**Cell Traffic Anomaly Detection**

**Use Case Description**

* + - Many a times cell traffic volumes are indicative of a situation that needs operations team to intervene to identify corrective/ preventive actions.

**Root Cause**

* + - Positive Deviation - A cell detected as « over subscribed ». Traffic level very high to that learned normal i.e. being multiple standard deviations above from its average. Indicating neighbor cell failing partly/fully or unknown event.
    - Negative Deviation - A cell detected as « under used». Traffic level very low to that of learned normal i.e. being multiple standard deviations below its average. Indicating a cell failed or failing partly.

**Business Value**

* + - The main objective of the use-case is to look for anomalous behaviors in cell traffic volumes and bring it to the notice of network managers for further investigations.

**Recommendations**

* + - Trigger problem management case to analyze and find root cause of detected anomaly

**Actuation**

* + - Ticket Creation

Anomaly detection is a process for identifying unexpected data, event or behavior that require some examination.There is a large number of algorithms to detect anomalies in a dataset depending on data type and business context. Z-score is probably the simplest algorithm that can rapidly screen candidates for further examination to determine whether they are suspicious or not.

**What is Z-score**

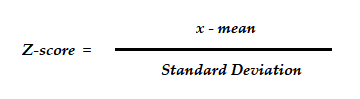
Simply speaking, Z-score is a statistical measure that tells you how far is a data point from the rest of the dataset. In a more technical term, Z-score tells how many standard deviations away a given observation is from the mean.

For example, a Z score of 2.5 means that the data point is 2.5 standard deviation far from the mean. And since it is far from the center, it’s flagged as an outlier/anomaly.

**How it works?**

Z-score is a parametric measure and it takes two parameters — mean and standard deviation.

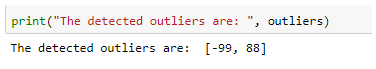
Once you calculate these two parameters, finding the Z-score of a data point is easy.



Note that mean and standard deviation are calculated for the whole dataset, whereas *x* represents every single data point. That means, every data point will have its own z-score, whereas mean/standard deviation remains the same everywhere.

**Example-**

# import numpy  
import numpy as np# random data points to calculate z-score  
data = [5, 5, 5, -99, 5, 5, 5, 5, 5, 5, 88, 5, 5, 5]# calculate mean  
mean = np.mean(data) # calculate standard deviation  
sd = np.std(data)# determine a threhold  
threshold = 2# create empty list to store outliers  
outliers = []# detect outlier  
for i in data:   
 z = (i-mean)/sd # calculate z-score  
 if abs(z) > threshold: # identify outliers  
 outliers.append(i) # add to the empty list# print outliers   
print("The detected outliers are: ", outliers)

**­­­** 

**Caution and conclusion**

* There are 14 data points and Z-score correctly detected 2 outliers [-99 and 88]. However, if you remove five data points from the list it detects only 1 outlier [-99]. That means you need to have a certain number of data size for Z-score to work.
* In large production datasets, Z-score works best if data are normally distributed (aka. Gaussian distribution).
* I used an arbitrary threshold of 2, beyond which all data points are flagged as outliers. The rule of thumb is to use 2, 2.5, 3 or 3.5 as threshold.
* Finally, Z-score is sensitive to extreme values, because the mean itself is sensitive to extreme values.