Credit Suisse Hackathon :
Team members : Subhasish Basak, Purnendu Ghosh, Trina De
Institute : Chennai Mathematical Institute
Topic: Outlier/Anomaly Detection

All the team members collaboratively worked on a git repository which consists of all the used resources along with the python ipynb jupyter notebooks. It can be accessed through the following link :
https://github.com/Subhasishbasak/Anomaly_detection.git

Workflow :

We have applied a few methods to find out outliers in the dataset. The first thing we did was to identify and estimate the **missing values** in the dataset. The method used for this was **Linear Interpolation**, using dataframe.interpolation() package available in Pandas library.

LOF (Local Outlier Factor) :

The local outlier factor is based on a concept of a local density, where locality is given by k nearest neighbors, whose distance is used to estimate the density. We fix a **k** and find the corresponding radius **r** such that the neighbourhood of a point p has at least **k** points. Corresponding to each point p we find the smallest **r** called the **local radius** of p. Then the LOF factor of a point is given by,

LOF(p) = local_radius(p) / average local_radius

This has a disadvantage because the value of this being less than 1 is not a clear indication of an outlier. We thus apply a second method where instead of considering only the k nearest neighbours, we consider the density or sparsity of the entire dataset as a whole.

Isolation Forest :

This applies too, an algorithm based on the idea that an outlier point will be located in a region that is significantly different from a normal point. It exploits the idea of decision trees. It picks a feature at random and them chooses a random partition between the maximum and minimum values of the feature. It is believed that an outlier point would be identified at a shorter path length than usual. In this algorithm a score is assigned to each point based on the path length that is needed to identify it. This score is based on the path length to identify the point and the usual path length to categorize points in binary search.

Density Based approach :

In this method, we try to consider the variation of all points of a feature(i.e. A column) instead of considering only k nearest neighbours of it.

We take one column at a time. We then calculate the variation of points in that column. We have a function that computes the variation of values in that column omitting one value at a time. We then take the difference of each such value from the total variation of that column. Any values outside the **3-sigma neighbourhood of the mean of such difference** is believed to be an outlier and associated with the appropriate marker. The assumption is that something that causes more than usual reduction in the variation of the column is an outlier. And that the difference of the total variation with the variation leaving out a single value follows a normal distribution. We make this guess for the lack of a better more informed alternative.

Quantile method:

This is a nonparametric outlier detection method in a one dimensional feature space based on quantiles. Here outliers are calculated by means of the **IQR (InterQuartile Range)**.The first and the third quartile are $\left(Q_1, Q_3 \right)$ calculated.An outlier is then a data point x that lies outside the interquartile range.

$$X_i > Q_3 + k \times IQR \ or \ X_i < Q_1 - k \times IQR$$
$$where, \ IQR = Q_3 - Q_1$$

Using the interquartile multiplier value k=1.5, the range limits are the typical upper and lower whiskers of a box plot.

Empirical CDF method :

All the above discussed method gives binary output,i.e. In other words it classifies each observation to either an outlier or not. In this method we estimate the **empirical cdf (F(t))** from different asset variable. We estimate the probability of an observation to be an outlier as :

p(x is an outlier) = min (F(x), 1-F(x)) / 0.5

The minimum takes into account both right & left tail of the empirical distribution and estimates only the extreme tail probability. The 0.5 normalization factor is introduced to scale the value in [0,1] interval so that it can be interpreted as probability.

Combining the results :

The probabilities are converted to boolean using 0.5 as threshold and the classifiers are combined with appropriate weights. The weights are calculated based on the performance of the methods on the **checked** data.