College of Computing and Information Sciences

Introduction of Data Science

Quiz-2

Total marks: 43

Name: Syed Aashir Majeed

ID: 10173

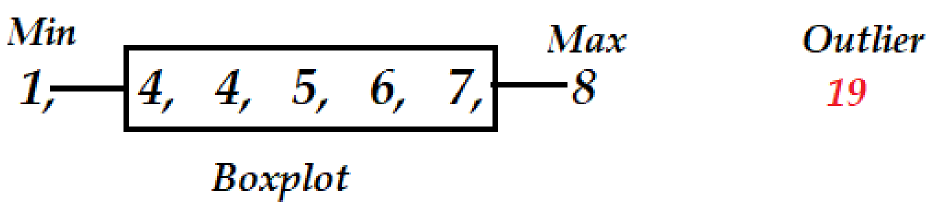
[Problem-1; marks=5] Write what is the outlier, and how it affects the machine learning algorithm.

Ans: Outliers are data points in a data set where there are abnormal observations among the normal observations and can lead to odd precision scores that can skew the measurements because the results are not representative of the true results. Outliers in the input data can distort and mislead the training process of machine learning algorithms, resulting in longer training times, less accurate models, and ultimately worse results. An unusual occurrence in the input data causes a machine learning model to provide false results, which is overfitting. Alternatively, the model can emphasize an illogical point. It is essential to remember that while some machine learning models may succeed even in the presence of outliers, others will utterly fail depending on how the model is built and designed.

[Problem-2; marks=5] Usually, we use boxplot to visualize the outliers; describe how it works.

Ans: If you sort the data from small to large, the center is the median. The median divides the data into two portions. The midpoints of each half are called "quartiles". So, we get two quartiles the 1st quartile is the midpoint of the first half and the 3rd quartile is the midpoint of the second half.

A boxplot provides several pieces of information, two important ones being the quartiles, represented by either end of the box. The distance between these two quartiles is called the Interquartile Range (IQR). In the boxplot, the length of the box is IQR, and the minimum and maximum values are represented by whiskers. The whiskers are generally extended to a distance of 1.5*IQR on each side of the box. Therefore, all data points outside these 1.5*IQR values are marked as outliers.



[Problem-3; marks=18] Consider the following data and answer the given questions (python code is not allowed) X = [24, 35, 19, 122, 41, 16, 136, 46, 132, 400, 28, 56, 329, 19, 274]

1. Find the boundaries (upper and lower) values of Whiskers.

sorted list = [16, 19, 19, 24, 28, 35, 41, 46, 56, 122, 132, 136, 274, 329, 400]

median = 46

First Quartile Q1 = 24

Second Quartile Q2 = 46

Third Quartile Q3 = 136

Interquartile Range IQR = Q3-Q1 = 112

Lower Bound = Q1 - 1.5*IQR = 24 - 1.5*112 = -144

Lower Whisker (LW) equals to minimum data observation value that is greater than or equal to Lower Bound.

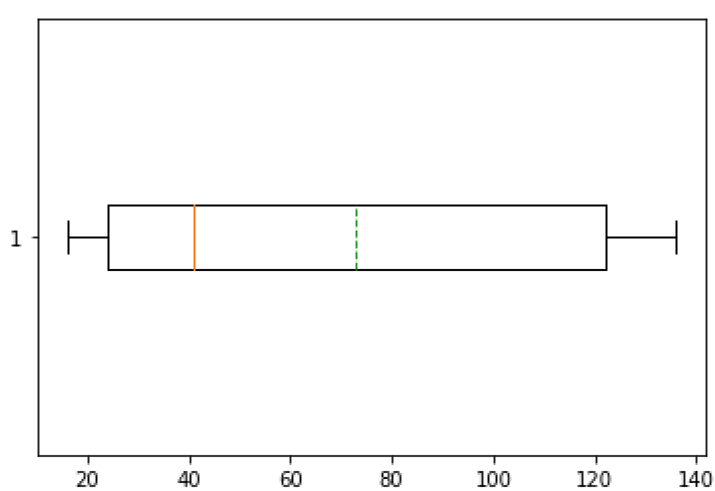
LW = 16

Upper Bound = Q3 + 1.5*IQR = 136 + 1.5*112 = 304

Upper Whisker (UW) equals to maximum data observation value that is less than or equal to Upper Bound.

UW = 274

2. Draw the boxplot according to boundary values calculated in part 1



3. Calculate the percentage of the outliers in the given data.

Values greater than Upper Bound or less than Lower Bound are considered to be outliers.

percentage = (outliers/total values) * 100 = (2/15) * 100 = 13.33%

4. Could you find the exact values of the outliers in the given data? What are those?

Ans: outliers are = 329,400

5. Write the name of possible handling methods of outliers.

Ans: Univariate method, Multivariate method, Minkowski error

6. Discuss the skewness of given data with help of the boxplot of part 2.

Skewness is an asymmetry measure of probability distribution of a real valued random variable. A positive skew specifies that the tail on the right side is longer than the left side and the size of the values lie to the left of the mean.

Data set = 16, 19, 19, 24, 28, 35, 41, 46, 56, 122, 132, 136, 274

Total number of elements = 13

formula of skewness = Σ (Yi = y) ^3/(n-1) ^3

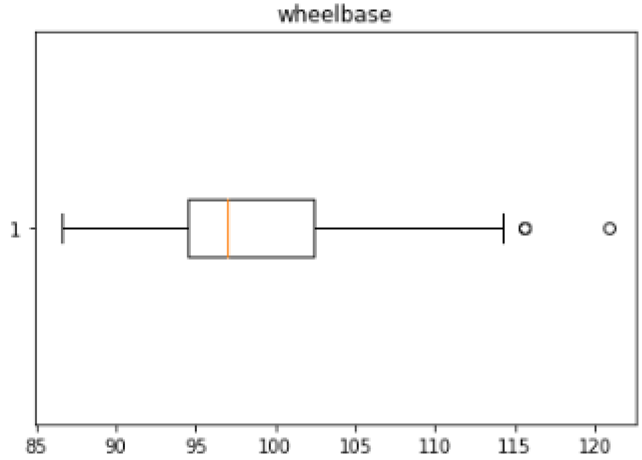
Skewness = 1.567

```python
#[Problem-4; marks=15]
#Read the dataset CarPrice.csv and write the python code for the following questions

df = pd.read_csv("CarPrice_Q2.csv")
df
```

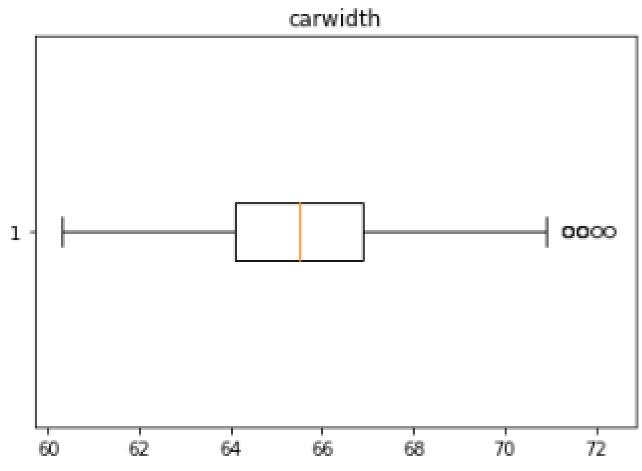
	car_ID	symboling	CarName	fueltype	aspiration	doornumber	carbody	drivewheel	enginelocation	wheelbase	...	enginesize	fuelsystem	boreratio
0	1	3	alfa-romero giulia	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
1	2	3	alfa-romero stelvio	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47
2	3	1	alfa-romero Quadrifoglio	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68
3	4	2	audi 100 ls	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19
4	5	2	audi 100ls	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
200	201	-1	volvo 145e (sw)	gas	std	four	sedan	rwd	front	109.1	...	141	mpfi	3.78
201	202	-1	volvo 144ea	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78
202	203	-1	volvo 244dl	gas	std	four	sedan	rwd	front	109.1	...	173	mpfi	3.58
203	204	-1	volvo 246	diesel	turbo	four	sedan	rwd	front	109.1	...	145	idi	3.01
204	205	-1	volvo 264gl	gas	turbo	four	sedan	rwd	front	109.1	...	141	mpfi	3.78

205 rows × 26 columns

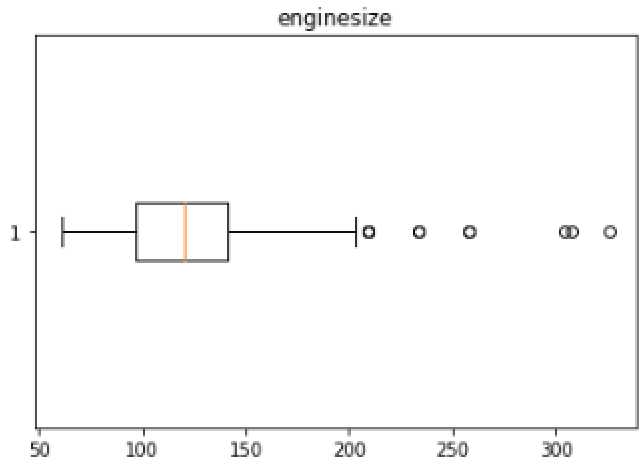
```
#1. Visualize the outliers in the following attributes. "wheelbase", "carwidth", and "enginesize"

plot.title('wheelbase')
plot.boxplot(df['wheelbase'],vert=False)
plot.show()
```




```
plot.title('carwidth')
plot.boxplot(df['carwidth'],vert=False)
plot.show()
```



```
plot.title('enginesize')
plot.boxplot(df['enginesize'],vert=False)
plot.show()
```




```
#2. Count the number of outliers in each attribute of part 1
#create a function to find outliers using IQR

def find_outliers_IQR(df):
    q1=df.quantile(0.25)
    q3=df.quantile(0.75)
    IQR=q3-q1
    outliers = df[((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]
    return outliers

outliers = find_outliers_IQR(df['wheelbase'])
print('number of outliers in wheelbase: '+ str(len(outliers)))
outliers = find_outliers_IQR(df['carwidth'])
print('number of outliers in carwidth: '+ str(len(outliers)))
outliers = find_outliers_IQR(df['enginesize'])
print('number of outliers in enginesize: '+ str(len(outliers)))
```

```
number of outliers in wheelbase: 3
number of outliers in carwidth: 8
number of outliers in enginesize: 10
```

```python
#3. If the outliers count is less than 4 in any above attributes remove it.

def drop_outliers(df):
    q1=df.quantile(0.25)
    q3=df.quantile(0.75)
    IQR=q3-q1
    not_outliers = df[~((df<(q1-1.5*IQR)) | (df>(q3+1.5*IQR)))]
    outliers_dropped = outliers.dropna().reset_index()

    return outliers_dropped

outliers = find_outliers_IQR(df['wheelbase'])

drop_outliers(df['wheelbase'])
```

	index	wheelbase
0	70	115.6
1	71	115.6
2	73	120.9

```python
df['wheelbase']
```

```
0       88.6
1       88.6
2       94.5
3       99.8
4       99.4
       ...
200    109.1
201    109.1
202    109.1
203    109.1
204    109.1
Name: wheelbase, Length: 205, dtype: float64
```

```python
#4. If the outliers count is greater than or equal to 4 in any above attributes, make it NaN followed by
#filling it with the appropriate filling method.

outliers = find_outliers_IQR(df['carwidth'])
df['carwidth'].replace(to_replace= [outliers], value = np.nan, inplace=True)
df['carwidth'].head(10)
```

```
0    64.1
1    64.1
2    65.5
3    66.2
4    66.4
5    66.3
6     NaN
7     NaN
8     NaN
9    67.9
Name: carwidth, dtype: float64
```

```python
df['carwidth'].fillna(df['carwidth'].interpolate(), inplace=True)
df['carwidth'].head(10)
```

```
0    64.1
1    64.1
2    65.5
3    66.2
4    66.4
5    66.3
6    66.7
7    67.1
8    67.5
9    67.9
Name: carwidth, dtype: float64
```

```
outliers = find_outliers_IQR(df['enginesize'])
df['enginesize'].replace(to_replace= [outliers], value = np.nan, inplace=True)
df['enginesize'].head(20)
```

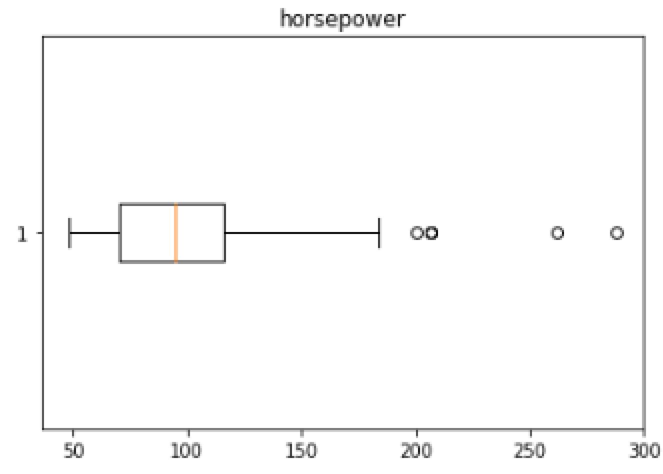
```
0     130.0
1     130.0
2     152.0
3     109.0
4     136.0
5     136.0
6     136.0
7     136.0
8     131.0
9     131.0
10    108.0
11    108.0
12    164.0
13    164.0
14    164.0
15      NaN
16      NaN
17      NaN
18     61.0
19     90.0
Name: enginesize, dtype: float64
```

```
df['enginesize'].fillna(df['enginesize'].interpolate(), inplace=True)
df['enginesize'].head(20)
```

```
0     130.00
1     130.00
2     152.00
3     109.00
4     136.00
5     136.00
6     136.00
7     136.00
8     131.00
9     131.00
10    108.00
11    108.00
12    164.00
13    164.00
```

```
5      136.00
6      136.00
7      136.00
8      131.00
9      131.00
10     108.00
11     108.00
12     164.00
13     164.00
14     164.00
15     138.25
16     112.50
17      86.75
18      61.00
19      90.00
Name: enginesize, dtype: float64
```

```
#5. Visualize the outlier in the "horsepower" attribute, and remove it by the binning method.

plot.title('horsepower')
plot.boxplot(df['horsepower'],vert=False)
plot.show()
```



```python
outliers = find_outliers_IQR(df['horsepower'])
outliers
outliers = np.linspace(outliers,4)
df['horsepower'] = pd.cut(df['horsepower'],bins=outliers)
df['horsepower']
```

```
0      111
1      111
2      154
3      102
4      115
      ...
200    114
201    160
202    134
203    106
204    114
Name: horsepower, Length: 205, dtype: int64
```