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 [ ]: