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

In [2]:
```
Medical_Insurance=pd.read_csv("C:\\Users\\Pranav\\Desktop\\DATA SCIENCE DATA\\CVC file\\insurance.csv")
Medical_Insurance.head()
```

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

In [3]:
```
#number of rows and columns
Medical_Insurance.shape
```

Out[3]:
```
(1338, 7)
```

In [4]:
```
#describe mathamatical data
Medical_Insurance.describe()
```

Out[4]:

	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 [5]:
```
#information about data
Medical_Insurance.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 [6]:
```
#to find missing value in data
Medical_Insurance.isnull().sum()
```

Out[6]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [7]:
```
Medical_Insurance['region'].value_counts()
```

Out[7]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [8]:
```
Medical_Insurance['sex'].value_counts()
```

Out[8]:
```
male      676
female    662
Name: sex, dtype: int64
```

In [9]:
```
Medical_Insurance['children'].value_counts()
```

Out[9]:
```
0    574
1    324
2    240
3    157
4     25
5     18
Name: children, dtype: int64
```

In [10]:
```
Medical_Insurance['smoker'].value_counts()
```

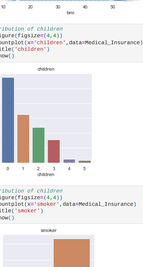
Out[10]:
```
no     1064
yes     274
Name: smoker, dtype: int64
```

In [11]:
```
Medical_Insurance["region"].value_counts()
```

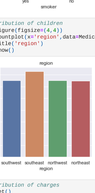
Out[11]:
```
southeast    364
southwest    325
northwest    325
northeast    324
Name: region, dtype: int64
```

In [12]:
```
#distribution of age value
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(Medical_Insurance['age'])
plt.title('Age Distribution')
plt.show()
```

```
C:\Users\Pranav\Searches\hjhkh\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

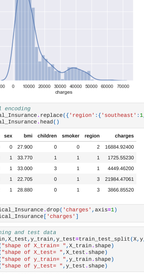


In [13]:
```
#distribution of sex
plt.figure(figsize=(4,4))
sns.countplot(x='sex',data=Medical_Insurance)
plt.title('Sex Distribution')
plt.show()
```



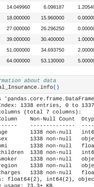
In [14]:
```
#distribution of bmi value
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(Medical_Insurance['bmi'])
plt.title('bmi Distribution')
plt.show()
```

```
C:\Users\Pranav\Searches\hjhkh\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

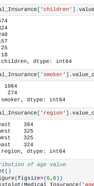


In [15]:
```
#distribution of children
plt.figure(figsize=(4,4))
sns.countplot(x='children',data=Medical_Insurance)
plt.title('children')
plt.show()
```

In [16]:
```
#distribution of children
plt.figure(figsize=(4,4))
sns.countplot(x='smoker',data=Medical_Insurance)
plt.title('smoker')
plt.show()
```



In [17]:
```
#distribution of children
plt.figure(figsize=(4,4))
sns.countplot(x='region',data=Medical_Insurance)
plt.title('region')
plt.show()
```



In [18]:
```
#distribution of charges
sns.set()
plt.figure(figsize=(6,6))
sns.distplot(Medical_Insurance['charges'])
plt.title('charges')
plt.show()
```

```
C:\Users\Pranav\Searches\hjhkh\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `histplot` (an axes-level function for histograms).
  warnings.warn(msg, FutureWarning)
```

In [19]:
```
#label encoding
Medical_Insurance.replace({'region':{'southeast':1,'southwest':2,'northwest':3,'northeast':4},'sex':{'female':0,'male':1},'smoker':{'yes':0,'no':1}},inplace=True)
Medical_Insurance.head()
```

Out[19]:

	age	sex	bmi	children	smoker	region	charges
0	19	0	27.900	0	0	2	16884.92400
1	18	1	33.770	1	1	1	1725.55230
2	28	1	33.000	3	1	1	4449.46200
3	33	1	22.705	0	1	3	21984.47061
4	32	1	28.880	0	1	3	3866.85520

In [20]:
```
X=Medical_Insurance.drop('charges',axis=1)
y=Medical_Insurance['charges']
```

In [21]:
```
#training and test data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=2)
print("shape of X_train=",X_train.shape)
print("shape of X_test= ",X_test.shape)
print("shape of y_train= ",y_train.shape)
print("shape of y_test= ",y_test.shape)
```

```
shape of X_train=  (1070, 6)
shape of X_test=  (268, 6)
shape of y_train=  (1070,)
shape of y_test=  (268,)
```

In [28]:
```
model=LinearRegression()
model.fit(X_train,y_train)
```

Out[28]:
```
LinearRegression()
```

In [30]:
```
traininga_data_prediction=model.predict(X_train)
print(traininga_data_prediction)
```

```
[  781.23289463   9150.38548207 13163.38600896 ... 17329.28768831
  9545.84287714 14088.60244423]
```

In [33]:
```
r2_train=metrics.r2_score(y_train,traininga_data_prediction)
print('R squared value =',r2_train)
```

```
R squared value = 0.7518195459072954
```

In [34]:
```
#Making a Predictive system
input_data1=(19,0,27.900,0,0,2)
input_data_numpy_array1=np.asarray(input_data1)
```

In [36]:
```
#reshape the np array as we are predicting for one instance
input_data_reshaped1=input_data_numpy_array1.reshape(1,-1)
input_data_reshaped1
```

Out[36]:
```
array([[19. ,  0. , 27.9,  0. ,  0. ,  2. ]])
```

In [38]:
```
prediction=model.predict(input_data_reshaped1)
print(prediction)
print('The Insurance Cost is USD ',prediction[0])
```

```
[25558.92638242]
The Insurance Cost is USD  25558.92638242239
```

```
C:\Users\Pranav\Searches\hjhkh\lib\site-packages\sklearn\base.py:450: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```