smartinternz-assignment-2

July 25, 2024

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
```

[4]:
```
data = pd.read_csv("/content/insurance.csv")
```

[5]:
```
data.head()
```

[5]:

	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

[6]:
```
data.tail()
```

[6]:

	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

[7]:
```
data.shape
```

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

[8]:
```
data.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
```

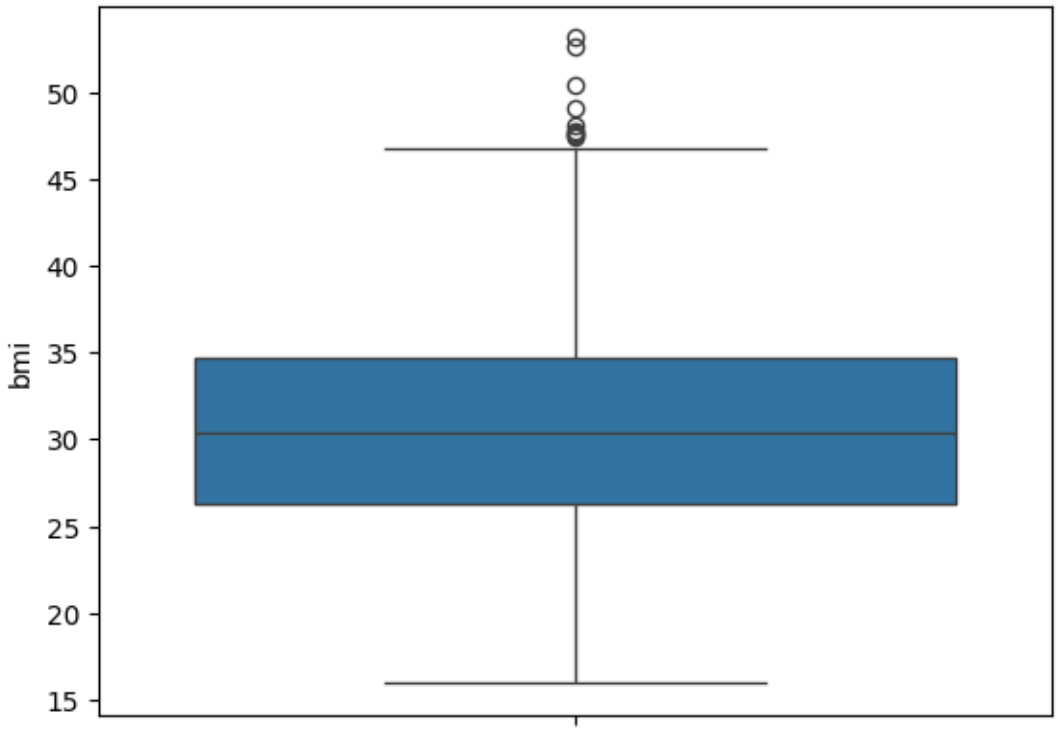
```
 3   children  1338 non-null   int64
 4   smoker    1338 non-null   object
 5   region    1338 non-null   object
 6   charges   1338 non-null   float64
dtypes: float64(2), int64(2), object(3)
memory usage: 73.3+ KB
```

[13]:
```
data.isnull().sum()
```

[13]:
```
age         0
sex         0
bmi         0
children    0
smoker      0
region      0
charges     0
dtype: int64
```

[16]:
```
sns.boxplot(data['bmi'])
```

[16]:
```
<Axes: ylabel='bmi'>
```



[17]:
```
IQR = data['bmi'].quantile(0.75)-data['bmi'].quantile(0.25)
IQR
```

[17]: 8.3975

[18]:
```
lowerBound=data['bmi'].quantile(0.25)-(1.5*IQR)
lowerBound
```

[18]: 13.7

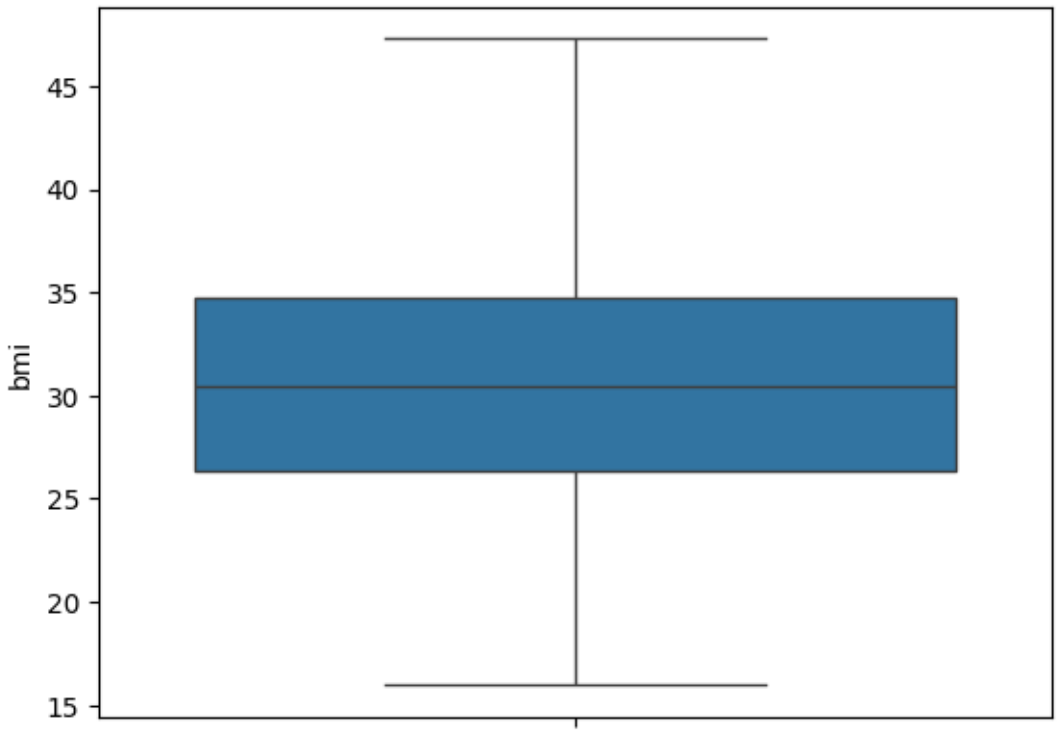
[19]:
```
upperBound=data['bmi'].quantile(0.75)+(1.5*IQR)
upperBound
```

[19]: 47.290000000000006

[20]:
```
data['bmi']=np.where(data['bmi']>upperBound,upperBound,data['bmi'])
```

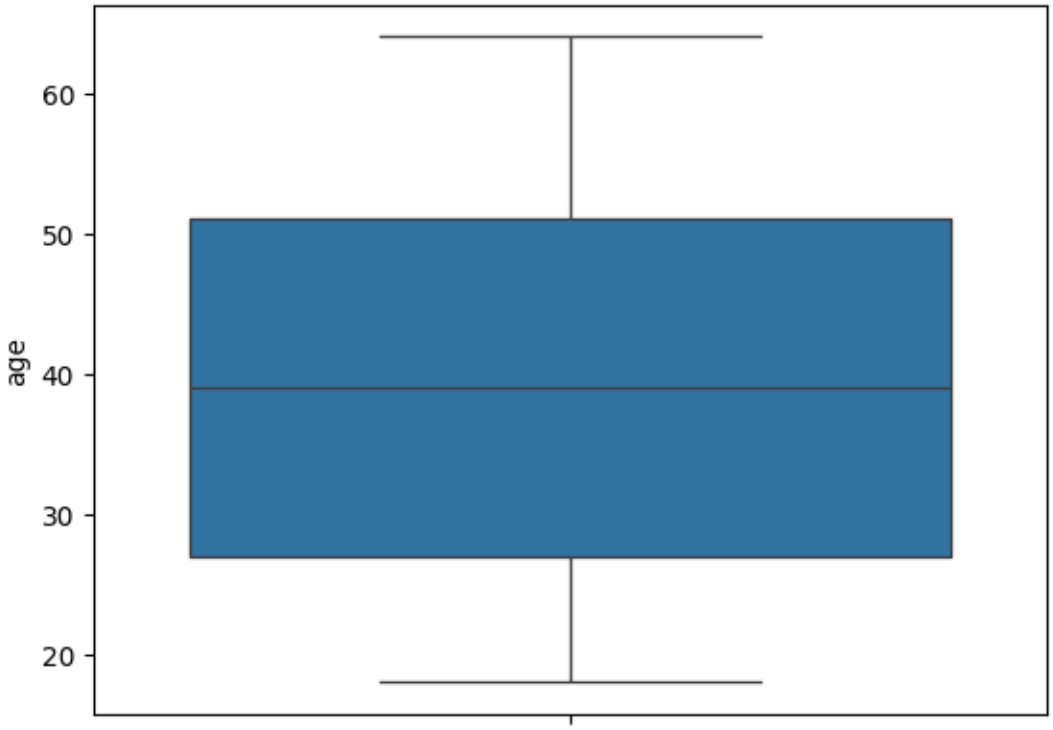
[21]:
```
sns.boxplot(data['bmi'])
```

[21]: <Axes: ylabel='bmi'>



[23]:
```
sns.boxplot(data['age'])
```

[23]: <Axes: ylabel='age'>



1 There are no outliers in age . It is cleaned data

[24]:
```
from sklearn.preprocessing import LabelEncoder
```

[25]:
```
lb = LabelEncoder()
```

[26]:
```
data.head()
```

[26]:

	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

[27]:
```
data['sex'] = lb.fit_transform(data['sex'])
data['smoker'] = lb.fit_transform(data['smoker'])
data['region'] = lb.fit_transform(data['region'])
```

[28]:
```
data.head()
```

[28]:

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

[29]:
```
x=data.drop(columns=['charges'],axis=1)
y=data["charges"]
```

[30]:
```
x
```

[30]:

	age	sex	bmi	children	smoker	region
0	19	0	27.900	0	1	3
1	18	1	33.770	1	0	2
2	28	1	33.000	3	0	2
3	33	1	22.705	0	0	1
4	32	1	28.880	0	0	1
…	…	…	…	…	…	…
1333	50	1	30.970	3	0	1
1334	18	0	31.920	0	0	0
1335	18	0	36.850	0	0	2
1336	21	0	25.800	0	0	3
1337	61	0	29.070	0	1	1

[1338 rows x 6 columns]

[31]:
```
y
```

[31]:
```
0       16884.92400
1        1725.55230
2        4449.46200
3       21984.47061
4        3866.85520
            …
1333    10600.54830
1334     2205.98080
1335     1629.83350
1336     2007.94500
1337    29141.36030
Name: charges, Length: 1338, dtype: float64
```

[32]:
```
from sklearn.preprocessing import StandardScaler
```

[33]:
```
sc = StandardScaler()
```

[36]:
```
X = sc.fit_transform(x)
X
```

[36]:
```
array([[-1.43876426, -1.0105187 , -0.45420102, -0.90861367,  1.97058663,
         1.34390459],
       [-1.50996545,  0.98959079,  0.51529985, -0.07876719, -0.5074631 ,
         0.43849455],
       [-0.79795355,  0.98959079,  0.38812512,  1.58092576, -0.5074631 ,
         0.43849455],
       …,
       [-1.50996545, -1.0105187 ,  1.02399878, -0.90861367, -0.5074631 ,
         0.43849455],
       [-1.29636188, -1.0105187 , -0.8010412 , -0.90861367, -0.5074631 ,
         1.34390459],
       [ 1.55168573, -1.0105187 , -0.2609615 , -0.90861367,  1.97058663,
        -0.46691549]])
```

[40]:
```
X = pd.DataFrame(x)
X
```

[40]:

	age	sex	bmi	children	smoker	region
0	19	0	27.900	0	1	3
1	18	1	33.770	1	0	2
2	28	1	33.000	3	0	2
3	33	1	22.705	0	0	1
4	32	1	28.880	0	0	1
…	…	…	…	…	…	…
1333	50	1	30.970	3	0	1
1334	18	0	31.920	0	0	0
1335	18	0	36.850	0	0	2
1336	21	0	25.800	0	0	3
1337	61	0	29.070	0	1	1

[1338 rows x 6 columns]

[41]:
```
from sklearn.model_selection import train_test_split
```

[42]:
```
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.
↪2,random_state=0)
```

[43]:
```
x_train.head()
```

[43]:

	age	sex	bmi	children	smoker	region
621	37	1	34.100	4	1	3
194	18	1	34.430	0	0	2
240	23	0	36.670	2	1	0
1168	32	1	35.200	2	0	3

```
1192   58    0  32.395         1       0       0
```

[44]:
```
y_train.head()
```

[44]:
```
621     40182.24600
194      1137.46970
240     38511.62830
1168     4670.64000
1192    13019.16105
Name: charges, dtype: float64
```

[45]:
```
x_test.head()
```

[45]:
```
      age  sex     bmi  children  smoker  region
578    52    1  30.200         1       0       3
610    47    0  29.370         1       0       2
569    48    1  40.565         2       1       1
1034   61    1  38.380         0       0       1
198    51    0  18.050         0       0       1
```

[46]:
```
y_test.head()
```

[46]:
```
578      9724.53000
610      8547.69130
569     45702.02235
1034    12950.07120
198      9644.25250
Name: charges, dtype: float64
```

[47]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
lr.fit(x_train,y_train)
```

[47]:
```
LinearRegression()
```

[50]:
```
y_pred= lr.predict(x_test)
y_pred
```

[50]:
```
array([ 1.10160603e+04,  9.77923984e+03,  3.80371234e+04,  1.61507863e+04,
        6.88293315e+03,  3.95067196e+03,  1.55123946e+03,  1.42956942e+04,
        8.95021753e+03,  7.44314397e+03,  4.50437538e+03,  1.02256118e+04,
        8.65146548e+03,  4.10852998e+03,  2.78244491e+04,  1.10218314e+04,
        1.12295093e+04,  6.05468839e+03,  8.17646090e+03,  2.70651298e+04,
        3.35621239e+04,  1.42697082e+04,  1.16839508e+04,  3.23986297e+04,
        4.44425117e+03,  9.18415186e+03,  1.08997992e+03,  1.00937120e+04,
        4.07628557e+03,  1.03609350e+04,  8.95258282e+03,  4.02848093e+04,
        1.54994342e+04,  1.36917690e+04,  2.47208893e+04,  5.12934575e+03,
```

```
 1.28727064e+04,  3.05630769e+04,  3.34394174e+04,  3.50793589e+03,
 3.96949616e+03,  4.28321470e+03,  3.04622802e+04,  3.93735907e+04,
 2.80418209e+04,  5.01890815e+03,  1.09122746e+04,  7.77082349e+03,
 3.58429161e+03,  1.05247646e+04,  5.65723135e+03,  3.37687152e+03,
 3.28442630e+04,  3.83787779e+04,  1.62889366e+04,  7.09748275e+03,
 6.00601567e+03,  9.38841693e+03,  9.23826629e+03,  1.16714163e+04,
 1.71082701e+03,  3.88109782e+04,  1.51836757e+04,  1.16481563e+04,
 1.39980927e+04,  1.38665326e+04,  2.60877220e+04,  3.21195731e+04,
 1.14698573e+03,  1.01034295e+04,  1.22080884e+04,  1.18229868e+04,
 2.50720978e+04,  1.58765422e+04,  1.11513510e+04,  1.26080449e+04,
 6.43045973e+03,  9.88242950e+03,  3.01298042e+04,  3.88952188e+04,
 1.20040720e+04,  3.74833085e+04,  4.18289591e+03,  9.24159400e+03,
 3.47566840e+04,  2.91306125e+04,  8.49192462e+03,  4.84994689e+03,
 1.18941530e+04,  3.02629710e+04,  1.00213468e+04,  1.12350417e+04,
 8.30887825e+03,  9.20289167e+03,  8.36614130e+03,  7.27672770e+03,
 3.59374611e+04,  3.30025677e+04,  7.58539901e+03,  1.49412347e+04,
 4.29603823e+03,  8.74143868e+03,  6.60986283e+03,  3.17848590e+04,
 3.28895198e+04,  1.93445524e+03,  8.89901349e+03,  6.60341975e+03,
 1.45131037e+04,  3.70378509e+04,  1.01045463e+04,  1.08253241e+04,
 1.01598083e+04,  2.69050692e+04,  4.01064892e+04,  8.43940085e+03,
 1.81895072e+02,  8.87199820e+03,  1.50812587e+04,  9.47938393e+03,
 3.53963113e+04,  7.23104553e+03,  1.67864609e+04,  9.59251469e+03,
 8.10818424e+03,  2.91525185e+03,  3.28539904e+04,  3.14103715e+04,
 3.93949811e+04,  5.50635406e+03,  9.60673704e+03,  3.87217751e+03,
 7.93601026e+03,  8.60701975e+03,  3.15453652e+04,  2.97829959e+04,
 3.00556463e+04,  9.04612972e+03,  3.27181565e+04,  3.30134482e+03,
 3.62928632e+03,  1.10659007e+04,  1.34140784e+04,  1.28052636e+04,
 5.39593232e+03,  1.58378860e+04,  1.51949117e+04,  2.37199184e+03,
-1.53478940e+01,  1.08186625e+04,  7.37844077e+03,  3.20597268e+04,
 1.23289897e+04,  2.63053425e+03,  6.37236260e+03,  8.15084926e+03,
 4.37249044e+03,  2.41142074e+03,  1.12983002e+04,  1.24616977e+04,
 7.22353675e+03,  1.66291178e+04,  1.17505932e+04,  1.39164717e+04,
 3.17628890e+03,  7.25188801e+03,  2.29759165e+04,  7.56011690e+03,
 5.46135935e+03,  5.45526991e+03,  6.68111566e+03,  5.17431994e+03,
 9.93348170e+03,  5.64510099e+03,  5.59569390e+03,  6.93584603e+03,
 3.69198381e+03,  5.52800182e+03,  3.79933114e+04,  1.47170243e+03,
 1.25618680e+04,  8.90103692e+03,  1.36860085e+04,  5.67520815e+03,
 5.18141582e+03,  3.63872865e+04,  4.34648808e+03,  1.91260592e+03,
 1.51300039e+04,  1.26272951e+04,  3.50484540e+04,  5.08034273e+03,
 5.54903853e+03,  3.15035795e+04,  6.11032691e+03,  2.01553140e+03,
 8.40194855e+03,  9.99099033e+03,  8.27113696e+03,  5.73931412e+03,
 1.31059342e+04,  3.87001294e+04,  1.36692833e+04,  2.87214797e+04,
 6.71502002e+03,  3.57620503e+04,  3.75151400e+03,  1.21586633e+04,
 9.33382099e+03,  6.49785967e+03,  1.12787558e+04,  1.45027542e+04,
 5.12216400e+03,  4.30162045e+03,  7.76282703e+03,  1.21875498e+03,
 7.83342284e+03,  4.40785528e+03,  1.31937312e+04,  4.25525066e+03,
 9.94294341e+03,  7.19110927e+03,  9.14504415e+03,  2.37953372e+03,
```

```
       1.30743547e+04,  1.67535765e+04,  1.52050035e+04,  1.04516323e+04,
       5.62621950e+03,  2.53885560e+03,  2.22514611e+03,  1.34269438e+04,
       1.43147777e+04,  4.99043823e+03,  4.07782835e+03,  9.33350069e+03,
       9.94493164e+03,  2.82052668e+04,  7.61429441e+03,  1.04558301e+04,
       6.17850512e+03,  2.97411340e+04,  1.09767114e+04,  7.47861108e+03,
       1.01696955e+04,  1.21677629e+04,  2.98984582e+03,  1.07846764e+04,
       1.52443098e+03,  7.01860008e+03,  2.87137133e+04,  3.84994154e+04,
       6.21151225e+03,  8.43854053e+03,  2.49741146e+03,  4.30913226e+02,
       1.04035596e+04,  4.46925909e+03,  4.91328289e+03,  2.66427677e+03,
       7.14505332e+03,  3.32349162e+04,  3.81534743e+04,  1.46868120e+04,
       8.21118778e+03,  1.60573354e+04,  3.31398626e+04,  9.45374170e+03])
```

[51]:
```
from sklearn.metrics import r2_score
an = r2_score(y_pred,y_test)
an
```

[51]:
```
0.7248156006571808
```