ML 5 - Handling Outliers By Virat Tiwari

November 7, 2023

1 HANDLING OUTLIERS

Outliers - Outliers distribution is totally different from the data distribution like in a dataset some datapoints are avilable in corner or some other place where they make there own distribution and thats distribution is called outliers .

Ex) Dataset - {1,2,3,6,5,4,100,200}

In this Dataset 100 and 200 are totally different from remaining datapoints so these two are our OUTLIERS .

Why we use " Handling outliers " - Outliers major impact the machine learning algorithm specifically linear regression some other algorithms due to which model reduce its accuracy and doesn't give high accuracy

TYPE 1 - 5 NUMBER SUMMARY CONCEPT -

1) MINIMUM VALUE,

2) Q1-25 PERCENTILE ,

3) MEDIAN ,

4) Q3-75 PERCENTILE ,

5) MAXIMUM ,

NOTE - BY CALCULATING ALL THESE 5 THINGS WE EASILY IDENTIFY THE OUTLIERS IN ANY DATASET

[1]:
```
# we import numpy that efficient for calculations

# Here we make a "list_marks" a dataset

import numpy as np
list_marks=[21,25,23,54,63,48,56,94,30,10,64,97,24,56,88,13,64,69,23,50,500,600,840,750]
```

[2]:
```
# len ( ) function is used for find the length of list or dataset

len(list_marks)
```

[2]: 24

[3]:
```
# np.percentile ( ) function is used for getting the percentile of numbers

# This function is built in function provided by numpy

np.percentile(list_marks,[100])
```

[3]:
```
array([840.])
```

[4]:
```
# We store np.percentile ( ) function in a variable "q1"

q1=np.percentile(list_marks,[25])
print(q1)
```

```
[24.75]
```

[5]:
```
# We create a lower fance and higher fance

# np.quantile ( ) is used for quantilastion od data in four parts that are␣
  ↪"minimum,q1,q2,q3,maximum"

minimum,q1,q2,q3,maximum=np.quantile(list_marks,[0,0.25,0.50,0.75,1.0])
```

[6]:
```
# maximum value

maximum
```

[6]:
```
840.0
```

[7]:
```
# minimum value

minimum
```

[7]:
```
10.0
```

[8]:
```
# lowest value

q1
```

[8]:
```
24.75
```

[9]:
```
q2
```

[9]:
```
56.0
```

[10]:
```
q3
```

[10]:
```
89.5
```

[11]:
```
# IQR stads for "Interquartile range" - It is the range between the first and third quartiles namely q1 and q3

IQR=q3-q1
print(IQR)
```

```
64.75
```

[12]:
```
# lower amd higher fence cordon off outliers from the bulk of data in set

lower_fence=q1-1.5*(IQR)
higher_fence=q3+1.5*(IQR)
```

[13]:
```
# Here we see lower and higher fence that shows the point like rhe values below -72.375 and value above 186.625 are the outliers

lower_fence,higher_fence
```

[13]:
```
(-72.375, 186.625)
```

[14]:
```
outlier=[]
for i in list_marks:
    if i>=-72.375 and i<=186.625:
        print("This elemnt is not an outlier")
    else:
        outlier.append(i)
```

```
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
This elemnt is not an outlier
```

[15]:
```
outlier
```

[15]:
```
[500, 600, 840, 750]
```

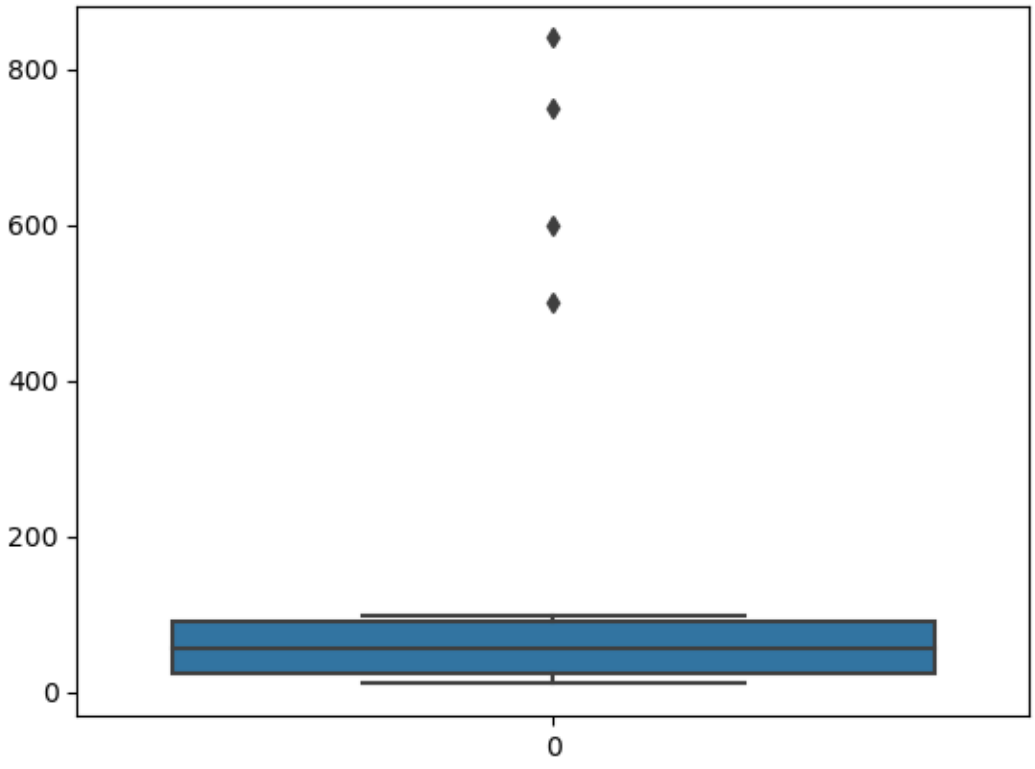
TYPE 2 - BOXPLOT

BOXPLOT - IT GIVES THE OUTLIERS AUTOMATICALLY BECOUSE IT CALCULATE EACH AND EVRYTHING INCLUDING PERCENTILE INSIDE IT

[16]:
```
# seaborn is used for visualizing the data in plot

import  seaborn as sns
```

[17]:
```
# 4 dotes shows the outliers

# sns.bpxplot  ( ) function is used for showing the outliers in a boxplot or in␣
 ↪graph manner

sns.boxplot(list_marks)
```

[17]:
```
<AxesSubplot: >
```



THANK YOU SO MUCH !!

VIRAT TIWARI :)