DataScience® foundation
MAKE BETTER DECISIONS

Knowing all about Outliers in Machine Learning

Author, Mayank Tripathi

A Data Science Foundation White Paper

June 2020

www.datascience.foundation

While working on various datasets to train a Machine Learning model. What is it, that you look for? What is the most important part of the Exploratory Data Analysis (EDA) phase? There are certain things which, if they are not done in the EDA phase, can affect further statistical / Machine Learning modelling.

One of the answers is to find the **"Outliers".**

In this post we will try to understand all about outliers by answering the following questions, and at the end of the paper, will use Python to create some examples.

- What Outlier is?
- How the Outlier are introduced in the datasets?
- How to detect Outliers?
- Why is it important to identify the outliers?
- What are the types of Outliers?
- What are the methods to prevent outliers?

What Outlier is?

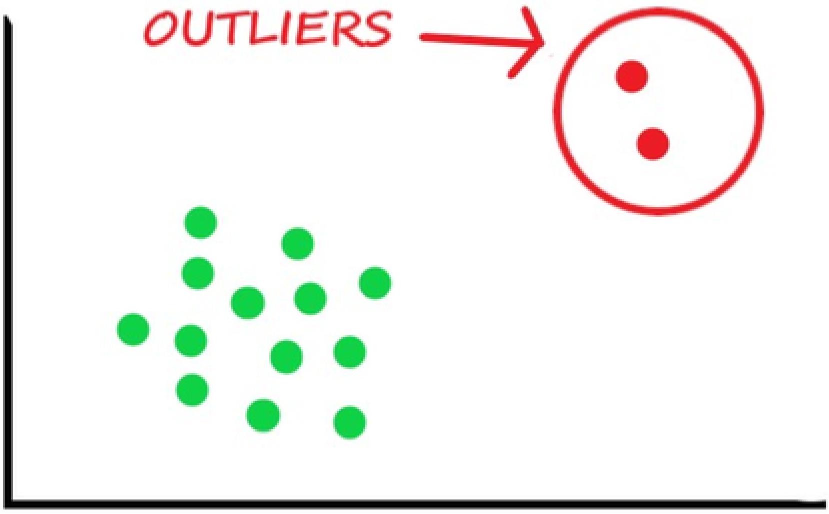
As per Wikipedia definition,

'*In statistics, an* ***Outlier*** *is an observation point that is distant from other observations.*'

The definition suggests to us that an outlier is something which is an odd-one-out or the one that is different from the crowd. Some statisticians define outliers as ‘having a different underlying behavior than the rest of the data’. Alternatively, an outlier is a data point that is distant from other points.

Please don’t confuse this definition with that of an Imbalanced dataset, though there are some similarities in the definitions. We will not going into much detail on this for now, and will have separate article on Imbalanced datasets later: An imbalanced data set in terms of machine learning is where one class label has far fewer samples compared to another class label.

From the image below we can see that the sample points in Green are close to each other, whereas the two sample points in Red are far apart from them. These red sample points are outliers.



How the Outlier are introduced in the datasets?

Now we know what outlier is. Are you also wondering how outlier are introduced to the population or dataset? The Outlier may be due to just variability in the measurement or may indicate experimental errors.

Outliers are first introduced to the population while gathering or collecting the data. Data gathering or data collection is itself a big topic to discuss. Will not be going into this now, but just to share some facts. Data can be collected in many ways be it via Interview; Questionnaires & Survey; Observations; Documents & Records; Focus groups; Oral History etc., and in this Tech era Internet; IT sensors etc., are generating data for us.

Another possible cause of outliers could be Incorrect entry; Misreporting of data or observations; Sampling errors while doing the experiment; Exceptional but True value. Though, you will not be aware of the outliers at in the collection phase. The outliers can be a result of a mistake during data collection or they can be just an indication of variance in your data.

If possible, outliers should be excluded from the data set. However, detecting anomalous instances might be difficult, and is not always possible. Data Science Developers and statisticians don't like to declare outliers too quickly. The 'too large' number could be a data entry error, a scale problem, or just a really big number.

The low income could be real, an error, or a confusion of household and inpidual income. And sometimes you get zeros–often, 'No', – or answers to surveys which seem questionable. Statisticians care about outliers from the point of view of how they impact the analysis.

Type of Outliers

There are mainly 3 types of Outliers.

1. **Point or global Outliers:** Observations anomalous with respect to the majority of observations in a feature. In-short A data point is considered a global outlier if its value is far outside the entirety of the data set in which it is found.

   Example: In a class all student age will be approx. similar, but if see a record of a student with age as 500. It's an outlier. It could be generated due to various reason.

2. **Contextual (Conditional) Outliers:** Observations considered anomalous given a specific context.A data point is considered a contextual outlier if its value significantly deviates from the rest of the data points in the same context. Note that this means that same value may not be considered an outlier if it occurred in a different context. If we limit our discussion to time series data, the "context" is almost always temporal, because time series data are records of a specific quantity over time. It's no surprise then that contextual outliers are common in time series data.In Contextual Anomaly values are not outside the normal global range but are abnormal compared to the seasonal pattern.

   Example: World economy falls drastically due to COVID-19. Stock Market crashes due to the scam in 1992; in 2020 due to COVID-19. Usual data points will be near to each other whereas data point during the specific period will either up or down very far. This is not due to erroneous, but it's an actual observation data point.

3. **Collective Outliers:** A collection of observations anomalous but appear close to one another because they all have a similar anomalous value.

   A subset of data points within a data set is considered anomalous if those values as a collection deviate significantly from the entire data set, but the values of the inpidual data points are not themselves anomalous in either a contextual or global sense. In time series data, one way this can manifest is as normal peaks and valleys occurring outside of a time frame when that seasonal sequence is normal or as a combination of time series that is in an outlier state as a group.

Why is it important to identify the outliers?

Machine learning algorithms are sensitive to the range and distribution of attribute values. Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results.

For more details refer to https://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/

How to detect Outliers?

Various techniques have been proposed for dealing with outliers, which may be grouped into two broad approaches, namely algorithm level and data level techniques.

The former aims to modify a learning algorithm to cope with the dataset and are known to have relatively high computational cost.

The latter is classifier-independent and relatively easy to apply because it focuses on data preprocessing techniques.

For example, to deal with outliers, some researchers identify and remove them completely, whereas others control the number of outliers to remove.

These are advanced level techniques, which we will cover in another article, instead in this will take the beginners level approach.

An Outlier can easily be detected using below techniques.

1. Visualization Technique
2. Visualization Technique

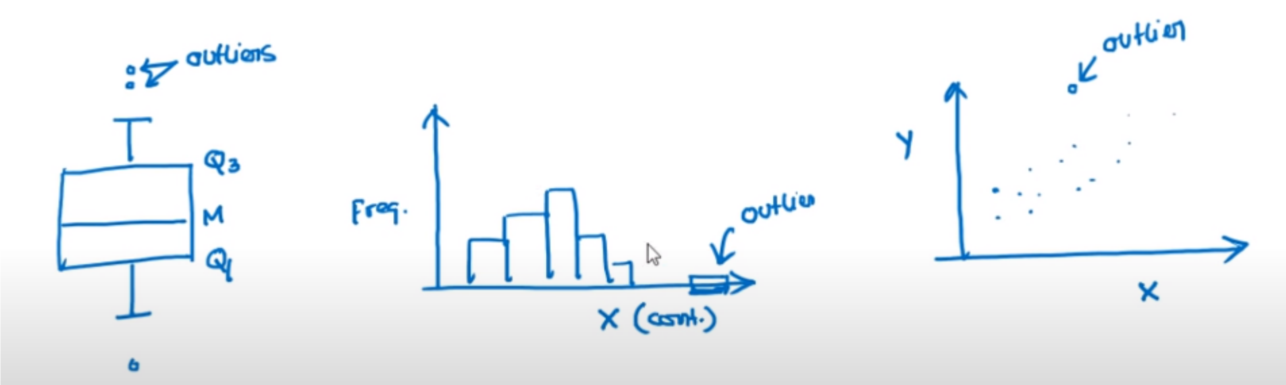
Visualization Technique

When we say visualization technique, it does not mean detection by the naked eye. Yes, might do that, but only if the sample data points are few and distinctive. For example, being presented with images of animals and having to identify the odd-man-out. Or from the emojis shown in the first image one can easily detect which one is different: smiling and in yellow color. But real-life data points are not like this, they are collected in millions, possibly billions of records. So, we will use plotting or graphical representation techniques to visualize the outliers.

The very first is a Box Plot.

A box plot is a graphical display for describing the distribution of data. Box plots use the median and the lower and upper quartiles.

An outlier can easily be detected via Box plot where any point above or below the whiskers represent an outlier. This is also known as “Univariate method” as here we are using one variable outlier analysis.



Note: Box Plot will also be used for multiple variable if we have categorical values.

Outliers can also be spotted using a Histogram, where bulk of observations are on one side, and a few observations appear away from the main group, these represent the outliers.

They can also be spotted using a scatter plot, which helps to understand the degree of associations between two numerical variables, and any observation which is far from the normal association, is an outlier.

You are free to use any kind of plot, as long as you are able to visualize and detect it. The above plotting methods are commonly used.

With mathematical functions

Along with Visual techniques, we could also use some mathematical functions to detect outliers:

- Z-Score
  Wikipedia Definition - The Z-score is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured.

  The intuition behind a Z-score is to describe any data point by finding their relationship with the Standard Deviation and Mean of the group of data points. A Z-score is finding the distribution of data where mean is 0 and standard deviation is 1 i.e. normal distribution.

While calculating the Z-score, re-scale and center the data and look for data points which are too far from zero. These data points which are way too far from zero will be treated as the outliers. In most of the cases a threshold of 3 or -3 is used i.e., if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be identified as outliers.

- IQR (Inter Quartile Range) Score
  Wikipedia Definition - The interquartile range (IQR), also called the mid-spread or middle 50%, or technically H-spread, is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles,

  $$IQR = Q3 - Q1.$$

  In other words, the IQR is the first quartile subtracted from the third quartile; these quartiles can be clearly seen on a box plot on the data. It is a measure of the dispersion like a standard deviation or variance but is much more robust against outliers. IQR is somewhat like Z-score in terms of finding the distribution of data and then keeping some threshold to identify the outlier.

Note: There are too many techniques to cover all of them in one short article.

What are the methods to prevent outliers?

Just how much an outlier affects your analysis depends, not surprisingly, on a few factors.

One factor is dataset size: In a large dataset, each inpidual point carries less weight, so an outlier is less worrisome than the same data point would be in a smaller dataset.

Another consideration is “how much” of an outlier a point might be – just how far out of line with the rest of your dataset a single point is. A point that is ten times as large as your upper boundary will do more damage than a point that is twice as large.

Again, the best way to guard against outliers is your Domain knowledge and experience with outliers.

Here are few tips.

- Drop the outlier records.
  Sometimes it's best to completely remove those records from your dataset to stop them from skewing your analysis.
- Cap your outliers' data.
  Another way to handle true outliers is to cap them. For example, if you're using income, you might find that people above a certain income level behave in the same way as those with a lower income. In this case, you can cap the income value at a level that keeps that intact.
- Assign a new value.
  If an outlier seems to be due to a mistake in your data, try imputing a new value. Common imputation methods include using the mean of a variable or utilizing a regression model to predict the missing value.
- Try a transformation.
  A different approach to true outliers could be to try creating a transformation of the data rather than using the data itself.

  For example, try creating a percentile version of your original field and working with that new field instead.

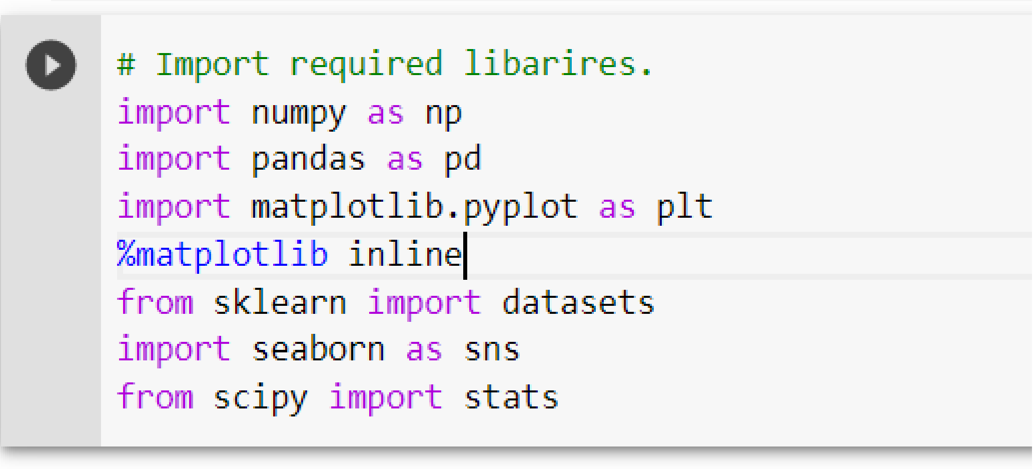
- **Mathematical Functions – to identify and drop the outliers**
  Z-score
  IQR Score

Working with Python

To work on a real-world data set, lets start with the Boston Housing Price dataset from Sklearn library.

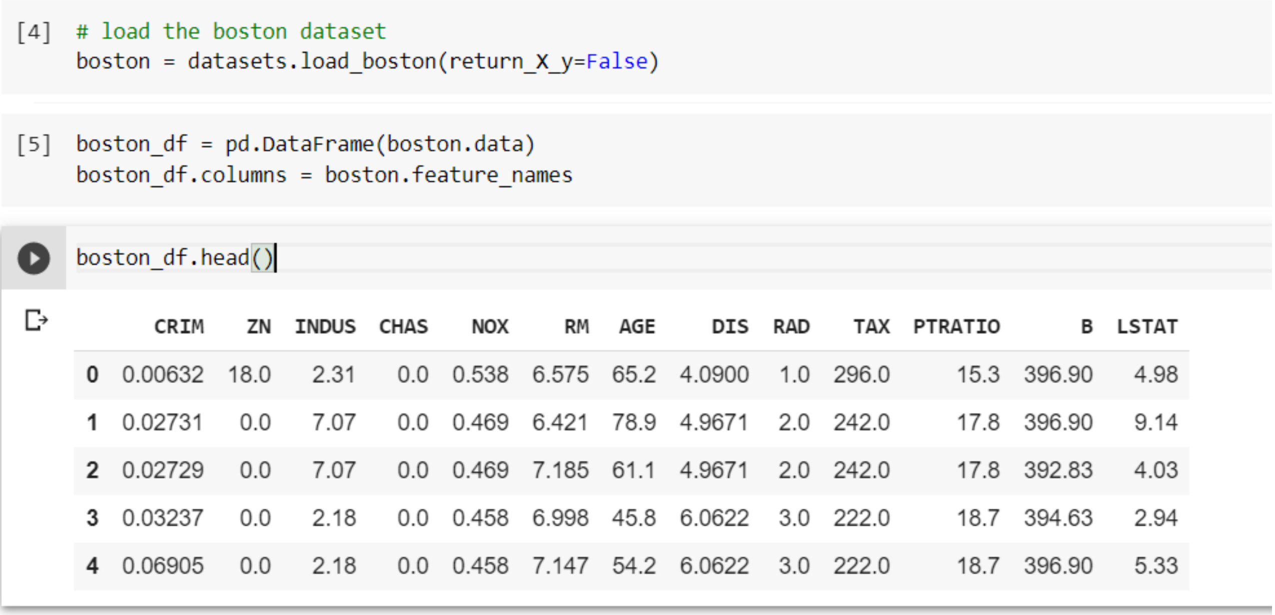
This dataset can be taken from https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_boston.html

As usual will start with importing the required libraries.



```
# Import required libarires.
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn import datasets
import seaborn as sns
from scipy import stats
```

Then we will upload the dataset into boston_df dataframe. This is same example which we have seen before in Linear Regression.



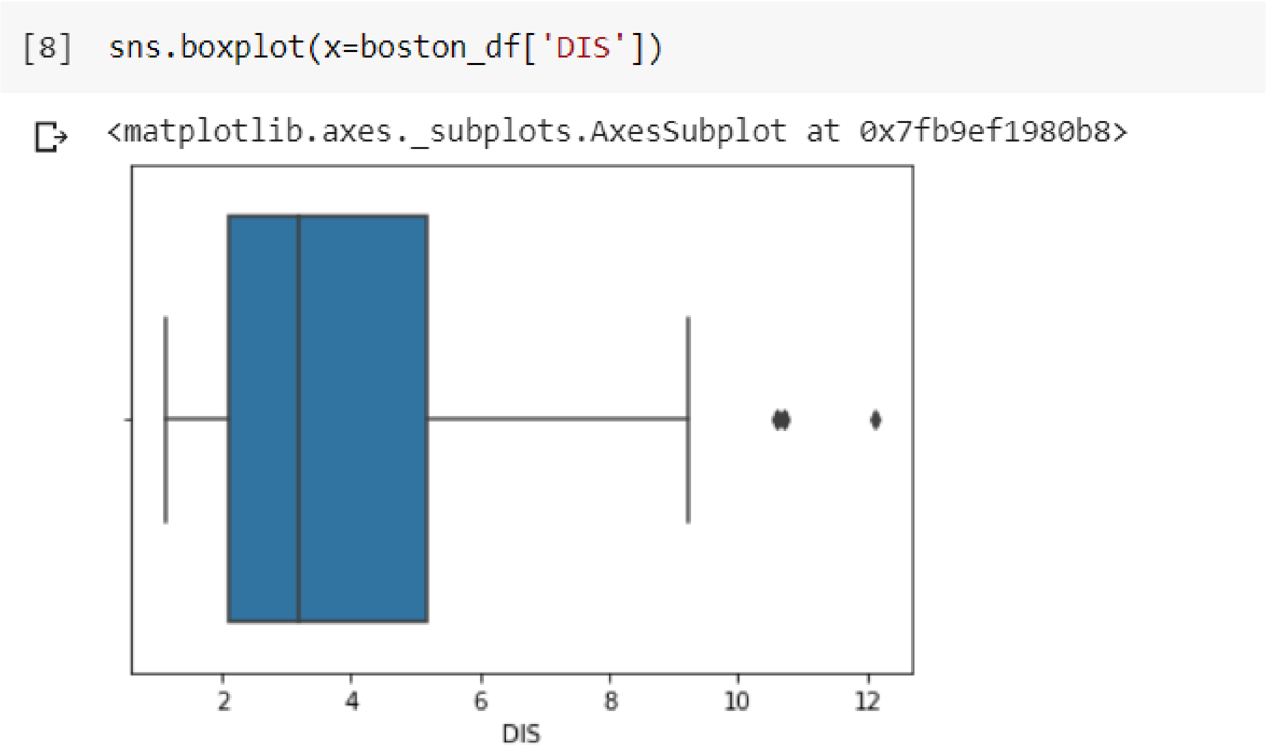
```
[4]  # load the boston dataset
     boston = datasets.load_boston(return_X_y=False)

[5]  boston_df = pd.DataFrame(boston.data)
     boston_df.columns = boston.feature_names

     boston_df.head()

        CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  PTRATIO       B  LSTAT
0    0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0     15.3  396.90   4.98
1    0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0     17.8  396.90   9.14
2    0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0     17.8  392.83   4.03
3    0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0     18.7  394.63   2.94
4    0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0     18.7  396.90   5.33
```

Let’s identify the outliers using an easy visualization technique which is Box Plot. For this we will use DIS column only to check the outlier. As this is a Uni-variate outlier.



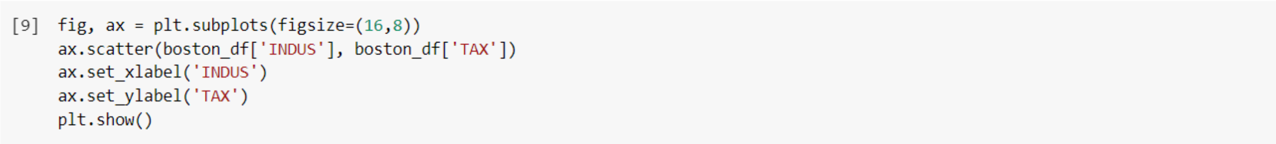
```
[8]  sns.boxplot(x=boston_df['DIS'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb9ef1980b8>
```

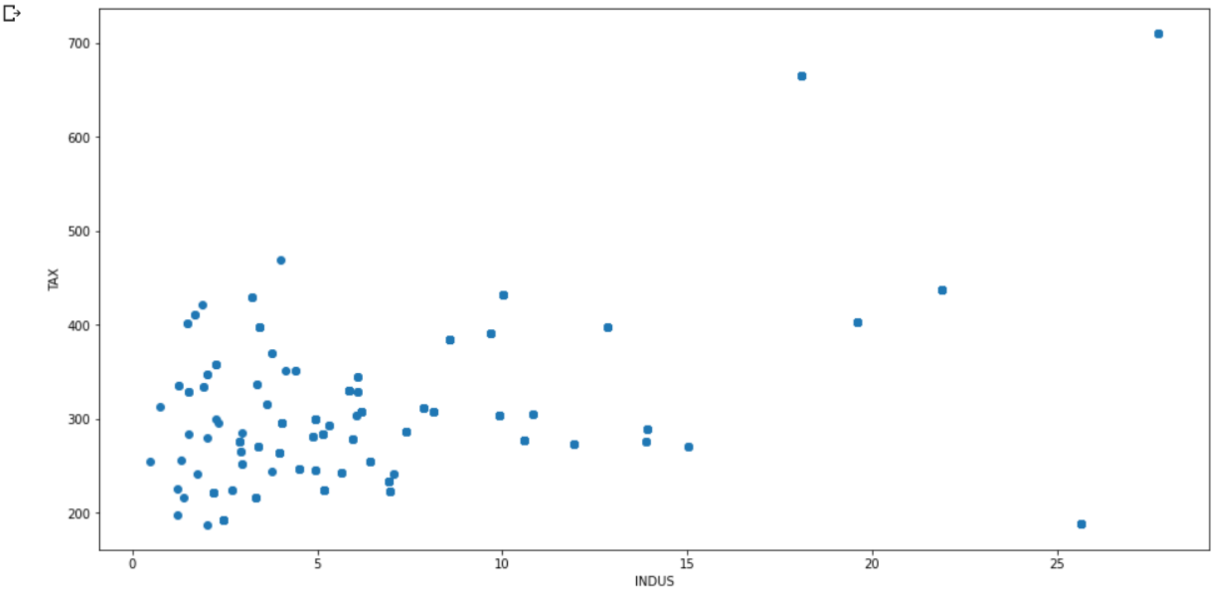
Based on the definition already discussed, that if there is an outlier it will plotted as point in boxplot, while the rest of the population will be grouped together and display as boxes. And the above plot shows three points between 10 to 12, these are the outliers as they are not included in the box of other normal observation.

Can we use multiple features with Box plot? Well it depends, if we have categorical features then we could have use that with any continuous variable and do multivariate outlier analysis. As we do not have categorical value in our Boston Housing dataset, we will jump on to the Scatter Plot.

Based on the definition discussed above for scatter plost to identify the outlier, the scatter plot is the collection of points that shows values for two variables. Let’s try and draw scatter plot for two variables from our housing dataset.



```
[9]  fig, ax = plt.subplots(figsize=(16,8))
     ax.scatter(boston_df['INDUS'], boston_df['TAX'])
     ax.set_xlabel('INDUS')
     ax.set_ylabel('TAX')
     plt.show()
```

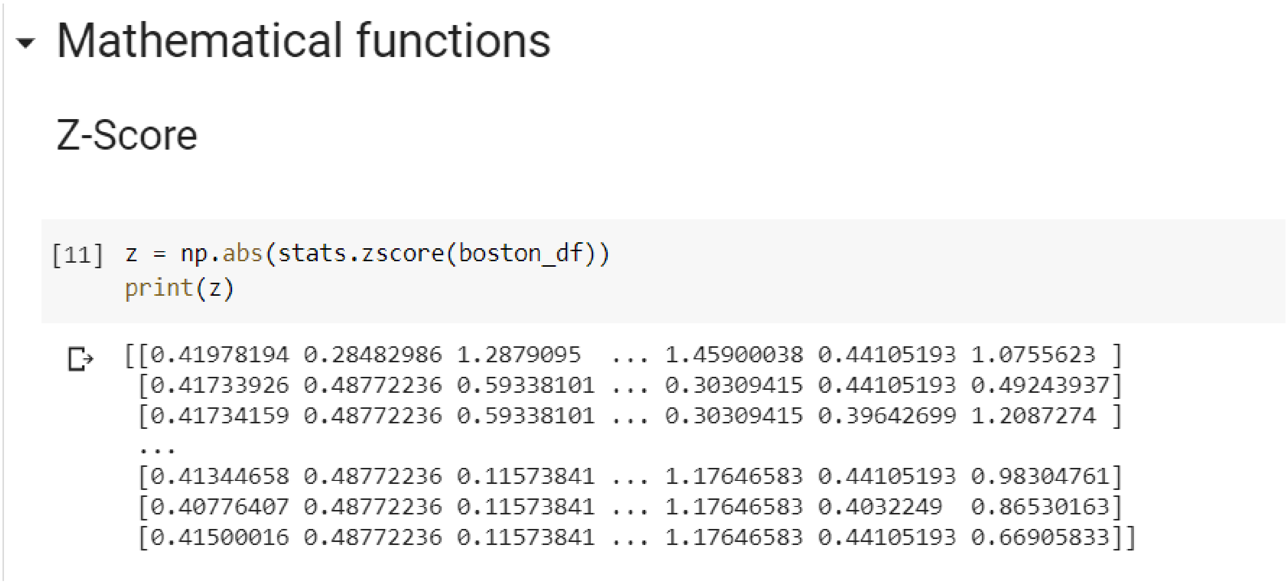


From the above plot, we can see most of the data points are clustered at bottom left side but there are few points which are far from the population, like top right corner. These are the Outliers.

Here I have used specific features, please play around with other features.

Now let’s discover the outliers with mathematical functions.

We start with Z-score function defined in scipy library to detect the outliers.



Mathematical functions

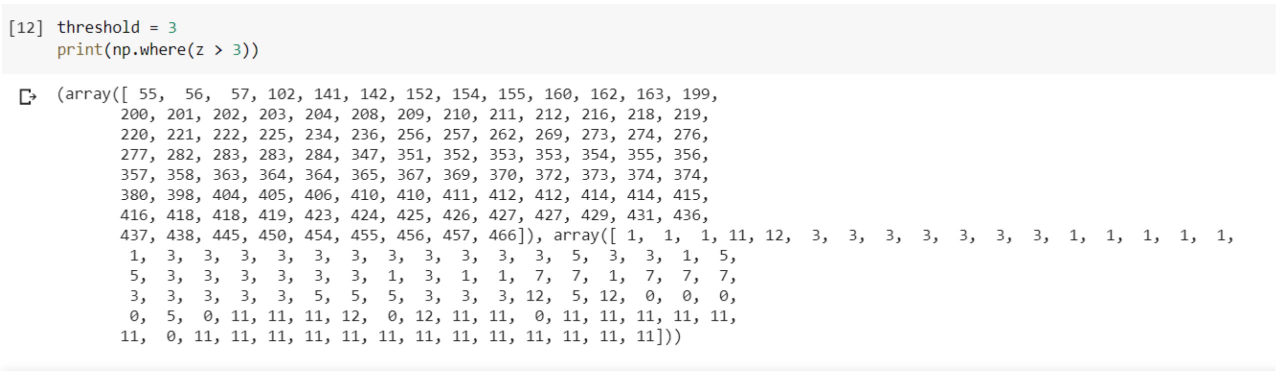
Z-Score

```
[11] z = np.abs(stats.zscore(boston_df))
     print(z)
```

```
[[0.41978194 0.28482986 1.2879095  ... 1.45900038 0.44105193 1.0755623 ]
 [0.41733926 0.48772236 0.59338101 ... 0.30309415 0.44105193 0.49243937]
 [0.41734159 0.48772236 0.59338101 ... 0.30309415 0.39642699 1.2087274 ]
 ...
 [0.41344658 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.98304761]
 [0.40776407 0.48772236 0.11573841 ... 1.17646583 0.4032249  0.86530163]
 [0.41500016 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.66905833]]
```

Looking the code and the output above, it is difficult to say which data point is an outlier.

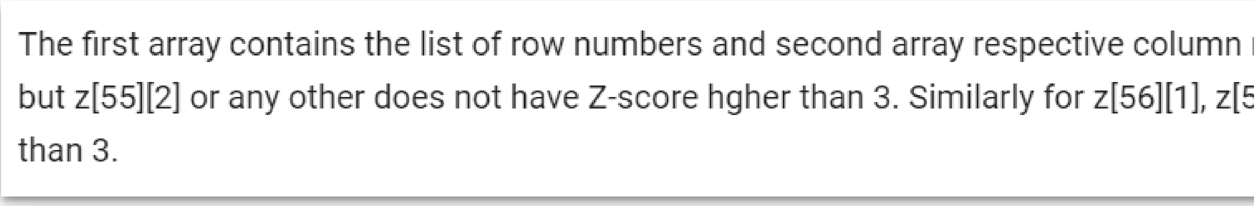
Let’s try and define a threshold to identify an outlier. As discussed above the default or usual value is 3.



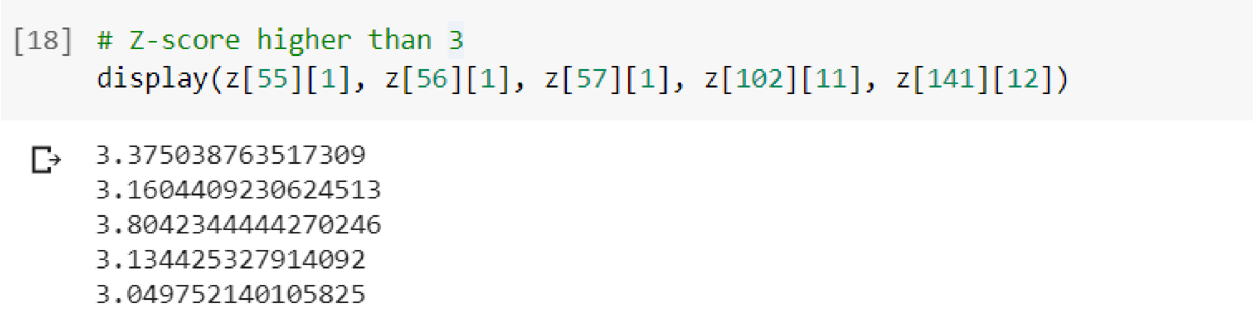
```
[12] threshold = 3
     print(np.where(z > 3))
```

```
(array([ 55,  56,  57, 102, 141, 142, 152, 154, 155, 160, 162, 163, 199,
        200, 201, 202, 203, 204, 208, 209, 210, 211, 212, 216, 218, 219,
        220, 221, 222, 225, 234, 236, 256, 257, 262, 269, 273, 274, 276,
        277, 282, 283, 283, 284, 347, 351, 352, 353, 353, 354, 355, 356,
        357, 358, 363, 364, 364, 365, 367, 369, 370, 372, 373, 374, 374,
        380, 398, 404, 405, 406, 410, 410, 411, 412, 412, 414, 414, 415,
        416, 418, 418, 419, 423, 424, 425, 426, 427, 427, 429, 431, 436,
        437, 438, 445, 450, 454, 455, 456, 457, 466]), array([ 1,  1,  1, 11, 12,  3,  3,  3,  3,  3,  3,  3,  1,  1,  1,  1,  1,
         1,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  3,  5,  3,  3,  1,  5,
         5,  3,  3,  3,  3,  3,  3,  1,  3,  1,  1,  7,  7,  1,  7,  7,  7,
         3,  3,  3,  3,  3,  5,  5,  5,  3,  3,  3, 12,  5, 12,  0,  0,  0,
         0,  5,  0, 11, 11, 11, 12,  0, 12, 11, 11,  0, 11, 11, 11, 11, 11,
        11,  0, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11, 11]))
```

Still I could not understand much from the output. Don’t worry. The first array contains the list of row numbers and second array respective column numbers, which mean z[55][1] have a Z-score higher than 3. but z[55][2] or any other does not have Z-score higher than 3. Similarly for z[56][1], z[57][1], z[102][11], z[141][12] etc. are having z-score higher than 3.



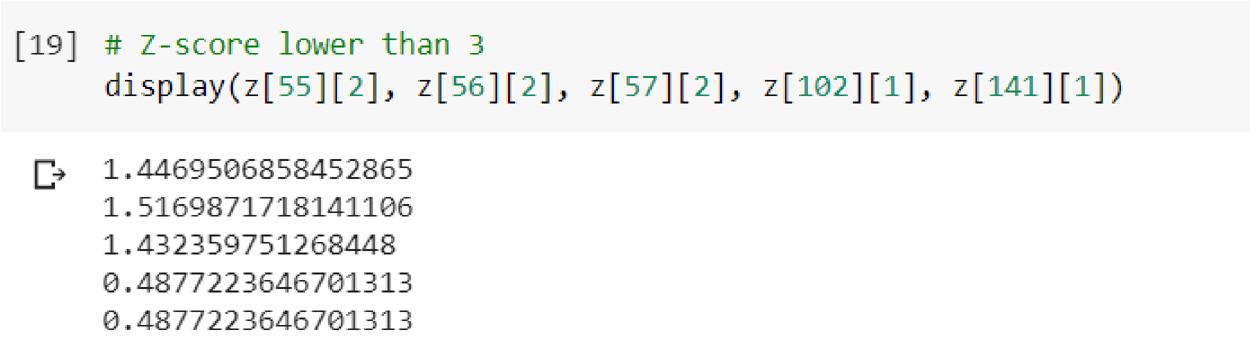
The first array contains the list of row numbers and second array respective column
but z[55][2] or any other does not have Z-score hgher than 3. Similarly for z[56][1], z[5
than 3.



```
[18] # Z-score higher than 3
     display(z[55][1], z[56][1], z[57][1], z[102][11], z[141][12])

     3.375038763517309
     3.1604409230624513
     3.8042344444270246
     3.134425327914092
     3.049752140105825
```

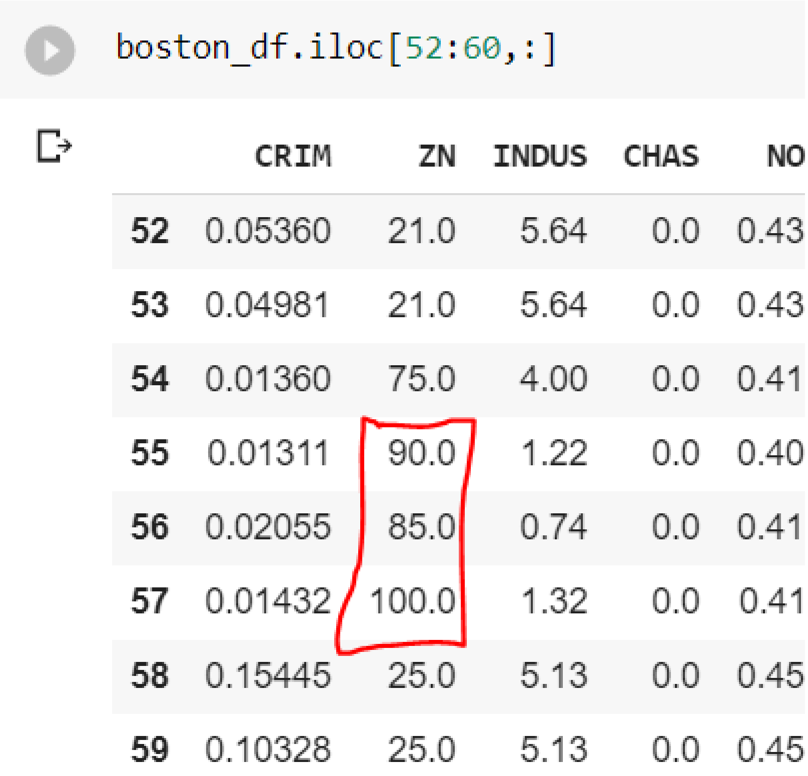
Here I tried to display a few of them… also one to verify the records with lower Z-score, we could do as below.



```
[19] # Z-score lower than 3
     display(z[55][2], z[56][2], z[57][2], z[102][1], z[141][1])

     1.4469506858452865
     1.5169871718141106
     1.432359751268448
     0.4877223646701313
     0.4877223646701313
```

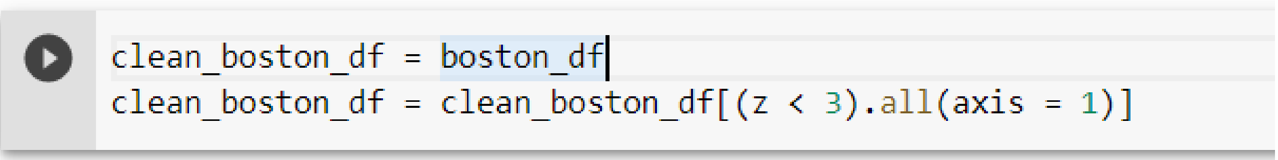
Let’s understand this in more details. Limiting down to one feature “ZN”. From above for row number 55 and column 1 has Z-Score higher than 3 (threshold) and from below we can see that that’s true. The rest of the values are in 20’s whereas for row # 55 its 90 which is clearly indicates that this value is far from being a usual value, and thus mathematically we have proved its an outlier.



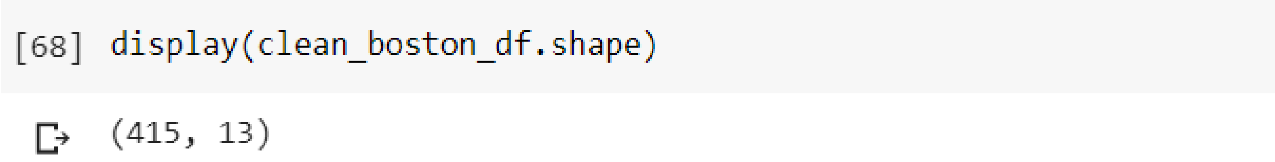
```
boston_df.iloc[52:60,:]
```

	CRIM	ZN	INDUS	CHAS	NO
52	0.05360	21.0	5.64	0.0	0.43
53	0.04981	21.0	5.64	0.0	0.43
54	0.01360	75.0	4.00	0.0	0.41
55	0.01311	90.0	1.22	0.0	0.40
56	0.02055	85.0	0.74	0.0	0.41
57	0.01432	100.0	1.32	0.0	0.41
58	0.15445	25.0	5.13	0.0	0.45
59	0.10328	25.0	5.13	0.0	0.45

Once we know which records are Outliers, we can easily remove them from our dataset if required.



```
clean_boston_df = boston_df
clean_boston_df = clean_boston_df[(z < 3).all(axis = 1)]
```



```
[68] display(clean_boston_df.shape)

(415, 13)
```

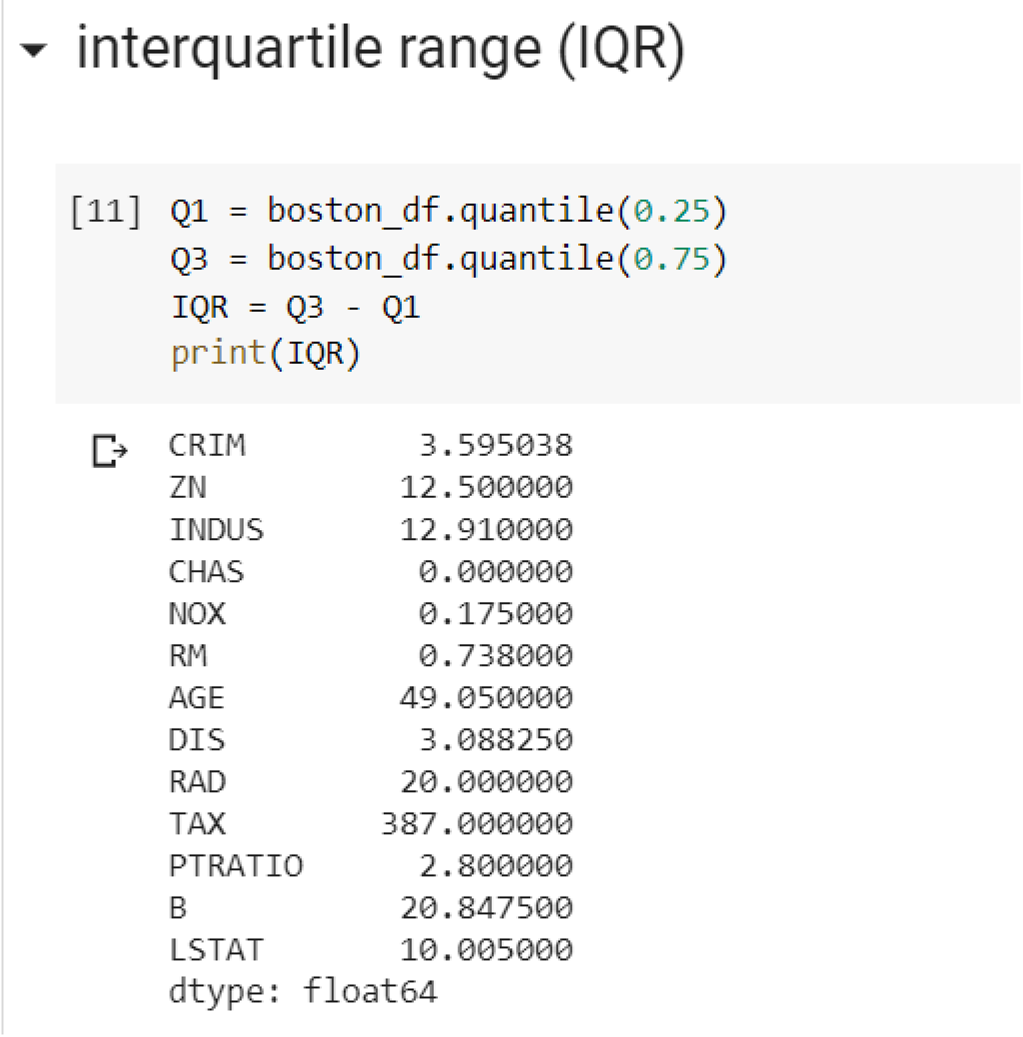
So, after removing the Outlier our dataset will have 415 rows. It means we have removed almost (506-415) 91 records from the dataset.

Another mathematical function is IQR (Inter Quartile Range). Python gave us the method quantile

which we directly use or alternatively we can use a method from numpy which is percentile.

The next step is to identify lower bound value and upper bound value.

Any value away from this lower and upper bound value is considered as an outlier.



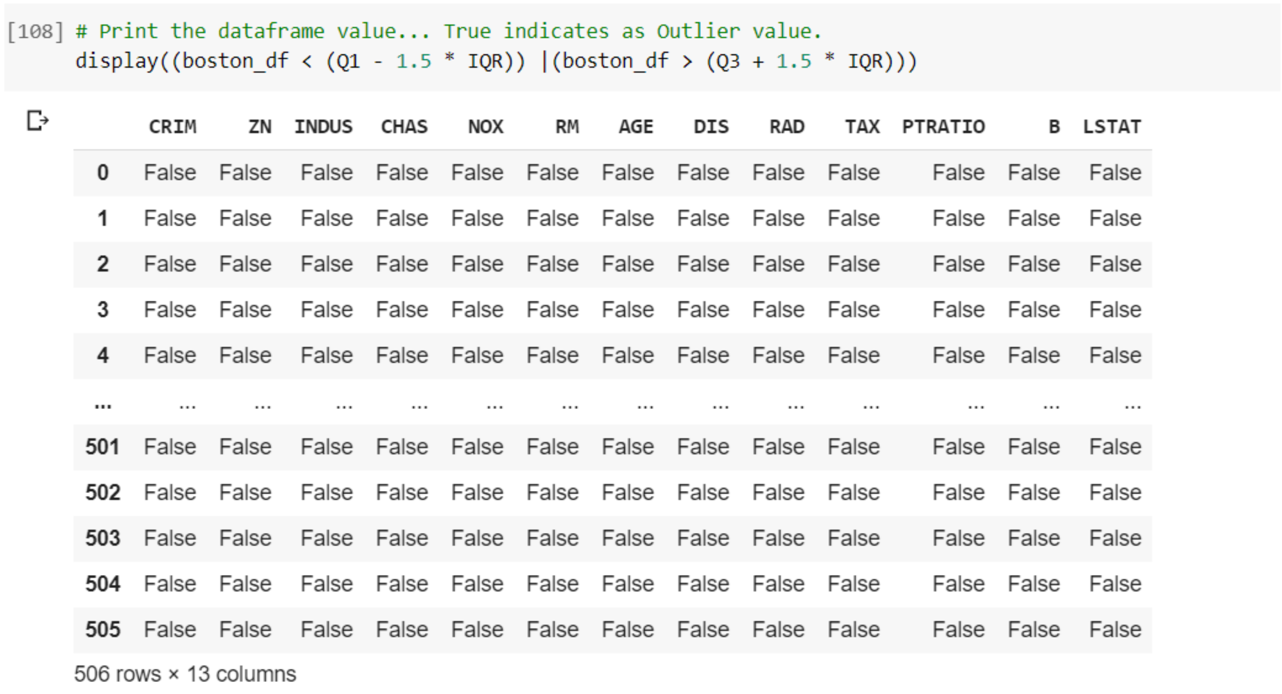
interquartile range (IQR)

```
[11] Q1 = boston_df.quantile(0.25)
     Q3 = boston_df.quantile(0.75)
     IQR = Q3 - Q1
     print(IQR)
```

```
CRIM         3.595038
ZN          12.500000
INDUS       12.910000
CHAS         0.000000
NOX          0.175000
RM           0.738000
AGE         49.050000
DIS          3.088250
RAD         20.000000
TAX        387.000000
PTRATIO      2.800000
B           20.847500
LSTAT       10.005000
dtype: float64
```

Once we have the quantile values, we will subtract Q3 from Q1 to get the IQR Value.

Next is to calculate the Lower Bound (Q1 – 1.5 * IQR) and Upper Bound (Q3 – 1.5 * IQR) Value, as based on this will check the dataset which is outlier indicated as True.



```
[108] # Print the dataframe value... True indicates as Outlier value.
      display((boston_df < (Q1 - 1.5 * IQR)) |(boston_df > (Q3 + 1.5 * IQR)))
```

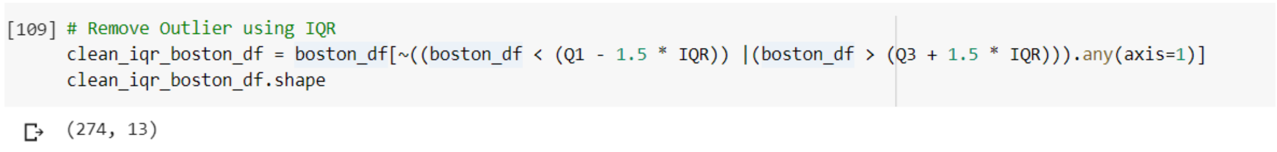
	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...
501	False	False	False	False	False	False	False	False	False	False	False	False	False
502	False	False	False	False	False	False	False	False	False	False	False	False	False
503	False	False	False	False	False	False	False	False	False	False	False	False	False
504	False	False	False	False	False	False	False	False	False	False	False	False	False
505	False	False	False	False	False	False	False	False	False	False	False	False	False

506 rows × 13 columns

Once we know that which records are Outlier, we can easily remove them from our dataset if required.

So, after removing the Outlier our dataset will have 274 rows. It means we have removed almost (506-274) 232 records from the dataset.

To remove the outlier, I am using the negate (~) in the code from above which we used to see the outlier.



```
[109] # Remove Outlier using IQR
      clean_iqr_boston_df = boston_df[~((boston_df < (Q1 - 1.5 * IQR)) |(boston_df > (Q3 + 1.5 * IQR))).any(axis=1)]
      clean_iqr_boston_df.shape

(274, 13)
```

Note: With the Z-score we were able to drop 91 rows but with IQR we have dropped 232 rows.

Also, different outlier treatments affect models differently.

Python Code is available at:
https://colab.research.google.com/drive/1ISWQ_muJGqvciMN_f8W37MZTgBfIazat?usp=sharin

g

References:

- https://www.anodot.com/
- https://www.wikipedia.org/
- https://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/
- https://www.kdnuggets.com/2017/01/3-methods-deal-outliers.html

About the Data Science Foundation

The Data Science Foundation is a professional body representing the interests of the Data Science Industry. Its membership consists of suppliers who offer a range of big data analytical and technical services and companies and individuals with an interest in the commercial advantages that can be gained from big data. The organisation aims to raise the profile of this developing industry, to educate people about the benefits of knowledge based decision making and to encourage firms to start using big data techniques.

Contact Data Science Foundation

Email:admin@datascience.foundation
Telephone: 0161 926 3641
Atlantic Business Centre
Atlantic Street
Altrincham
WA14 5NQ
web: www.datascience.foundation