I think your document should be as follows:

Remember, you have a product (academic or commertial) and you want to document it. Therefore, the main goal of this document is to describe YOUR abstraction layer.

Start with the introducing and motivating your product. What is the goal of this abstraction layer? Why do we need it? What is the input of it? What is the output? And so on.

Then, when for example you are explaining the input configuration methods, you can explain dboost with just a short description by your own. For example:

Dboost is an outlier detection system [refrence...]. It provides four different methods. Our abstraction layer supports these methods as follows...

And then you can describe how the user can use the abstraction layer to run different dboost methods. Always remember that the user, just want to know your module and its capabilities. If she needs to know more about dboost, she will read the paper.

**DBOOST:**

*a new approach for detecting outliers in highly heterogeneous datasets. Our tool systematically expands the limited space of SQL types to derive richer Information.*

*In the paper of the tool, first mentioned tuple expansion procedure which reconstructs rich information from semantically poor SQL data types such as strings, integers, and floating point numbers. This tool is suitable for traditional numerical datasets and in highly non-numerical contexts such as mostly textual datasets.*

*The expanded tuples are then used to train one of three data models (Histograms,*

*Gaussian, or Mixtures), with the help of the statistics and correlation hints gathered at the previous stage.[1]*

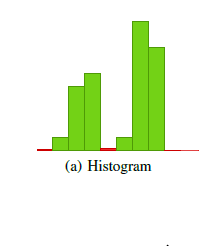
**Histogram:**

*histograms provide a way to study discrete heterogeneous distributions. Although histograms are widely used to capture statistics in databases, we present a simple pruning heuristic to limit the number and size of histograms that we construct for outlier detection purposes.*

*The histogram model (Figure) does not make any assumption about the data under study. Instead, it counts the occurrences of each unique value in each column of the expanded tuple and for each set of potentially correlated sub-tuples (as suggested by the analysis module). These counts, accumulated over the entire dataset, provide a de facto distribution of the data in each field and set of correlated fields. This makes histograms a powerful model for on-numerical and heterogeneous data.*

*Histograms also have the valuable property of treating sets of fields (obtained via correlation analysis) and single fields in the exact same way, thus permitting to model single columns*

*or groups of attributes indifferently.[1]*



***Distribution-independent*** *– Given a histogram with N bins, we count only the values in the top bin if 1  N  3, in the top 2 bins if 4  N  5, and in the top 3 bins for 3  N  16 (histograms with N > 16*

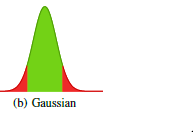
*bins were previously discarded). This method is stable when the set of bins is static (week days, booleans, . . . ), but it is sensitive to the addition of removal of bins.*

***Distribution-dependent*** *– We sort the bins in increasing order of bin size bi, and find the index imax such that the ratio r = bi+1=bi is maximal (this calculation is safe, because the bin sizes are non-zero integers). If that ratio is under a user-defined threshold, we reject the histogram; otherwise, we consider bins imax::end to be “top” bins.[1]*

**Gaussian:**

*Univariate Gaussians (Figure) are a widely used statistical model for data. They treat each value xi of the expanded tuples as random sample drawn from a normal distribution N(i; i).The model’s parameters (a pair (,) for each numerical column) are computed as each column’s mean and standard deviation. In the common case where the dataset has not significantly changed between the analysis and the modeling passes, the information obtained during the statistical analysis pass is sufficient to derive these parameters.*

*Despite its simplicity, this model presents the attractive property of requiring extremely little memory – on the order of the size of one expanded tuple.[1]*



*The simple Gaussian model measures how much each value differs from the mean computed*

*in the preceding pass. Given a tolerance parameter , a row is deemed an outlier if at least one of its attributes a has a value va such that [ ]where a and a are the model’s parameters for column a[1].*

**Mixture** :

*Multivariate Gaussian Mixture models (Figure) are another standard statistical model. They take advantage of the correlation hints supplied by the statistical analysis pass to model sub-tuples of the expanded tuples as samples from multivariate Gaussian mixtures (GMMs), creating one model per group of correlated columns.*

*In addition, when dealing with large amounts of data, it is possible – and indeed, preferable – to train the Mixture model on a randomly sampled subset of the data before running the full analysis. This approach is particularly relevant when using the Mixture model, but can be applied to all models to shorten the learning phase when dealing with very large datasets [1].*



*In the Mixture model, the likelihood of each (possibly multidimensional) field is evaluated using the corresponding GMM˙ This model operates under the assumption that data is accurately modeled by the chosen number of components in the GMM, and in particular that each non-outlying data point is well modeled by one of the Gaussians of the GMM [1].*

Reference:

[1] Mariet, Zelda, Rachael Harding, and Sam Madden. "Outlier Detection in Heterogeneous Datasets using Automatic Tuple Expansion." (2016).

**First step**: for using the abstract layer you should import it in the top of your script

**Second step**: make your string for passing to the function

Regarding to our structure you should first choose the **name of tool** that you wish to use and then **address of directory** that your data file is located then your **method** comes as following and for each method, it needs certain **argument** although you need to specify another **method** and their **argument**, it would the last step for making the string that you are needed for running the function.

--statistical arg1

--discretestats arg1,arg2

--cords arg1,arg2

run\_input=[**"Dboost"**,**"Address\11.csv"**,**"method"**,**"arg"**,**"--Method"**,**"arg"**,**"-F ,"**]

--gaussian arg1 separator

--histogram arg1,arg2

--mixture arg1,arg2

--partitionedhistogram arg1,arg2,arg3

directory=**'E:\work\DFKI\Tools\Dboost\dBoost-master'**

**you should change the directory to the place that you put the dboost file**

For the complimentary the help of the tool is printed in the following pages.

