PROBLEM STATEMENT:-

In this dataset which gender smokes highly

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

In [2]:
```
df=pd.read_csv(r"C:\Users\HP\OneDrive\Documents\insurance.csv")
df
```

Out[2]:

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

1338 rows × 7 columns

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

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

In [4]:
```
df.head()
```

Out[4]:

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

In [5]:
```
df.tail()
```

Out[5]:

	age	sex	bmi	children	smoker	region	charges
1333	50	male	30.97	3	no	northwest	10600.5483
1334	18	female	31.92	0	no	northeast	2205.9808
1335	18	female	36.85	0	no	southeast	1629.8335
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

In [6]:
```
1 df.describe()
```

Out[6]:

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

In [7]:
```
1 df.isnull().sum()
```

Out[7]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [8]:
```
1 df['smoker'].value_counts()
```

Out[8]:
```
smoker
no     1064
yes     274
Name: count, dtype: int64
```

In [9]:
```
1 df['sex'].value_counts()
```

Out[9]:
```
sex
male      676
female    662
Name: count, dtype: int64
```

In [10]:
```
1 df['region'].value_counts()
```

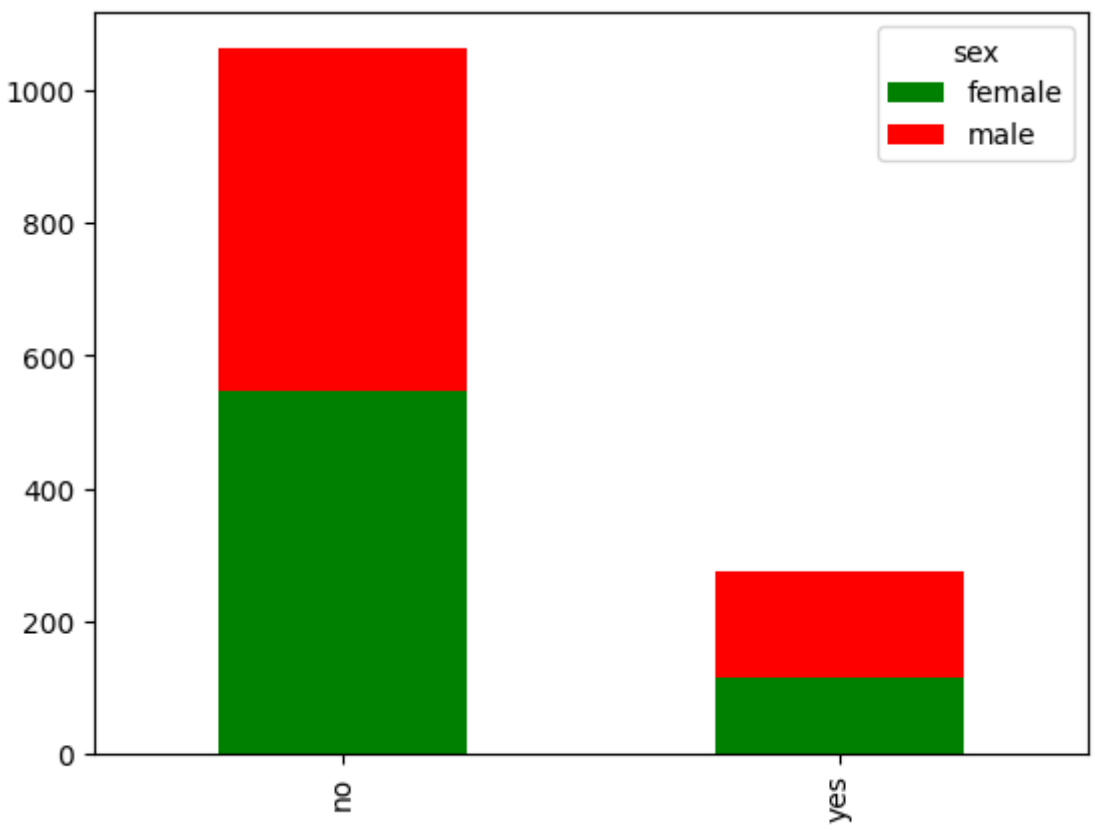
Out[10]:
```
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

In [11]:
```
1 s=pd.crosstab(df['smoker'],df['sex'])
2 print(s)
```

```
sex     female  male
smoker
no         547   517
yes        115   159
```

In [12]:
```
1 s.plot(kind='bar', stacked=True, color=['green','red'],grid=False)
```

Out[12]: <Axes: xlabel='smoker'>

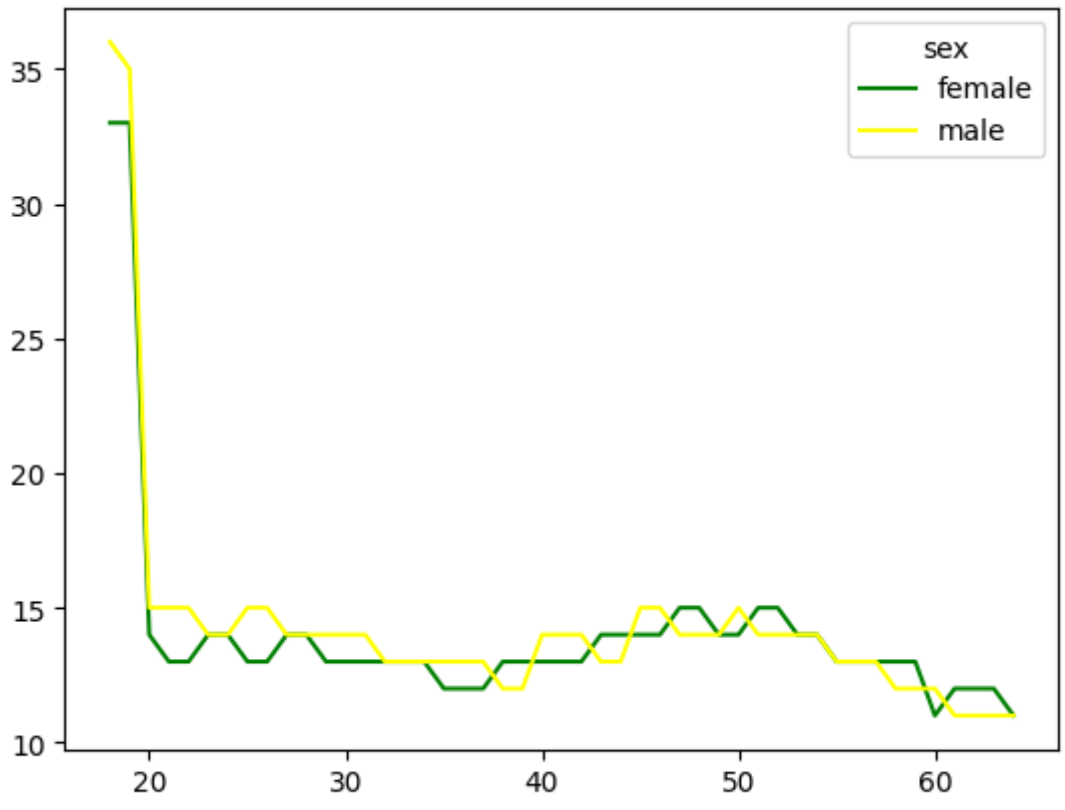


In [13]:
```
1 c=pd.crosstab(df['age'],df['sex'])
2 print(c)
```

sex	female	male
age		
18	33	36
19	33	35
20	14	15
21	13	15
22	13	15
23	14	14
24	14	14
25	13	15
26	13	15
27	14	14
28	14	14
29	13	14
30	13	14
31	13	14
32	13	13
33	13	13
34	13	13
35	12	13
36	12	13
37	12	13
38	13	12
39	13	12
40	13	14
41	13	14
42	13	14
43	14	13
44	14	13
45	14	15
46	14	15
47	15	14
48	15	14
49	14	14
50	14	15
51	15	14
52	15	14
53	14	14
54	14	14
55	13	13
56	13	13
57	13	13
58	13	12
59	13	12
60	11	12
61	12	11
62	12	11
63	12	11
64	11	11

In [14]:
```
1 c.plot(kind='line', stacked=False, color=['green','yellow'],grid=False)
```

Out[14]: <Axes: xlabel='age'>



In [15]:
```
1 s = {'region':{'northeast':1,'northwest':2,'southwest':3,'southeast':4}}
2 df = df.replace(s)
3 print(df)
```

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

[1338 rows x 7 columns]
```

In [16]:
```
1 S = {'sex':{'female':1,'male':2}}
2 df =df.replace(S)
3 print(df)
```

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

[1338 rows x 7 columns]
```

In [17]:
```
1 x = df.drop('smoker',axis=1)
2 y = df['smoker']
```

In [18]:
```
1 x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=40)
2 x_train.shape,x_test.shape
```

Out[18]: ((936, 6), (402, 6))

LogisticRegression

In [19]:
```
from sklearn.linear_model import LogisticRegression
lr=LogisticRegression()
lr.fit(x_train,y_train)
print(lr.score(x_test,y_test))
```

```
0.9378109452736318
```

DecisionTree

In [20]:
```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=0)
clf.fit(x_train,y_train)
score = clf.score(x_test,y_test)
print(score)
```

```
0.9601990049751243
```

RandomForestClassifier

In [21]:
```
from sklearn.ensemble import RandomForestClassifier
rfc=RandomForestClassifier()
rfc.fit(x_train,y_train)
print(rfc.score(x_test,y_test))
```

```
0.9651741293532339
```

In [22]:
```
params={'max_depth':[2,5,10,20,25],'min_samples_leaf':[5,20,30,50,100,200],'n_estimators':[10,40,50,60,100,20
```

In [23]:
```
from sklearn.model_selection import GridSearchCV
grid_search = GridSearchCV(estimator=rfc,param_grid=params,cv=2,scoring='accuracy')
grid_search.fit(x_train,y_train)
```

Out[23]:
```
GridSearchCV(cv=2, estimator=RandomForestClassifier(),
             param_grid={'max_depth': [2, 5, 10, 20, 25],
                         'min_samples_leaf': [5, 20, 30, 50, 100, 200],
                         'n_estimators': [10, 40, 50, 60, 100, 200]},
             scoring='accuracy')
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [24]:
```
grid_search.best_score_
```

Out[24]:
```
0.9487179487179487
```

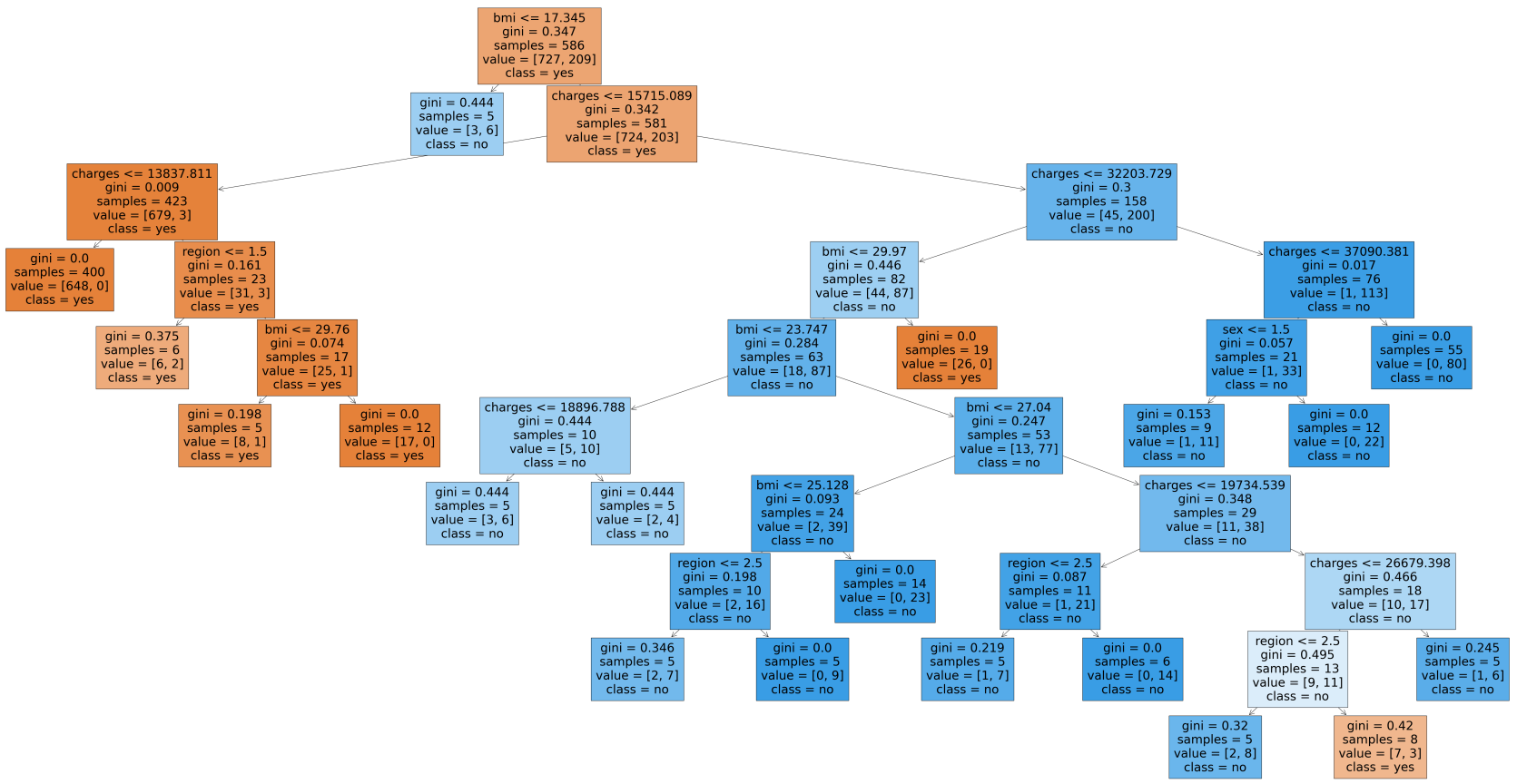
In [25]:
```
rfc_best = grid_search.best_estimator_
```

In [26]:
```
from sklearn.tree import plot_tree
plt.figure(figsize = (90,40))
plot_tree(rfc_best.estimators_[5],feature_names=x.columns,class_names=['yes','no'],filled=True)
```

Out[26]:
```
[Text(0.8117138364779874, 0.9583333333333334, 'bmi <= 44.825\ngini = 0.358\nsamples = 604\nvalue = [717, 219]\nclass = yes'),
 Text(0.6863207547169812, 0.875, 'region <= 3.5\ngini = 0.349\nsamples = 590\nvalue = [707, 206]\nclass = yes'),
 Text(0.5110062893081762, 0.7916666666666666, 'children <= 3.5\ngini = 0.319\nsamples = 437\nvalue = [552, 137]\nclass = yes'),
 Text(0.4858490566037736, 0.7083333333333334, 'age <= 59.5\ngini = 0.327\nsamples = 421\nvalue = [528, 137]\nclass = yes'),
 Text(0.3490566037735849, 0.625, 'children <= 0.5\ngini = 0.317\nsamples = 394\nvalue = [503, 124]\nclass = yes'),
 Text(0.20125786163522014, 0.5416666666666666, 'age <= 46.5\ngini = 0.268\nsamples = 173\nvalue = [227, 43]\nclass = yes'),
 Text(0.10062893081761007, 0.4583333333333333, 'age <= 28.5\ngini = 0.324\nsamples = 119\nvalue = [149, 38]\nclass = yes'),
 Text(0.050314465408805034, 0.375, 'charges <= 13586.758\ngini = 0.255\nsamples = 77\nvalue = [102, 18]\nclass = yes'),
 Text(0.025157232704402517, 0.2916666666666667, 'gini = 0.0\nsamples = 64\nvalue = [100, 0]\nclass = yes'),
 Text(0.07547169811320754, 0.2916666666666667, 'sex <= 1.5\ngini = 0.18\nsamples = 13\nvalue = [2, 18]\nclass = no'),
```

In [27]:
```
plt.figure(figsize=(80,40))
plot_tree(rfc_best.estimators_[7],feature_names=x.columns,class_names=['yes','no'],filled=True)
```

Out[27]:
```
[Text(0.35855263157894735, 0.95, 'bmi <= 17.345\ngini = 0.347\nsamples = 586\nvalue = [727, 209]\nclass = yes'),
 Text(0.3059210526315789, 0.85, 'gini = 0.444\nsamples = 5\nvalue = [3, 6]\nclass = no'),
 Text(0.41118421052631576, 0.85, 'charges <= 15715.089\ngini = 0.342\nsamples = 581\nvalue = [724, 203]\nclass = yes'),
 Text(0.10526315789473684, 0.75, 'charges <= 13837.811\ngini = 0.009\nsamples = 423\nvalue = [679, 3]\nclass = yes'),
 Text(0.05263157894736842, 0.65, 'gini = 0.0\nsamples = 400\nvalue = [648, 0]\nclass = yes'),
 Text(0.15789473684210525, 0.65, 'region <= 1.5\ngini = 0.161\nsamples = 23\nvalue = [31, 3]\nclass = yes'),
 Text(0.10526315789473684, 0.55, 'gini = 0.375\nsamples = 6\nvalue = [6, 2]\nclass = yes'),
 Text(0.21052631578947367, 0.55, 'bmi <= 29.76\ngini = 0.074\nsamples = 17\nvalue = [25, 1]\nclass = yes'),
 Text(0.15789473684210525, 0.45, 'gini = 0.198\nsamples = 5\nvalue = [8, 1]\nclass = yes'),
 Text(0.2631578947368421, 0.45, 'gini = 0.0\nsamples = 12\nvalue = [17, 0]\nclass = yes'),
 Text(0.7171052631578947, 0.75, 'charges <= 32203.729\ngini = 0.3\nsamples = 158\nvalue = [45, 200]\nclass = no'),
 Text(0.5657894736842105, 0.65, 'bmi <= 29.97\ngini = 0.446\nsamples = 82\nvalue = [44, 87]\nclass = no'),
 Text(0.5131578947368421, 0.55, 'bmi <= 23.747\ngini = 0.284\nsamples = 63\nvalue = [18, 87]\nclass = no'),
 Text(0.3684210526315789, 0.45, 'charges <= 18896.788\ngini = 0.444\nsamples = 10\nvalue = [5, 10]\nclass = no'),
 Text(0.3157894736842105, 0.35, 'gini = 0.444\nsamples = 5\nvalue = [3, 6]\nclass = no'),
 Text(0.42105263157894735, 0.35, 'gini = 0.444\nsamples = 5\nvalue = [2, 4]\nclass = no'),
 Text(0.6578947368421053, 0.45, 'bmi <= 27.04\ngini = 0.247\nsamples = 53\nvalue = [13, 77]\nclass = no'),
 Text(0.5263157894736842, 0.35, 'bmi <= 25.128\ngini = 0.093\nsamples = 24\nvalue = [2, 39]\nclass = no'),
 Text(0.47368421052631576, 0.25, 'region <= 2.5\ngini = 0.198\nsamples = 10\nvalue = [2, 16]\nclass = no'),
 Text(0.42105263157894735, 0.15, 'gini = 0.346\nsamples = 5\nvalue = [2, 7]\nclass = no'),
 Text(0.5263157894736842, 0.15, 'gini = 0.0\nsamples = 5\nvalue = [0, 9]\nclass = no'),
 Text(0.5789473684210527, 0.25, 'gini = 0.0\nsamples = 14\nvalue = [0, 23]\nclass = no'),
 Text(0.7894736842105263, 0.35, 'charges <= 19734.539\ngini = 0.348\nsamples = 29\nvalue = [11, 38]\nclass = no'),
 Text(0.6842105263157895, 0.25, 'region <= 2.5\ngini = 0.087\nsamples = 11\nvalue = [1, 21]\nclass = no'),
 Text(0.631578947368421, 0.15, 'gini = 0.219\nsamples = 5\nvalue = [1, 7]\nclass = no'),
 Text(0.7368421052631579, 0.15, 'gini = 0.0\nsamples = 6\nvalue = [0, 14]\nclass = no'),
 Text(0.8947368421052632, 0.25, 'charges <= 26679.398\ngini = 0.466\nsamples = 18\nvalue = [10, 17]\nclass = no'),
 Text(0.8421052631578947, 0.15, 'region <= 2.5\ngini = 0.495\nsamples = 13\nvalue = [9, 11]\nclass = no'),
 Text(0.7894736842105263, 0.05, 'gini = 0.32\nsamples = 5\nvalue = [2, 8]\nclass = no'),
 Text(0.8947368421052632, 0.05, 'gini = 0.42\nsamples = 8\nvalue = [7, 3]\nclass = yes'),
 Text(0.9473684210526315, 0.15, 'gini = 0.245\nsamples = 5\nvalue = [1, 6]\nclass = no'),
 Text(0.618421052631579, 0.55, 'gini = 0.0\nsamples = 19\nvalue = [26, 0]\nclass = yes'),
 Text(0.868421052631579, 0.65, 'charges <= 37090.381\ngini = 0.017\nsamples = 76\nvalue = [1, 113]\nclass = no'),
 Text(0.8157894736842105, 0.55, 'sex <= 1.5\ngini = 0.057\nsamples = 21\nvalue = [1, 33]\nclass = no'),
 Text(0.7631578947368421, 0.45, 'gini = 0.153\nsamples = 9\nvalue = [1, 11]\nclass = no'),
 Text(0.868421052631579, 0.45, 'gini = 0.0\nsamples = 12\nvalue = [0, 22]\nclass = no'),
 Text(0.9210526315789473, 0.55, 'gini = 0.0\nsamples = 55\nvalue = [0, 80]\nclass = no')]
```



In [28]:
```
rfc_best.feature_importances_
```

Out[28]:
```
array([0.04486405, 0.00595937, 0.06919516, 0.01160914, 0.01123564,
       0.85713664])
```

In [29]:
```
imp_df = pd.DataFrame({"Varname":x_train.columns,'Imp':rfc_best.feature_importances_})
imp_df.sort_values(by='Imp',ascending=False)
```

Out[29]:

	Varname	Imp
5	charges	0.857137
2	bmi	0.069195
0	age	0.044864
3	children	0.011609
4	region	0.011236
1	sex	0.005959

In [30]:
```
df['bmi'].value_counts()
```

Out[30]:
```
bmi
32.300    13
28.310     9
30.495     8
30.875     8
31.350     8
          ..
46.200     1
23.800     1
44.770     1
32.120     1
30.970     1
Name: count, Length: 548, dtype: int64
```

CONCLUSION:-

Based dataset We conclude that male smoker are high compared to female smokers

In [31]:
```
import pickle
```

In [32]:
```
filename="Insurance prediction"
pickle.dump(lr,open(filename,'wb'))
```