Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

**Department of Computer Science and Engineering (Data Science)**

# Lab Manual

**Subject: Foundations of Data Analysis Laboratory (DJ19DSL303)**

**Semester: III**

## Experiment 8

## (Outlier Detection)

**Aim:** Perform different outlier detection methods on given data.

**Theory:**

Anomaly detection (aka outlier analysis) is a step in data mining that identifies data points, events, and/or observations that deviate from a dataset's normal behavior. Anomalous data can indicate critical incidents, such as a technical glitch, or potential opportunities, for instance a change in consumer behavior.

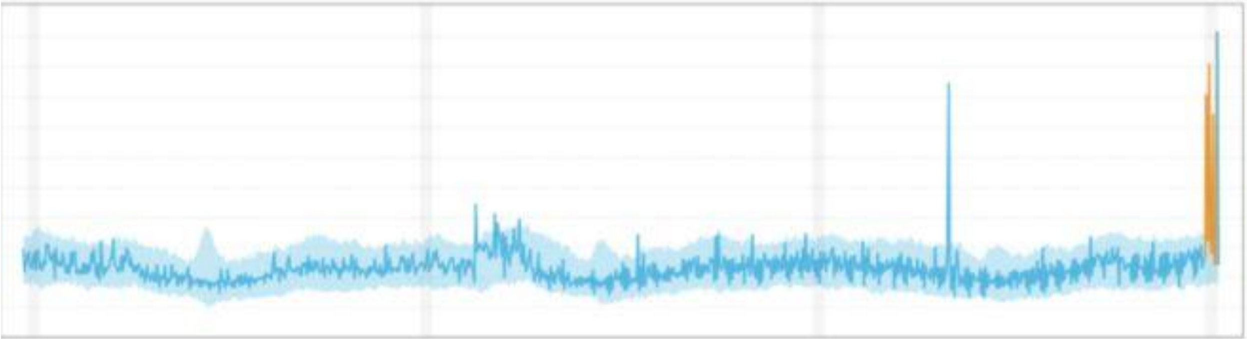
With all the analytics programs and various management software available, it's now easier than ever for companies to effectively measure every single aspect of business activity. This includes the operational performance of applications and infrastructure components as well as key performance indicators (KPIs) that evaluate the success of the organization. With millions of metrics that can be measured, companies tend to end up with quite an impressive dataset to explore the performance of their business.

Within this dataset are data patterns that represent business as usual. An unexpected change within these data patterns, or an event that does not conform to the expected data pattern, is considered an anomaly. In other words, an anomaly is a deviation from business as usual.

Generally speaking, anomalies in your business data fall into three main categories — global outliers, contextual outliers, and collective outliers.

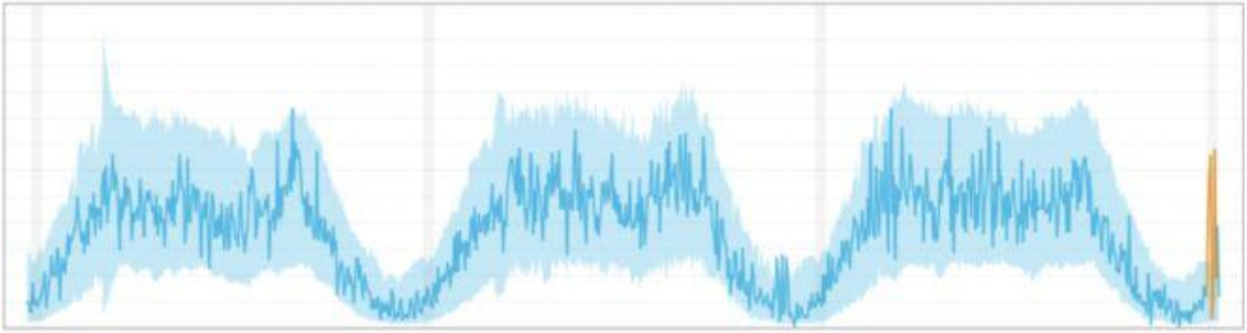
### 1. Global outliers

Also known as point anomalies, these outliers exist far outside the entirety of a data set.



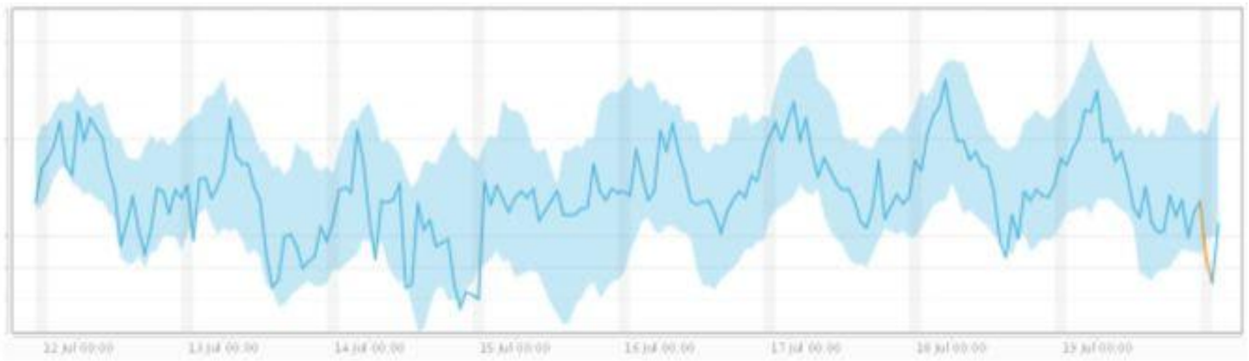
### 2. Contextual outliers

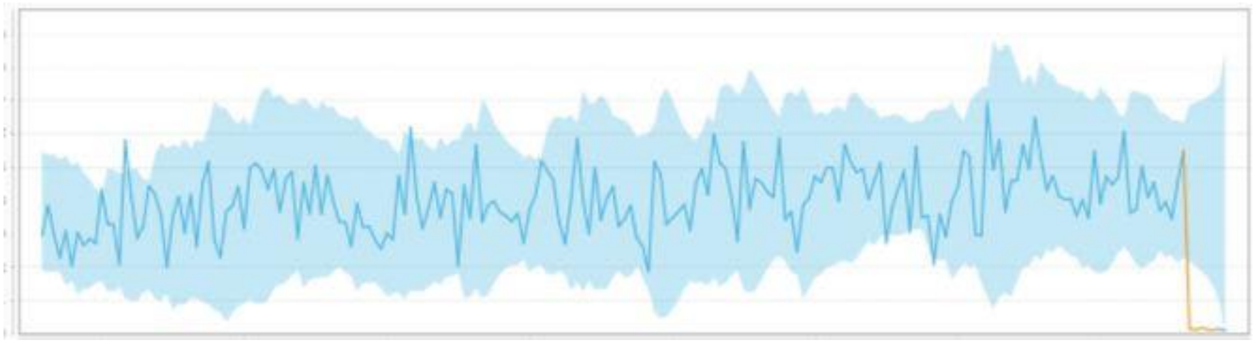
Also called conditional outliers, these anomalies have values that significantly deviate from the other data points that exist in the same context. An anomaly in the context of one dataset may not be an anomaly in another. These outliers are common in time series data because those datasets are records of specific quantities in a given period. The value exists within global expectations but may appear anomalous within certain seasonal data patterns.



### 3. Collective outliers

When a subset of data points within a set is anomalous to the entire dataset, those values are called collective outliers. In this category, individual values aren't anomalous globally or contextually. You start to see these types of outliers when examining distinct time series together. Individual behavior may not deviate from the normal range in a specific time series dataset. But when combined with another time series dataset, more significant anomalies become clear.





**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.

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

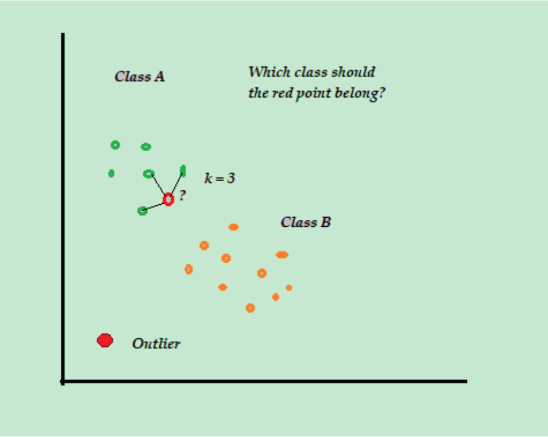
Once you calculate these two parameters, finding the Z-score of a data point is easy.

$$\text{Z-score} = \frac{x - \text{mean}}{\text{Standard Deviation}}$$

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.

### kNN for anomaly detection

Although kNN is a supervised ML algorithm, when it comes to anomaly detection it takes an unsupervised approach. This is because there is no actual "learning" involved in the process and there is no pre-determined labeling of "outlier" or "not-outlier" in the dataset, instead, it is entirely based upon threshold values. Data scientists arbitrarily decide the cutoff values beyond which all observations are called anomalies (as we will see later). That is also why there is no train-test-split of data or an accuracy report.



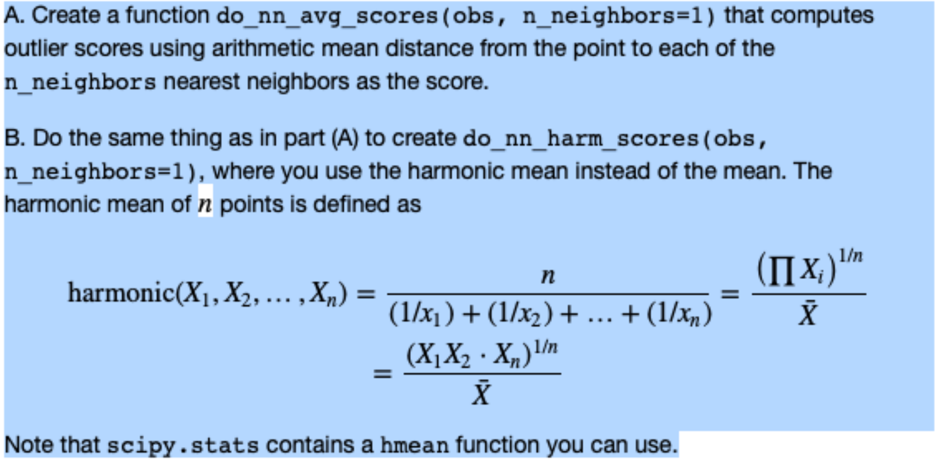
### Local Outlier Factor (LOF)

LOF is an unsupervised (well, semi-supervised) machine learning algorithm that uses the density of data points in the distribution as a key factor to detect outliers.
LOF compares the density of any given data point to the density of its neighbors. Since outliers come from low-density areas, the ratio will be higher for anomalous data points. As a rule of thumb, a normal data point has a LOF between 1 and 1.5 whereas anomalous observations will have much higher LOF. The higher the LOF the more likely it is an outlier. If the LOF of point X is 5, it means the average density of X's neighbors is 5 times higher than its local density.
In mathematical terms,

LOF(X)=[(LRD(1st neighbor) + LRD(2nd neighbor ) + ................+ LRD(kth neighbor))/LRD(X)]/k
where LRD is Local Reachability Distance and is computed as follows.
LRD(X) = 1/(sum of Reachability Distance (X, n))/k)where n is neighbors upto k

## Lab Assignments to complete in this session

1. For SAT dataset show the outliers using z-score and modifier z-score.
2. For Football data show the outliers using z-score and modifier z-score.
3. Generate 8 random points and perform KNN and LOF for K=1 and K=3. Write your observations.
4.



A. Create a function `do_nn_avg_scores(obs, n_neighbors=1)` that computes outlier scores using arithmetic mean distance from the point to each of the `n_neighbors` nearest neighbors as the score.

B. Do the same thing as in part (A) to create `do_nn_harm_scores(obs, n_neighbors=1)`, where you use the harmonic mean instead of the mean. The harmonic mean of $n$ points is defined as

$$\text{harmonic}(X_1, X_2, \ldots, X_n) = \frac{n}{(1/x_1) + (1/x_2) + \ldots + (1/x_n)} = \frac{\left(\prod X_i\right)^{1/n}}{\bar{X}}$$
$$= \frac{(X_1 X_2 \cdot X_n)^{1/n}}{\bar{X}}$$

Note that `scipy.stats` contains a `hmean` function you can use.

5. This exercise refers to density-based methods (Local Outlier Factor; Section 3).
   A. Create a function do_lof_outlier_scores(obs, n_neighbors=3) that computes outlier scores using the LOF method. Recall that the values returned by sklearn's implementation are negatives of what we want.