In [72]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [73]:
```
#df= pd.read_csv('helth insurance.csv')
df=pd.read_csv('insurance.csv')
df
```

Out[73]:

	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 [74]:
```
df.head()
```

Out[74]:

	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 [75]:
```
df.dtypes
```

Out[75]:
```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

In [76]:
```
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 [77]:
```
df.describe
```

Out[77]:
```
<bound method NDFrame.describe of       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 x 7 columns]>
```

In [78]:
```
df.describe()
```

Out[78]:

	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 [79]:
```
df.count
```

Out[79]:
```
<bound method DataFrame.count of       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 x 7 columns]>
```

In [80]:
```
df.shape
```

Out[80]:
```
(1338, 7)
```

In [81]:
```
df.isnull().sum()
```

Out[81]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [82]:
```
df.isnull().sum().sum()
```

Out[82]:
```
np.int64(0)
```

In [83]:
```
df.duplicated().sum()
```

Out[83]:
```
np.int64(1)
```

In [84]:
```
df.columns
```

Out[84]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtyp
e='object')
```

total catg_col finds in the dataset

In [85]:
```
catg_colmns=df.select_dtypes(include= 'object').columns.tolist()
catg_colmns
```

Out[85]:
```
['sex', 'smoker', 'region']
```

Apply label encoding technique

In [86]:
```
from sklearn.preprocessing import LabelEncoder
```

In [87]:
```
label_encoder= LabelEncoder()

#apply label encoding in your target column of the dataset(ex- 'smoker')
df['smoker']= label_encoder.fit_transform(df['smoker'])

df.head()
```

Out[87]:

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

one hot encoding

In [88]:
```
nominal_categorical_vars= ['sex', 'smoker', 'region']
for i in nominal_categorical_vars:
    print(i, ":", df[i].nunique())
    print(df[i].value_counts())
    print(" ")
```

```
sex : 2
sex
male      676
female    662
Name: count, dtype: int64

smoker : 2
smoker
0    1064
1     274
Name: count, dtype: int64

region : 4
region
southeast    364
southwest    325
northwest    325
northeast    324
Name: count, dtype: int64
```

method 1

In [89]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [90]:
```
df1= df.copy()

#create a onehotencoder instance
Onehot_encoder= OneHotEncoder(sparse_output=False, drop='first')
#columns to one_hot_encode
columns_to_encode= ['sex', 'smoker', 'region']

#apply one-hot encoding to the selected columns
encoded_columns= Onehot_encoder.fit_transform(df1[columns_to_encode])

#create a df for one_hot_encoded columns
encoded_cols_df1= pd.DataFrame(encoded_columns,columns=Onehot_encoder.get_feat

#concatenate the df1 with the original dataframe
df1= pd.concat([df1, encoded_cols_df1], axis=1)

#drop the original df that we encoded
df1.drop(columns_to_encode, axis=1)

df1.head()
```

Out[90]:

	age	sex	bmi	children	smoker	region	charges	sex_male	smo
0	19	female	27.900	0	1	southwest	16884.92400	0.0	
1	18	male	33.770	1	0	southeast	1725.55230	1.0	
2	28	male	33.000	3	0	southeast	4449.46200	1.0	
3	33	male	22.705	0	0	northwest	21984.47061	1.0	
4	32	male	28.880	0	0	northwest	3866.85520	1.0	

In [91]:
```
df.shape[1]    #before encoding
```

Out[91]: 7

In [92]:
```
df1.shape[1]   #after encoding
```

Out[92]: 12

method 2

In [93]:
```
df3= pd.get_dummies(df, columns=['sex', 'smoker', 'region'], drop_first=True)
df3.head()
```

Out[93]:

	age	bmi	children	charges	sex_male	smoker_1	region_northwest	re
0	19	27.900	0	16884.92400	False	True	False	
1	18	33.770	1	1725.55230	True	False	False	
2	28	33.000	3	4449.46200	True	False	False	
3	33	22.705	0	21984.47061	True	False	True	
4	32	28.880	0	3866.85520	True	False	True	

ordinal encoding

In [63]:
```
from sklearn.preprocessing import OrdinalEncoder
```

In [98]:
```
#define the order of categories for which column do you use
region_order= ['southwest', 'southeast', 'northwest', 'northeast']

#create an ordinalencoder instance and specify the order
ordinal_encoder= OrdinalEncoder(categories=[region_order])

df['region_encoded']= ordinal_encoder.fit_transform(df[['region']])

#display the update df
```

```
df[['region', 'region_encoded']].head(8)
```

Out[98]:

	region	region_encoded
0	southwest	0.0
1	southeast	1.0
2	southeast	1.0
3	northwest	2.0
4	northwest	2.0
5	southeast	1.0
6	southeast	1.0
7	northwest	2.0

custom encoding

In [97]:

```
#define the custom encoding dictionary:
custom_encoding= {'male':10,'female': 20}

#apply the custom encoding to the column that you have to use
df['sex_encoded']= df['sex'].replace(custom_encoding)

#display the update dataframe
df[['sex', 'sex_encoded']].sample(10)
```

```
C:\Users\shaw3\AppData\Local\Temp\ipykernel_5616\3489771706.py:5: FutureWarnin
g: Downcasting behavior in `replace` is deprecated and will be removed in a fut
ure version. To retain the old behavior, explicitly call `result.infer_object
s(copy=False)`. To opt-in to the future behavior, set `pd.set_option('future.n
o_silent_downcasting', True)`
  df['sex_encoded']= df['sex'].replace(custom_encoding)
```

Out[97]:

	sex	sex_encoded
465	female	20
896	female	20
835	male	10
195	male	10
386	female	20
815	female	20
863	female	20
1313	female	20
850	female	20
631	male	10

In [ ]:
```
#df.drop('sex', axis=1, inplace=True) #drop the original column
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [2]:
```
df= pd.read_csv('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.head()
```

Out[3]:

	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 [4]:
```
df.describe
```

Out[4]:
```
<bound method NDFrame.describe of       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 x 7 columns]>
```

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

Out[5]:

	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 [6]:
```
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 [7]:
```
df.columns
```

Out[7]:
```
Index(['age', 'sex', 'bmi', 'children', 'smoker', 'region', 'charges'], dtype='object')
```

In [8]:
```
df.dtypes
```

Out[8]:
```
age           int64
sex          object
bmi         float64
children      int64
smoker       object
region       object
charges     float64
dtype: object
```

In [9]:
```
df.shape
```

Out[9]:
```
(1338, 7)
```

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

Out[10]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

In [13]:
```
df.count
```

Out[13]:
```
<bound method DataFrame.count of       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 x 7 columns]>
```

creating a list that holds the features which are categorical in this dataset

In [15]:
```
catg_cols= df.select_dtypes(include='object').columns.tolist()
catg_cols
```

Out[15]:
```
['sex', 'smoker', 'region']
```

one hot encoding

In [19]:
```
from sklearn.preprocessing import OneHotEncoder
encoder= OneHotEncoder(sparse_output= False, handle_unknown= 'ignore')
encoder.fit(df[catg_cols])
```

Out[19]:

OneHotEncoder

```
OneHotEncoder(handle_unknown='ignore', sparse_output=False)
```

In [21]:
```
encoder.categories_
```

Out[21]:
```
[array(['female', 'male'], dtype=object),
 array(['no', 'yes'], dtype=object),
 array(['northeast', 'northwest', 'southeast', 'southwest'], dtype=object)]
```

In [23]:
```
encoder_cols=list(encoder.get_feature_names_out(catg_cols))
encoder_cols
```

Out[23]:
```
['sex_female',
 'sex_male',
 'smoker_no',
 'smoker_yes',
 'region_northeast',
 'region_northwest',
 'region_southeast',
 'region_southwest']
```

transforming the data (cat- num):

In [24]:
```
df[encoder_cols]= encoder.transform(df[catg_cols])
```

In [25]:
```
df.head()
```

Out[25]:

	age	sex	bmi	children	smoker	region	charges	sex_female	se
0	19	female	27.900	0	yes	southwest	16884.92400	1.0	
1	18	male	33.770	1	no	southeast	1725.55230	0.0	
2	28	male	33.000	3	no	southeast	4449.46200	0.0	
3	33	male	22.705	0	no	northwest	21984.47061	0.0	
4	32	male	28.880	0	no	northwest	3866.85520	0.0	

In [26]:
```
df.drop(columns= catg_cols, inplace=True)
```

In [27]:
```
df.head()
```

Out[27]:

	age	bmi	children	charges	sex_female	sex_male	smoker_no	smoke
0	19	27.900	0	16884.92400	1.0	0.0	0.0	
1	18	33.770	1	1725.55230	0.0	1.0	1.0	
2	28	33.000	3	4449.46200	0.0	1.0	1.0	
3	33	22.705	0	21984.47061	0.0	1.0	1.0	
4	32	28.880	0	3866.85520	0.0	1.0	1.0	

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: