



Abstract layer for data cleaning tools

By:

Milad Abbaszadeh Jahromi

Supervision:

Prof. Dr. rer. nat. Ziawash Abedjan
Eng. Mohammad Mahdavi

In the:

Institute for Software Engineering and Theoretical Computer Science
Department Big Data Management

October 2017

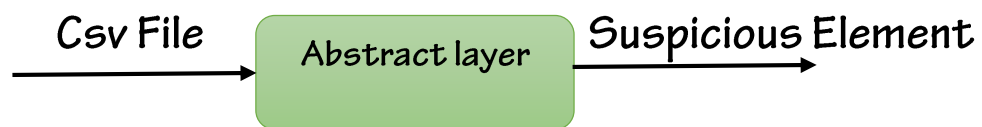
What's going on?

This report talks about the abstract layer which can help users to avoid from conflicting with the tools for cleaning purpose and help them for unique and constant input and output which can be really remarkable to save time for people who wish to work with these tools.

What we do:

If you have a file and wish to clean it with cleaning tools and you wouldn't like to get involved in installation and command usage which is required for running, you can simply use our abstract layer.

With this layer you can simply send your file and then get the Suspicious element by the constant type and form in output for each tool.



What do you need to have for start:

We kindly recommend you to set Linux as your operation system for doing the scientific purpose although this layer perfectly work in windows.

You should install:

- Python 2.7
- [Java 1.8](#) (Please make sure that you install Java from oracle)
- [Apache Ant 1.8.2+](#)
- [Postgres SQL 9.2+](#)

What you need to do:

For starting with abstract layer you should make sure that you make dictionary like dictionary that is introduced below as input for our layer.

This dictionary has two keys that you can set the first one with the type of your file and the location of the file as path. In fact, for now the layer only support the files that generate with csv format but we will keep going to add other formats for future.

```
run_input = {  
  "dataset": {  
    "type": "csv",  
    "path": "Put_address_of_your_file.csv"  
  },  
  "tool": {  
    "name": "dboost or nadeef",  
    "param": ["your parameters come here"]  
  }  
}
```

One example for dboost :


```
run_input = {  
  "dataset": {  
    "type": "csv",  
    "path": "datasets/sample.csv"  
  },  
  "tool": {  
    "name": "dboost",  
    "param": ["--gaussian", "1", "--statistical", "1"]  
  }  
}
```

Note: If you like to use other methods of dboost for cleaning purpose and they need more parameter like histogram or even mixture you can easily make your param in your dictionary with more elements as mentioned below:

```
"param": ["--gaussian", "1", "--discretstats", "1", "2"]
```

You can find more information for parameters in the table

Method	Argument
--gaussian	One
--histogram	Two
--mixture	Two
--Partitionedhistogram	Three



Second method	Argument
--statistical	One
--discretstats	Two
--cords	Two

One example for Nadeef:

```
run_input = {  
  "dataset": {  
    "type": "csv",  
    "path": "/home/milad/Desktop/nadeef/dataset_sample.csv"  
  },  
  "tool": {  
    "name": "nadeef",  
    "param": [  
      {  
        "type": "fd",  
        "value": ["first_author | language"]  
      }  
    ]  
  }  
}
```

Note: make sure that you generate database in your PostgreSQL with name of "nadeef" and owner of "tester" and password should be "tester" as well.

What you get:

You will get one list as output that you can find it below

[row, column, suspicions value]