                       Albano, Ms.  Julie
3       Riveros Gonzalez, Mr.  Juan D. Sr.
4                    Brietzke, Mr.  Jordan
                         ...
2329                Baker, Mr.  Russell B.
2330                  Kadala, Ms.  Kristyn
2331                  Osborne, Ms.  Kelsey
2333                Lehner, Mr.  Matthew D
2334                     Hawks, Ms.  Kelly
Name: name, Length: 2325, dtype: object
```

In [32]:
```
main_df['title'] = main_df.name.str.split('[,.]').str[1].str.strip()
main_df
```

Out[32]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	numberofmajorsurgeries	smoker	name
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	1	No	German Mr. Aaron K
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	1	No	Rosendahl Mr. Evan F
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	1	No	Albano Ms. Julie
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	1	No	Riveros Gonzalez Mr. Juan D. Sr
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	1	No	Brietzke Mr. Jordan
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	0	yes	Baker, Mr Russell B
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	0	yes	Kadala Ms. Kristyn
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	0	yes	Osborne Ms. Kelsey
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	0	yes	Lehner, Mr Matthew D
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	0	yes	Hawks Ms. Kelly

2325 rows × 24 columns

In [33]:
```
main_df['gender'] = np.where(main_df['title'] == 'Mr','Male','Female')
main_df
```

Out[33]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	smoker	name	hospital_tier_ord	city_t
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	No	German, Mr. Aaron K	1.0	
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	No	Rosendahl, Mr. Evan P	1.0	
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	No	Albano, Ms. Julie	1.0	
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	No	Riveros Gonzalez, Mr. Juan D. Sr.	2.0	
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	No	Brietzke, Mr. Jordan	2.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	yes	Baker, Mr. Russell B.	0.0	
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	yes	Kadala, Ms. Kristyn	0.0	
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	yes	Osborne, Ms. Kelsey	0.0	
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	yes	Lehner, Mr. Matthew D	1.0	
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	yes	Hawks, Ms. Kelly	0.0	

2325 rows × 25 columns

In [34]:
```
main_df.loc[main_df['title'] == 'Mrs']
```

Out[34]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	smoker	name	hospital_tier_ord	city_tie
24	Id2311	2001	Aug	19	0	964.71	tier - 3	tier - 2	R1013	25.19	...	No	Keys, Mrs. Kathleen	2.0	
172	Id2163	2004	Dec	27	0	1863.45	tier - 3	tier - 1	R1025	27.06	...	No	Stanislav, Mrs. Grace H	2.0	
197	Id2138	2004	Jun	12	0	2094.10	tier - 3	tier - 2	R1025	27.74	...	No	Padula, Mrs. Lauren	2.0	
328	Id2007	1993	Sep	25	0	3162.02	tier - 2	tier - 3	R1013	25.61	...	No	Martin, Mrs. Kristen M	1.0	
348	Id1987	2003	Dec	5	0	3300.70	tier - 2	tier - 2	R1025	30.54	...	No	Mendez-Karr, Mrs. Cynthia	1.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1790	Id545	1963	Jul	4	0	18208.34	tier - 1	tier - 2	R1026	44.20	...	No	Shigezumi, Mrs. Teiko	0.0	
1808	Id527	1963	Dec	6	0	18883.33	tier - 1	tier - 1	R1026	46.19	...	No	Hughey, Mrs. Ashley E	0.0	
1811	Id524	1963	Oct	20	0	18954.56	tier - 1	tier - 1	R1026	46.40	...	No	Rogers, Mrs. Anita L.	0.0	
1839	Id496	1966	Aug	10	0	19995.29	tier - 1	tier - 3	R1026	51.74	...	No	Oehlke, Mrs. Jessica	0.0	
1848	Id487	1962	Jul	2	0	20354.50	tier - 3	tier - 2	R1026	49.77	...	No	Argall, Mrs. Tara R	2.0	

142 rows × 25 columns

In [35]:
```
main_df.drop(['title'],axis=1)
```

Out[35]:

	customer_id	year	month	date	children	charges	hospital_tier	city_tier	state_id	bmi	...	numberofmajorsurgeries	smoker	name
0	Id2335	1992	Jul	9	0	563.84	tier - 2	tier - 3	R1013	17.580	...	1	No	German Mr. Aaror K
1	Id2334	1992	Nov	30	0	570.62	tier - 2	tier - 1	R1013	17.600	...	1	No	Rosendahl Mr. Evan F
2	Id2333	1993	Jun	30	0	600.00	tier - 2	tier - 1	R1013	16.470	...	1	No	Albano Ms. Julie
3	Id2332	1992	Sep	13	0	604.54	tier - 3	tier - 3	R1013	17.700	...	1	No	Riveros Gonzalez Mr. Juar D. Sr
4	Id2331	1998	Jul	27	0	637.26	tier - 3	tier - 3	R1013	22.340	...	1	No	Brietzke Mr. Jordar
...	...	...	...	...	...	...	...	...	...	...	...	...	...	..
2329	Id6	1962	Aug	4	0	52590.83	tier - 1	tier - 3	R1011	32.800	...	0	yes	Baker, Mr Russell B
2330	Id5	1989	Jun	19	0	55135.40	tier - 1	tier - 2	R1012	35.530	...	0	yes	Kadala Ms. Kristyr
2331	Id4	1991	Jun	6	1	58571.07	tier - 1	tier - 3	R1024	38.095	...	0	yes	Osborne Ms. Kelsey
2333	Id2	1977	Jun	8	0	62592.87	tier - 2	tier - 3	R1013	30.360	...	0	yes	Lehner, Mr Matthew D
2334	Id1	1968	Oct	12	0	63770.43	tier - 1	tier - 3	R1013	47.410	...	0	yes	Hawks Ms. Kelly

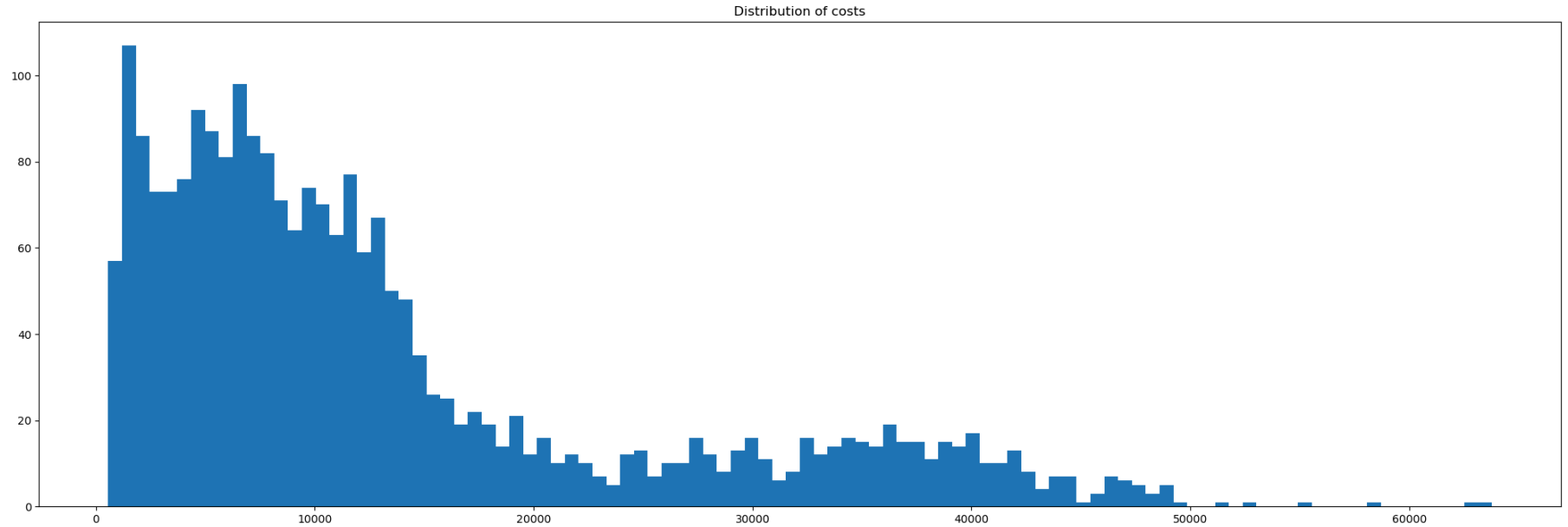
2325 rows × 24 columns

Visualisation

Distribution of charges

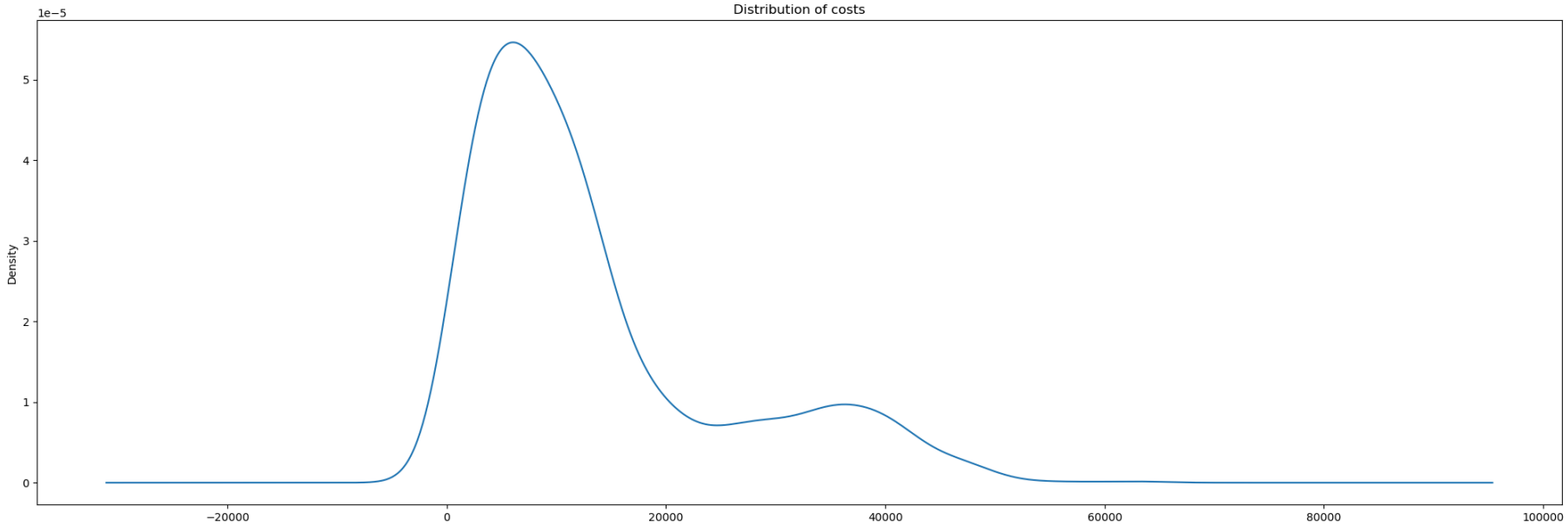
Histogram

In [36]:
```
plt.figure(figsize=(25,8))
plt.title('Distribution of costs')
plt.hist(main_df.charges, bins = 100)
plt.show()
```



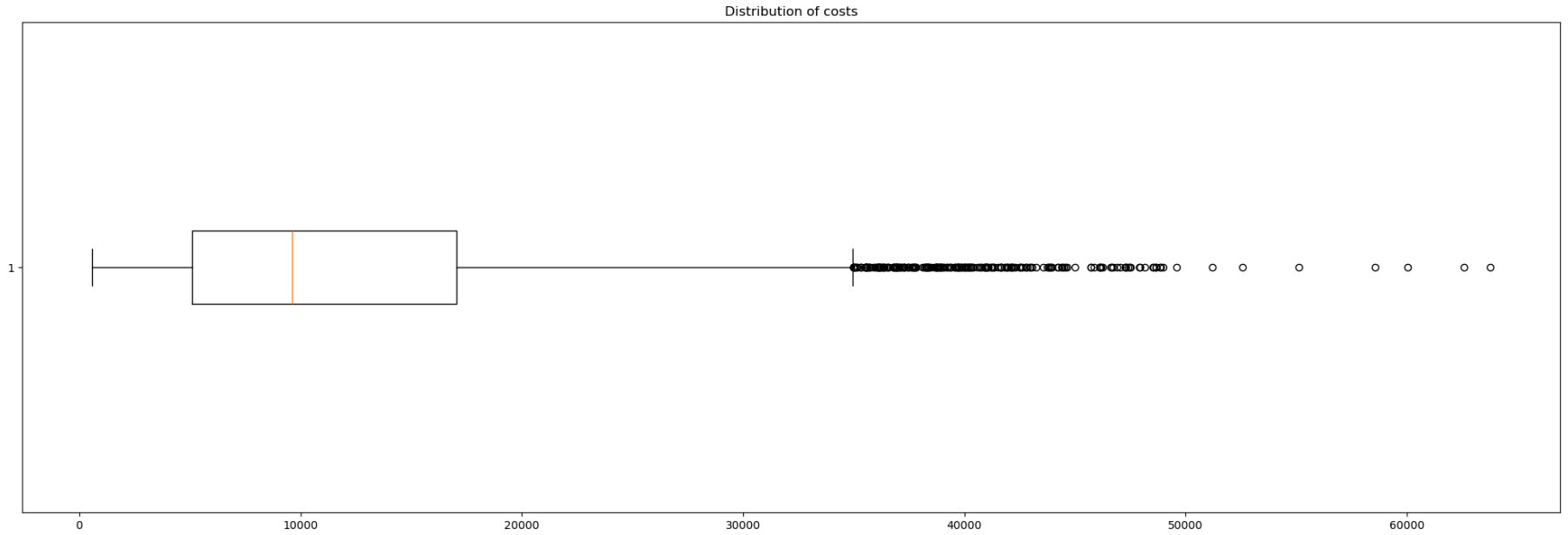
Swarm plot

In [37]:
```
plt.figure(figsize=(25,8))
plt.title('Distribution of costs')
main_df.charges.plot.kde()
plt.show()
```



Boxplot

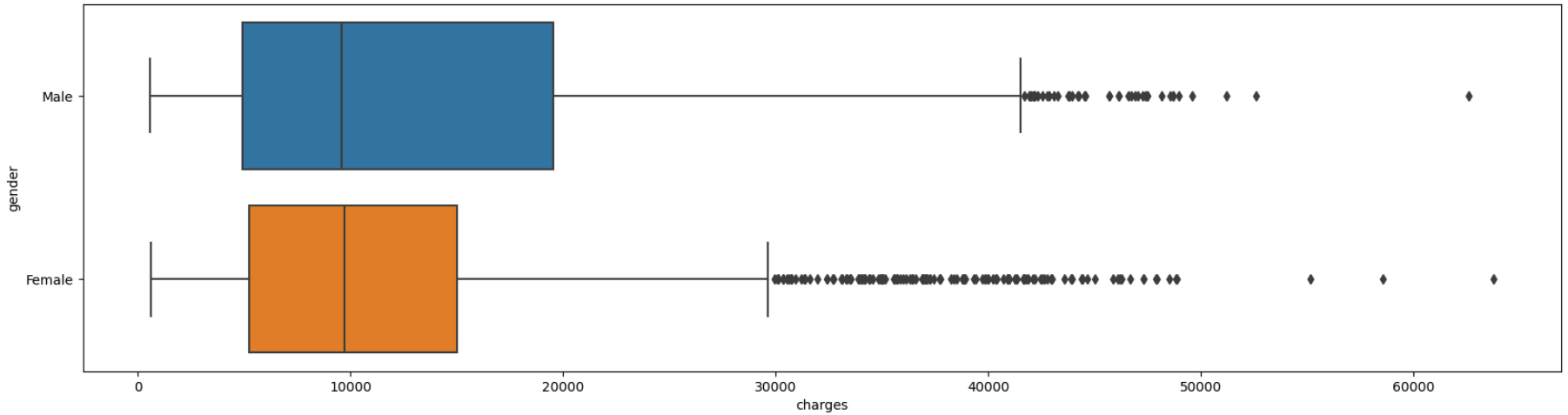
In [38]:
```
plt.figure(figsize=(25,8))
plt.title('Distribution of costs')
dis_cost = df_hos['charges']
plt.boxplot(dis_cost,vert=False)
plt.show()
```



Distribution across various variables

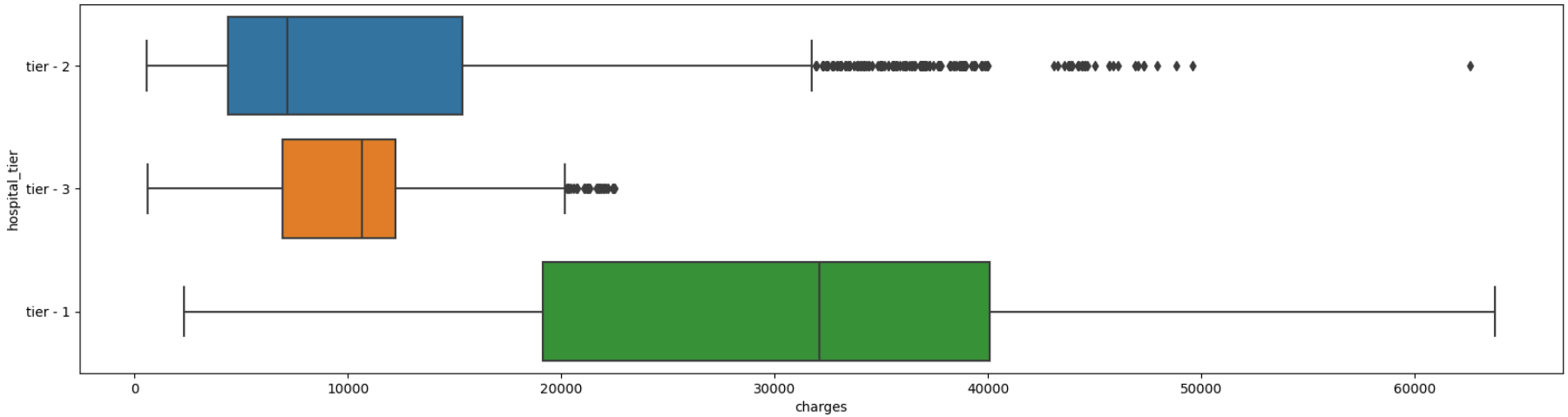
With respect to gender

In [39]:
```
plt.figure(figsize = (20,5))
sns.boxplot(x = 'charges',y = 'gender', data = main_df)
plt.show()
```



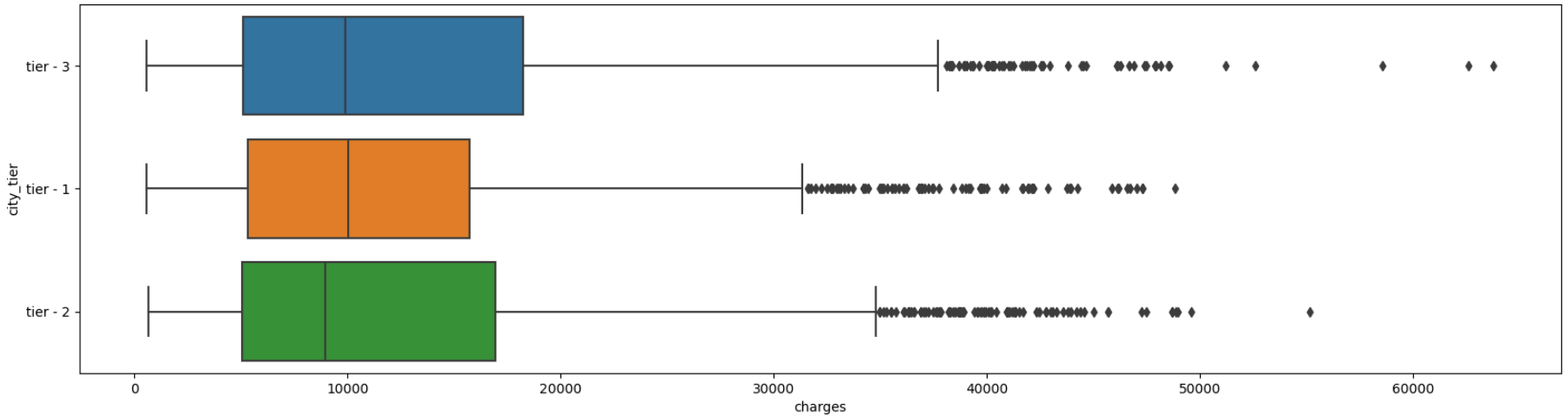
With respect to hospital tier

In [40]:
```
plt.figure(figsize = (20,5))
sns.boxplot(x = 'charges',y = 'hospital_tier', data = main_df)
plt.show()
```



With respect to city tier

In [41]:
```
plt.figure(figsize = (20,5))
sns.boxplot(x = 'charges',y = 'city_tier', data = main_df)
plt.show()
```



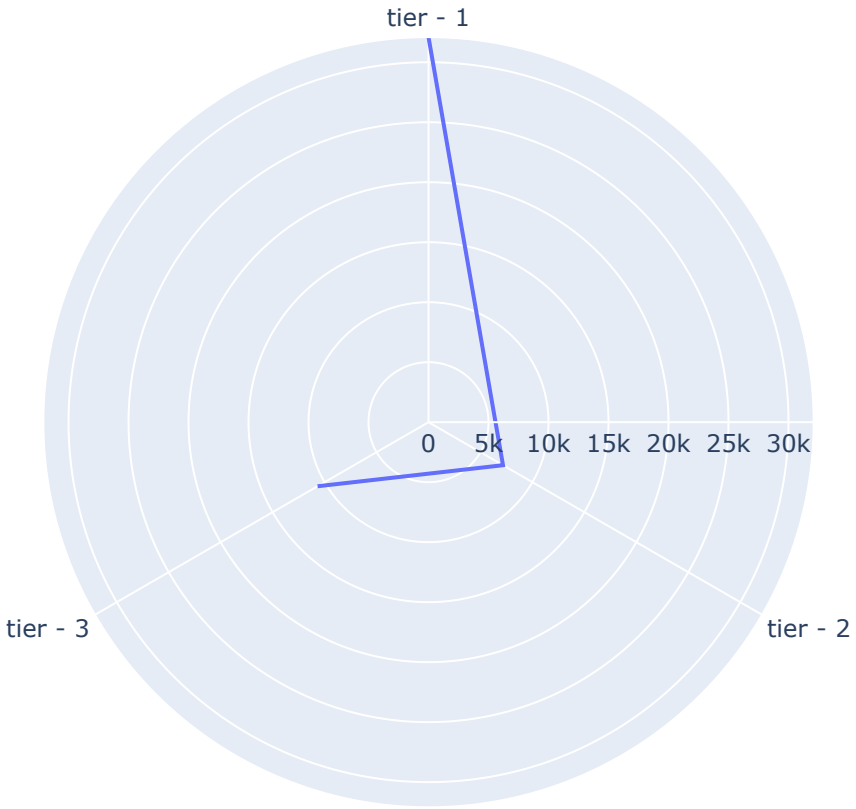
Radar Chart

In [42]:
```
med_hos = main_df.groupby('hospital_tier')['charges'].median().reset_index()
med_hos
```

Out[42]:

	hospital_tier	charges
0	tier - 1	32097.435
1	tier - 2	7168.760
2	tier - 3	10676.830

In [43]:
```
rad_chart = px.line_polar(med_hos,r='charges',theta='hospital_tier')
rad_chart.show()
```



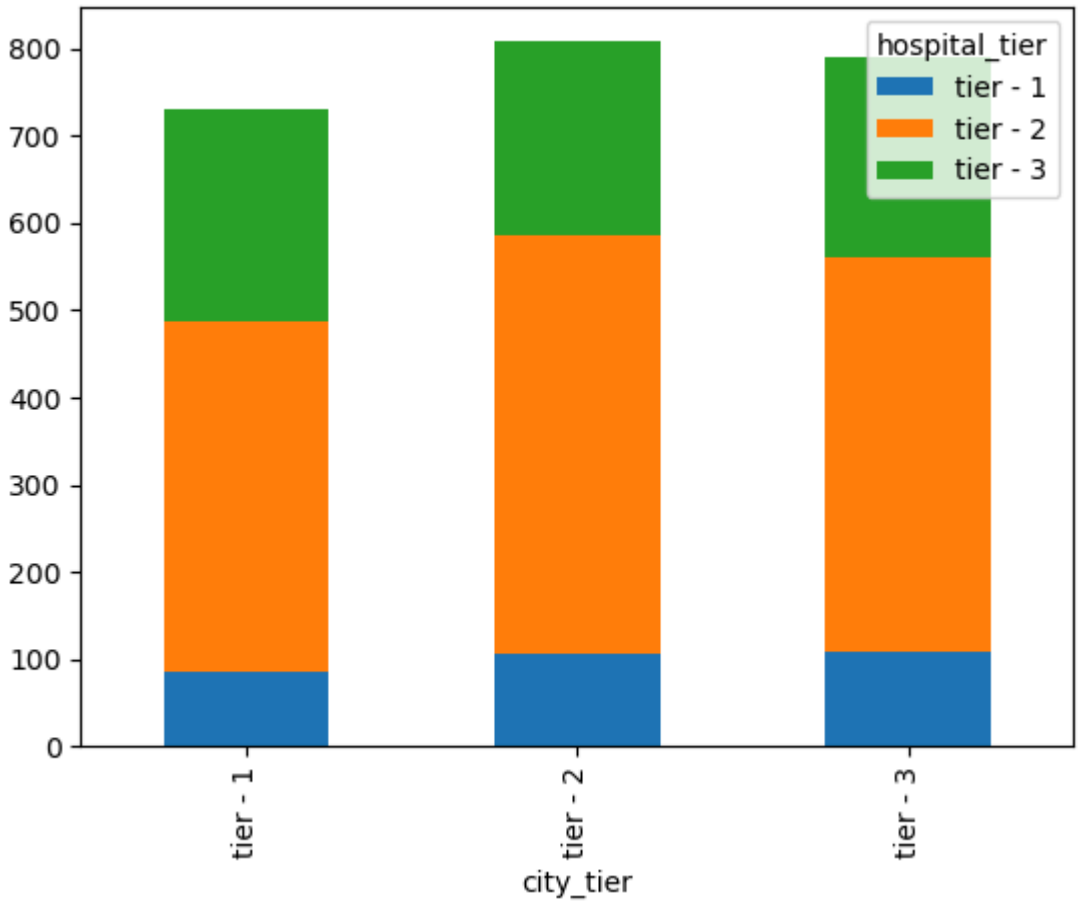
Frequency Table

In [44]:
```
fre_table = pd.crosstab(main_df.city_tier,main_df.hospital_tier)
fre_table
```

Out[44]:

hospital_tier	tier - 1	tier - 2	tier - 3
city_tier			
tier - 1	85	403	241
tier - 2	106	479	222
tier - 3	109	452	228

In [45]:
```
fre_table.plot.bar(stacked=True)
plt.show()
```



Hypothesis Testing

Hypothesis 1

- **H0 : Average hospitalization costs across the 3 types of hospitals are not significantly different.**
- **H1 : Average hospitalization costs across the 3 types of hospitals are significantly different.**

In [46]:
```
main_df.groupby('hospital_tier')['charges'].mean().round(2)
```

Out[46]:
```
hospital_tier
tier - 1    30132.00
tier - 2    11875.88
tier - 3     9487.46
Name: charges, dtype: float64
```

- **It is clearly visible that that the average hospitalisation costs across the hospital tiers are significantly different. So,we have to reject the null hypothesis.**

Hypothesis 2

- **H0 : Average hospitalization costs across the 3 types of cities are not significantly different**
- **H1 : Average hospitalization costs across the 3 types of cities are significantly different**

In [47]:
```
main_df.groupby('city_tier')['charges'].mean().round(2)
```

Out[47]:
```
city_tier
tier - 1    13009.97
tier - 2    13471.92
tier - 3    14045.31
Name: charges, dtype: float64
```

- **From the above data, we can conclude that the average hospitalisation costs across the city tiers are not significantly different. So,we have to accept the null hypothesis.**

Hypothesis 3

- **H0: Average hospitalization costs for smokers are not significantly different than non-smokers.**

- **H1: Average hospitalization costs for smokers are significantly different than non-smokers**

In [48]:
```
main_df.groupby('smoker')['charges'].mean().round(2)
```

Out[48]:
```
smoker
No      8409.20
yes    32866.96
Name: charges, dtype: float64
```

- **The above data depicts that the the average hospitalisation costs for smokers are significantly different than non-smokers. So,we have to reject the null hypothesis.**

Hypothesis 4

- **H0 : Smoking and Heart issues are independent**
- **H1 : Smoking and Heart issues are not independent**

In [49]:
```
smo_heart_table = pd.crosstab(main_df.smoker,main_df.heart_issues)
smo_heart_table
```

Out[49]:

heart_issues	No	yes
smoker		
No	1108	731
yes	297	189

In [50]:
```
chi, p, df, expected = stats.chi2_contingency(smo_heart_table)
chi, p, df, expected
```

Out[50]:
```
(0.08588150449910657,
 0.7694797581780767,
 1,
 array([[1111.30967742,  727.69032258],
        [ 293.69032258,  192.30967742]]))
```

- **The above table and data illustrates that smokers and heart issues are independent.So, we have to accept the null hypothesis.>**

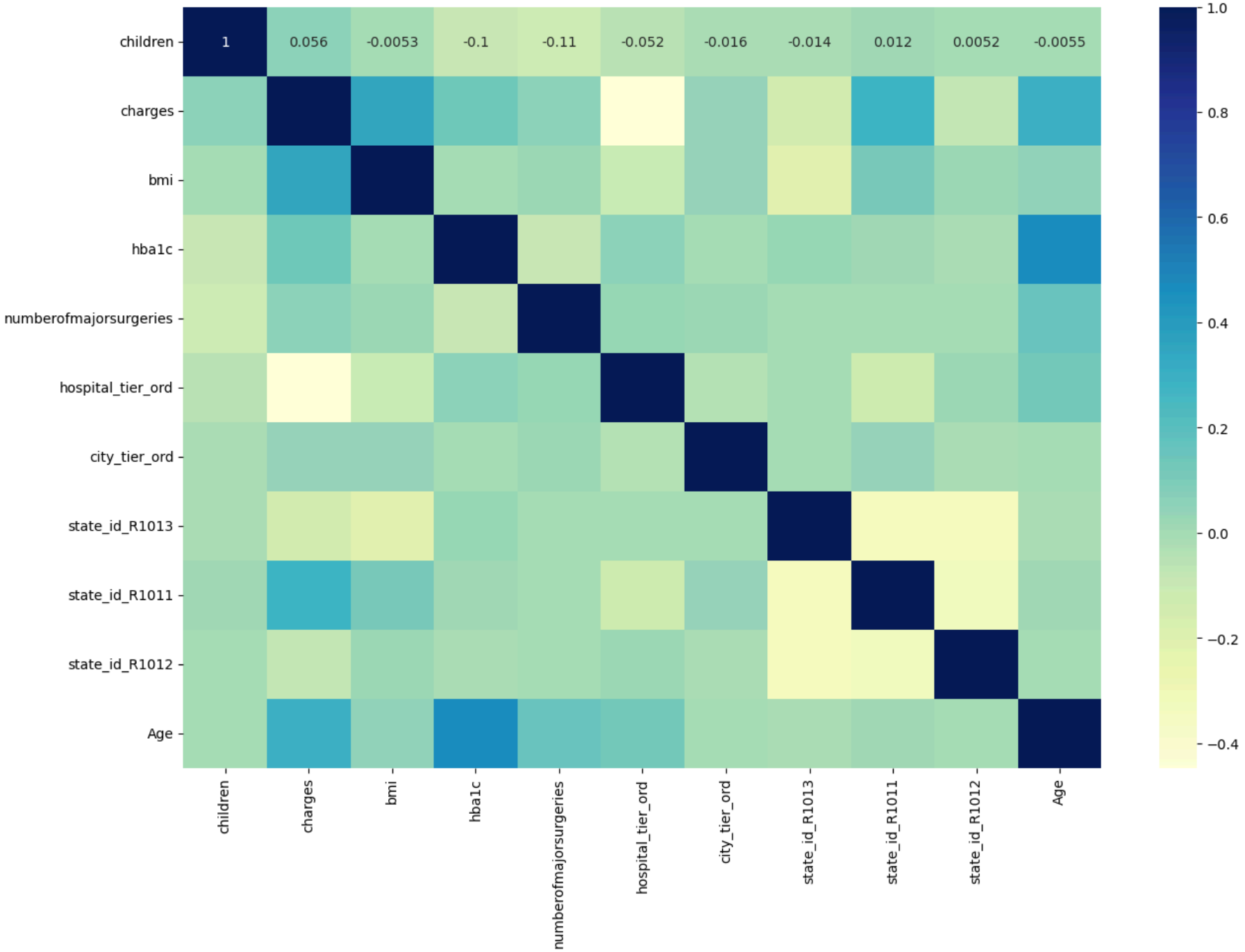
MACHINE LEARNING

Correlation

In [51]:
```
df2= main_df.drop(columns = ['customer_id','name', 'year', 'month', 'date','hospital_tier',
       'city_tier', 'state_id','title'])
```

In [52]:
```
df_correlation = df2.select_dtypes(exclude='object').corr()
```

In [53]:
```
plt.figure(figsize=(15,10))
sns.heatmap(df_correlation,annot=True,cmap='YlGnBu')
plt.show()
```


children
charges
bmi
hba1c
numberofmajorsurgeries
hospital_tier_ord
city_tier_ord
state_id_R1013
state_id_R1011
state_id_R1012
Age
1 0.056 -0.0053 -0.1 -0.11 -0.052 -0.016 -0.014 0.012 0.0052 -0.0055
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4

Final Model Development and Evaluation

In [54]:
```
df3 = pd.get_dummies(df2,drop_first=True)
df3
```

Out[54]:

	children	charges	bmi	hba1c	numberofmajorsurgeries	hospital_tier_ord	city_tier_ord	state_id_R1013	state_id_R1011	state_id_R1012	A
0	0	563.84	17.580	4.51	1	1.0	2.0	1	0	0	
1	0	570.62	17.600	4.39	1	1.0	0.0	1	0	0	
2	0	600.00	16.470	6.35	1	1.0	0.0	1	0	0	
3	0	604.54	17.700	6.28	1	2.0	2.0	1	0	0	
4	0	637.26	22.340	5.57	1	2.0	2.0	1	0	0	
...	...	...	...	...	...	...	...	...	...	...	
2329	0	52590.83	32.800	6.59	0	0.0	2.0	0	1	0	
2330	0	55135.40	35.530	5.45	0	0.0	1.0	0	0	1	
2331	1	58571.07	38.095	6.05	0	0.0	2.0	0	0	0	
2333	0	62592.87	30.360	5.77	0	1.0	2.0	1	0	0	
2334	0	63770.43	47.410	7.47	0	0.0	2.0	1	0	0	

2325 rows × 16 columns

In [55]:
```
df3.columns = df3.columns.str.lower()
```

In [56]:
```
y = df3['charges']
x = df3.drop(columns = 'charges')
```

In [57]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size = 0.20)
```

In [58]:
```
def get_score(model,x_train,x_test,y_train,y_test):
    model.fit(x_train,y_train)
    return model.score(x_test,y_test)
```

In [59]:
```
get_score(LinearRegression(),x_train,x_test,y_train,y_test)
```

Out[59]:
```
0.8316569085826744
```

In [60]:
```
def get_mse_score(model,x_train,x_test,y_train,y_test):
    model.fit(x_train,y_train)
    predictions = model.predict(x_test)
    score = mean_squared_error(y_test,predictions)
    return score
```

In [61]:
```
get_mse_score(LinearRegression(),x_train,x_test,y_train,y_test)
```

Out[61]:
```
22205901.593855884
```

Hyperparameter

In [62]:
```
df_hyperparameter = {'regressor__alpha': [0.001, 0.01, 0.1, 1, 10, 100]}
```

Pipeline

In [63]:
```
df_pipeline = Pipeline(steps=[('scaler', StandardScaler()), ('regressor', Ridge())])
```

Kfold

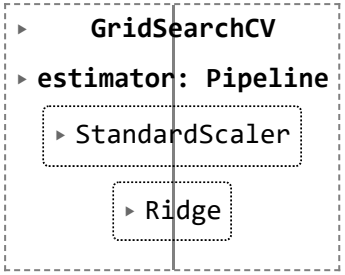
In [64]:
```
df_kfold = KFold(n_splits=5, shuffle=True, random_state=42)
```

GridSearchCV

In [65]:
```
df_grid = GridSearchCV(df_pipeline,df_hyperparameter,cv=df_kfold,scoring='neg_mean_squared_error')
```

In [66]:
```
df_grid.fit(x,y)
```

Out[66]:



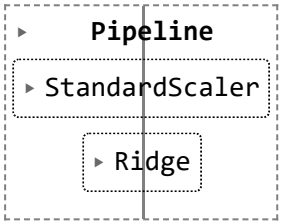
Best parameter and best model

In [67]:
```
df_grid.best_params_
```

Out[67]:
```
{'regressor__alpha': 10}
```

In [68]:
```
df_grid.best_estimator_
```

Out[68]:



Gradient Boosting Algorithm

In [69]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.30)
```

In [70]:
```
model = GradientBoostingRegressor()
model.fit(x_train,y_train)
```

Out[70]:
```
GradientBoostingRegressor

GradientBoostingRegressor()
```

In [71]:
```
print(model.feature_importances_)
```

```
[9.24542340e-03 1.09479880e-01 4.62723030e-03 3.69712779e-04
 1.88358000e-02 9.32756562e-05 6.17979106e-03 8.05201251e-03
 1.08710560e-03 9.28266126e-02 7.34153831e-05 1.67693403e-05
 0.00000000e+00 7.48903515e-01 2.09455847e-04]
```

In [72]:
```
imp_var = pd.DataFrame({'Features':model.feature_names_in_,'Importance':model.feature_importances_}).sort_values("Impo
imp_var
```

Out[72]:

	Features	Importance
13	smoker_yes	0.748904
1	bmi	0.109480
9	age	0.092827
4	hospital_tier_ord	0.018836
0	children	0.009245
7	state_id_r1011	0.008052
6	state_id_r1013	0.006180
2	hba1c	0.004627
8	state_id_r1012	0.001087
3	numberofmajorsurgeries	0.000370
14	gender_male	0.000209
5	city_tier_ord	0.000093
10	heart_issues_yes	0.000073
11	any_transplants_yes	0.000017
12	cancer_history_yes	0.000000

- **From the above table, all the variable having importance value less than 0.001 are reduntant variables.**
- gender_Male
- city_tier_ord
- heart_issues_yes
- numberofmajorsurgeries
- cancer_history_Yes
- state_id_R1012
- any_transplants_yes

In [73]:
```
model.score(x_train,y_train)
```

Out[73]: 0.9409643162176635

In [74]:
```
model.score(x_test,y_test)
```

Out[74]: 0.9074714764297274

Prediction

In [75]:
```
df_pred = pd.DataFrame({'Name' : ['Christopher, Ms. Jayna'],
                        'DOB' : ['12/28/1988'],
                        'city_tier' : ['tier - 1'], 'children' :[ 2],
                         'HbA1c' : [5.8],
                         'smoker_yes' : [1],
                         'heart_issues_yes' : [0],
                         'any_transplants_yes' : [0],
                         'numberofmajorsurgeries' :[ 0],
                         'cancer_history_yes' : [1],
                         'hospital_tier' : ['tier - 1'],
                         'bmi' : [85/(1.70 **2)],
                         'state_id_R1011' : [1]
                        })
df_pred
```

Out[75]:

	Name	DOB	city_tier	children	HbA1c	smoker_yes	heart_issues_yes	any_transplants_yes	numberofmajorsurgeries	cancer_history_
0	Christopher, Ms. Jayna	12/28/1988	tier - 1	2	5.8	1	0	0	0	

In [76]:
```
df_pred.columns = df_pred.columns.str.lower()
```

In [77]:
```
Title = df_pred.name.str.split('[,.]').str[1]
Title
```

Out[77]:
```
0     Ms
Name: name, dtype: object
```

In [78]:
```
df_pred['gender_male']  = 0
df_pred.loc[Title == 'Mr', 'gender_male'] = 1
```

In [79]:
```
df_pred.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   name                    1 non-null      object 
 1   dob                     1 non-null      object 
 2   city_tier               1 non-null      object 
 3   children                1 non-null      int64  
 4   hba1c                   1 non-null      float64
 5   smoker_yes              1 non-null      int64  
 6   heart_issues_yes        1 non-null      int64  
 7   any_transplants_yes     1 non-null      int64  
 8   numberofmajorsurgeries  1 non-null      int64  
 9   cancer_history_yes      1 non-null      int64  
 10  hospital_tier           1 non-null      object 
 11  bmi                     1 non-null      float64
 12  state_id_r1011          1 non-null      int64  
 13  gender_male             1 non-null      int64  
dtypes: float64(2), int64(8), object(4)
memory usage: 244.0+ bytes
```

In [80]:
```
df_pred['dob'] = pd.to_datetime(df_pred['dob'])
df_pred.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1 entries, 0 to 0
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   name                    1 non-null      object
 1   dob                     1 non-null      datetime64[ns]
 2   city_tier               1 non-null      object
 3   children                1 non-null      int64
 4   hba1c                   1 non-null      float64
 5   smoker_yes              1 non-null      int64
 6   heart_issues_yes        1 non-null      int64
 7   any_transplants_yes     1 non-null      int64
 8   numberofmajorsurgeries  1 non-null      int64
 9   cancer_history_yes      1 non-null      int64
 10  hospital_tier           1 non-null      object
 11  bmi                     1 non-null      float64
 12  state_id_r1011          1 non-null      int64
 13  gender_male             1 non-null      int64
dtypes: datetime64[ns](1), float64(2), int64(8), object(3)
memory usage: 244.0+ bytes
```

In [81]:
```
df_pred['age'] = 2024 - df_pred['dob'].dt.year
```

In [82]:
```
df_pred
```

Out[82]:

	name	dob	city_tier	children	hba1c	smoker_yes	heart_issues_yes	any_transplants_yes	numberofmajorsurgeries	cancer_history_yes
0	Christopher, Ms. Jayna	1988-12-28	tier - 1	2	5.8	1	0	0	0	1

In [83]:
```
df_pred= df_pred.drop(columns = ['name','dob'])
df_pred
```

Out[83]:

	city_tier	children	hba1c	smoker_yes	heart_issues_yes	any_transplants_yes	numberofmajorsurgeries	cancer_history_yes	hospital_tier	
0	tier - 1	2	5.8	1	0	0	0	1	tier - 1	29.411

In [84]:
```
ordinal = OrdinalEncoder(categories= [['tier - 1', 'tier - 2', 'tier - 3'],['tier - 1', 'tier - 2', 'tier - 3']])
df_pred[['hospital_tier_ord','city_tier_ord']] = ordinal.fit_transform(df_pred[['hospital_tier','city_tier']])
```

In [85]:
```
df_pred.drop(columns =['city_tier', 'hospital_tier'], inplace = True )
df_pred
```

Out[85]:

	children	hba1c	smoker_yes	heart_issues_yes	any_transplants_yes	numberofmajorsurgeries	cancer_history_yes	bmi	state_id_r1011	ge
0	2	5.8	1	0	0	0	1	29.411765	1	

In [86]:
```
df_pred.shape
```

Out[86]:
```
(1, 13)
```

In [87]:
```
for col in df3.columns:
    if col not in df_pred.columns and col != 'charges':
        df_pred[col] = 0
```

In [88]:
```
df_pred
```

Out[88]:

	children	hba1c	smoker_yes	heart_issues_yes	any_transplants_yes	numberofmajorsurgeries	cancer_history_yes	bmi	state_id_r1011	ge
0	2	5.8	1	0	0	0	1	29.411765	1	

In [89]:
```
df_pred.shape
```

Out[89]:
```
(1, 15)
```

In [90]:
```
df_pred=df_pred[df3.drop(columns='charges').columns]
```

In [91]:
```
model.predict(df_pred)
```

Out[91]:
```
array([29477.66193613])
```