inspirata
Inspired Data Solutions

HDR UK: Data Quality Tool Assessment Request

Overview

As part of the work in HDR-UK Data Utility Workstream, we are testing several open-source/free tools to profile data quality. This will allow us to determine the impact of activities to improve the data quality.

We would sincerely like to thank you for your willingness to evaluate/run specified tools in order to provide the outputs and any feedback to HDR-UK and Inspirata.

This document provides a high-level overview and product information for the two data profiling tools you are being requested to evaluate/run against real-world data.

Each tool should be tested against your identified dataset(s) and evaluated in terms of ability to achieve the below mentioned criteria 1-3.

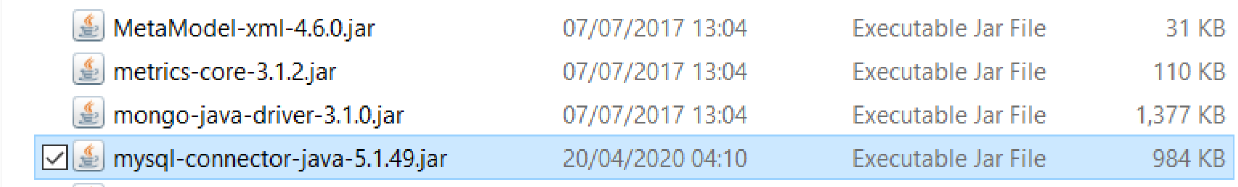
In addition, if possible, we also request that you evaluate and provide feedback on a Data Quality Monitoring/Testing tool called MobyDQ – Criteria 1 **only**. MobyDQ is a tool for data engineering teams to automate data quality checks on their data pipeline, capture data quality issues and trigger alerts in case of anomaly.

Criteria 1: Please provide any feedback on your experience of the tool

Please provide any feedback, screenshots and comments that you believe may be pertinent.

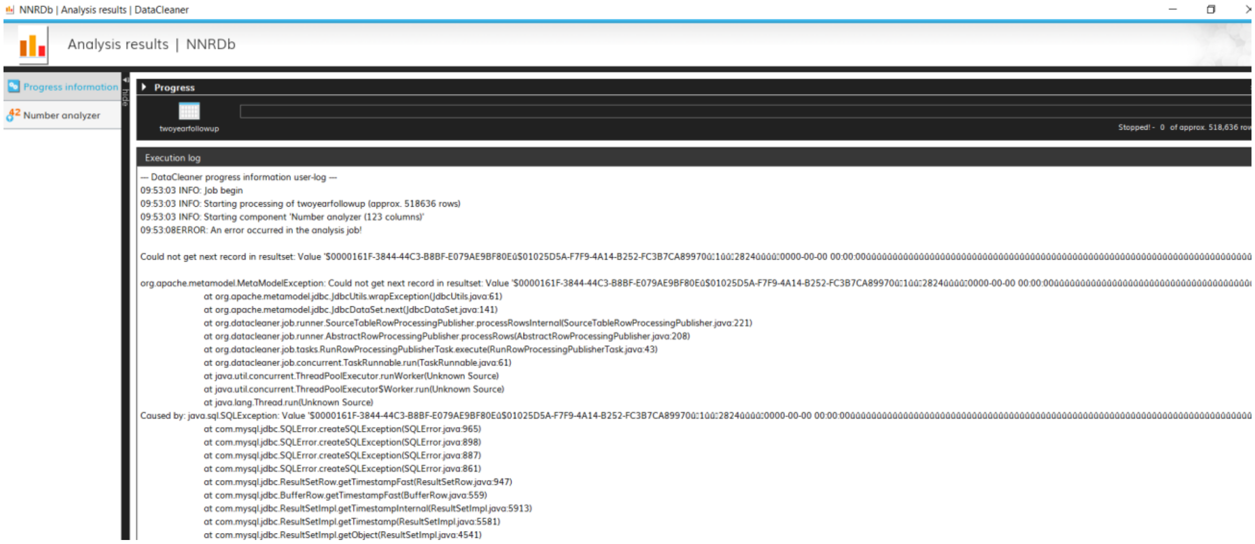
Experience with DataCleaner:

DataCleaner was easier to setup, even on a windows machine. Was possible to link to a MySQL database. Was possible to run analysis on more than one table at once.
I had to change the MySQL connector to a more recent version to match support for a MySQL 5.7 database.



MetaModel-xml-4.6.0.jar	07/07/2017 13:04	Executable Jar File	31 KB
metrics-core-3.1.2.jar	07/07/2017 13:04	Executable Jar File	110 KB
mongo-java-driver-3.1.0.jar	07/07/2017 13:04	Executable Jar File	1,377 KB
☑ mysql-connector-java-5.1.49.jar	20/04/2020 04:10	Executable Jar File	984 KB

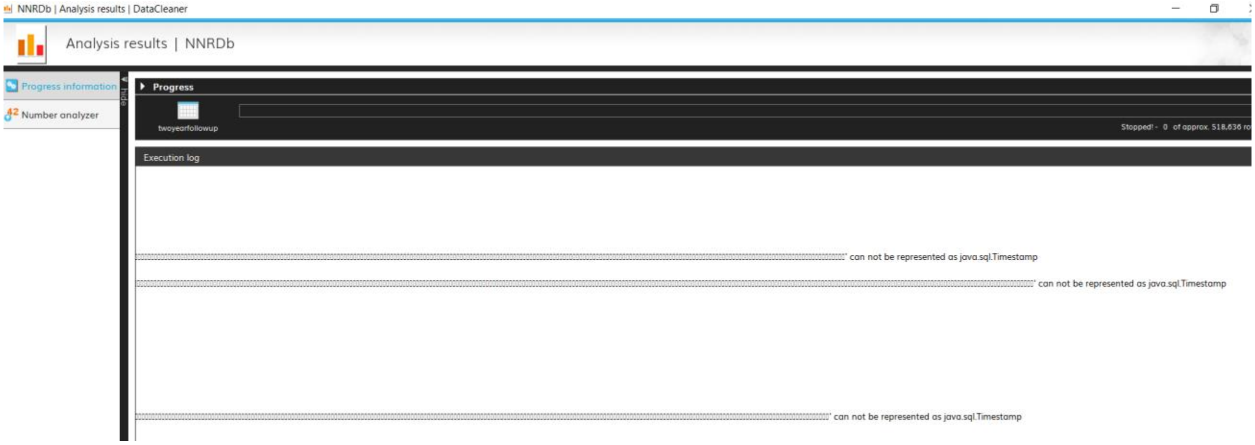
To access timeliness on criteria 3, a table with datetime fields was chosen, and due to a default value of 0000-00-00, the following error presented:



```
--- DataCleaner progress information user-log ---
09:53:03 INFO: Job begin
09:53:03 INFO: Starting processing of twoyearfollowup (approx. 518636 rows)
09:53:03 INFO: Starting component 'Number analyzer (123 columns)'
09:53:08ERROR: An error occurred in the analysis job!

Could not get next record in resultset: Value '$0000161F-3844-44C3-B8BF-E079AE9BF80E0$01025D5A-F7F9-4A14-B252-FC3B7CA8997001002824000000000-00-00 00:00:000000000000000000000000000000000000000000000000000000000000000000000000

org.apache.metamodel.MetaModelException: Could not get next record in resultset: Value '$0000161F-3844-44C3-B8BF-E079AE9BF80E0$01025D5A-F7F9-4A14-B252-FC3B7CA8997001002824000000000-00-00 00:00:0000000000000000000000000000000000000000000
            at org.apache.metamodel.jdbc.JdbcUtils.wrapException(JdbcUtils.java:61)
            at org.apache.metamodel.jdbc.JdbcDataSet.next(JdbcDataSet.java:141)
            at org.datacleaner.job.runner.SourceTableRowProcessingPublisher.processRowsInternal(SourceTableRowProcessingPublisher.java:221)
            at org.datacleaner.job.runner.AbstractRowProcessingPublisher.processRows(AbstractRowProcessingPublisher.java:208)
            at org.datacleaner.job.tasks.RunRowProcessingPublisherTask.execute(RunRowProcessingPublisherTask.java:43)
            at org.datacleaner.job.concurrent.TaskRunnable.run(TaskRunnable.java:61)
            at java.util.concurrent.ThreadPoolExecutor.runWorker(Unknown Source)
            at java.util.concurrent.ThreadPoolExecutor$Worker.run(Unknown Source)
            at java.lang.Thread.run(Unknown Source)
Caused by: java.sql.SQLException: Value '$0000161F-3844-44C3-B8BF-E079AE9BF80E0$01025D5A-F7F9-4A14-B252-FC3B7CA8997001002824000000000-00-00 00:00:000000000000000000000000000000000000000000000000000000000000000000000000000
            at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:965)
            at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:898)
            at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:887)
            at com.mysql.jdbc.SQLError.createSQLException(SQLError.java:861)
            at com.mysql.jdbc.ResultSetRow.getTimestampFast(ResultSetRow.java:947)
            at com.mysql.jdbc.BufferRow.getTimestampFast(BufferRow.java:559)
            at com.mysql.jdbc.ResultSetImpl.getTimestampInternal(ResultSetImpl.java:5913)
            at com.mysql.jdbc.ResultSetImpl.getTimestamp(ResultSetImpl.java:5581)
            at com.mysql.jdbc.ResultSetImpl.getObject(ResultSetImpl.java:4541)
```



```
0000000000000000000000000000000000000000000000000000000000000000000000000000000000' can not be represented as java.sql.Timestamp

000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000' can not be represented as java.sql.Timestamp




0000000000000000000000000000000000000000000000000000000000000000000000000000000' can not be represented as java.sql.Timestamp
```

Which was resolved by adding a *zeroDateTimeBehavior=convertToNull* property to the connection string. i.e. jdbc:mysql://localhost:3306/NNRDb?defaultFetchSize=-2147483648&largeRowSizeThreshold=1024&zero-DateTimeBehavior=convertToNull

Experience with Orange:

Orange could not directly interact with the MySQL database, data migration was done, first to MS SQL Server. Since data was too big to download to local memory on an SQL Server instance on SQL Table widget, we later migrated to postgres SQL as it had support to uncheck the "download to local memory" option. However to adequately interact with the Postgres database, an installation of two modules: quantile and sm_system_time was required. Installing these two on a windows machine is not fully supported documentation wise, when compared to Mac OS and Linux. We failed to setup a connection of the pgxn client to both standalone orange or the anaconda instance. I believe this exercise would have been best carried via Linux machine. We could set up a virtual machine if we are given additional time for this. If need be.

With these challenges, we still connected the NNRD but through the MS SQL instance using pymssql.

Experience with Mobydq

Setup of MobyDQ on the windows platform was via the Docker Desktop for Windows. Though not successfully to fully run the app on our NNRD table, the two year follow up table, as with Orange and Data Cleaner, We explored it's setup via

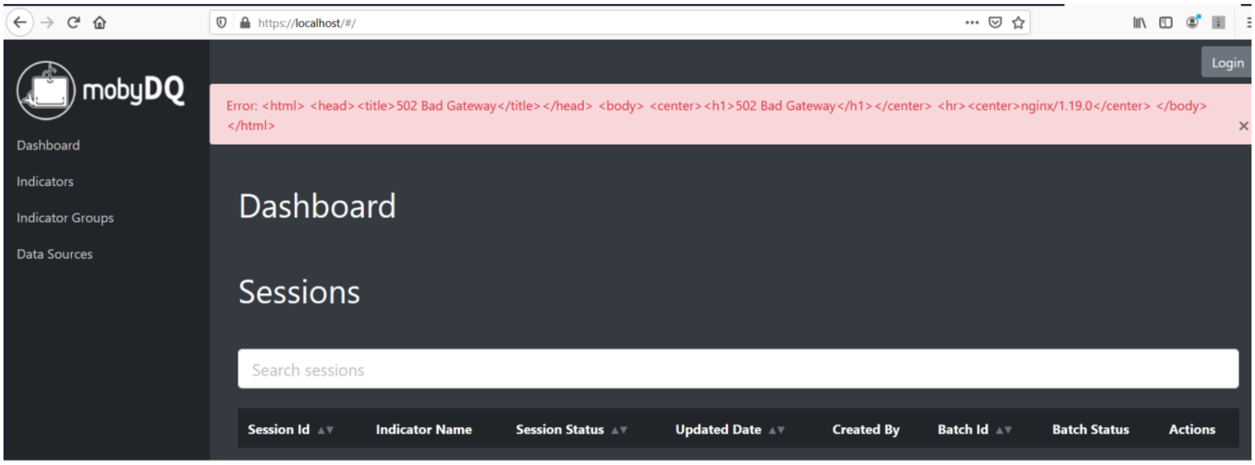
1. A pre-exisiting Postgres server:
   Creating the .env with database credentials of our pre existing postgres server encountered an error that seemed to be associated with private keys:

```
mobydq-graphql | ⚠ WARNING⚠  You requested to use schema 'base'; however we couldn't find
some of those! Missing schemas are: 'base'
mobydq-graphql | A serious error occurred when building the initial schema. Exiting because
`retryOnInitFail` is not set. Error details:
mobydq-graphql |
mobydq-graphql | Error: Could not find JWT type '"base"."token"'
mobydq-graphql |     at PgJWTPlugin/init/PgJWT (/home/node/app/node_modules/graphile-build-
pg/node8plus/plugins/PgJWTPlugin.js:47:13)
mobydq-graphql |      at SchemaBuilder.applyHooks (/home/node/app/node_modules/graphile-
build/node8plus/SchemaBuilder.js:252:20)
mobydq-graphql |      at SchemaBuilder.createBuild (/home/node/app/node_modules/graphile-
build/node8plus/SchemaBuilder.js:313:10)
mobydq-graphql |      at SchemaBuilder.buildSchema (/home/node/app/node_modules/graphile-
build/node8plus/SchemaBuilder.js:321:26)
mobydq-graphql |     at Object.exports.createPostGraphileSchema (/home/node/app/node_mod-
ules/postgraphile-core/node8plus/index.js:225:28)
mobydq-graphql |     at processTicksAndRejections (internal/process/task_queues.js:97:5)
mobydq-graphql |         at async createGqlSchema (/home/node/app/node_modules/post-
graphile/build/postgraphile/postgraphile.js:77:33)
```

   We explored with recreating the cert.pem and key.pem files in C:\mobydq\nginx\config, to no avail.
   In the .env file, we also changed the POSTGRES_DB variable from mobydq to our custom database, but still more, the error showed up.
   Finally, I ran the 00-database.sql script independently in Postgres, just to successfully create the base schema, though it was able to create the schema, for some reason this mobdy docker image could not interface with our Postgres instance, and therefore the no base schema error still showed up.

2. A Postgres docker Image:
   With the above finding, it indicated as though our Postgres was not interacting with our mobydq docker image, and so we resorted to unistalling it and downloading and using a postgres docker image instead.
   This presented issues of not being able to setup or identify postgres database credentials (username and password). Uisng the default ones, that were mentioned on https://hub.docker.com/_/postgres/, the username was obvious, but the POSTGRES_PASSWORD was tricky to identify.
   The documentation stated that it is generated when the initdb script runs during initial container startup.
   The location of the docker-entrypoint-initdb.d script was not found and this made it impossible to identify the correct password to specify in the .env file.
   As a result we kept on getting a failed authentication for user postgres.

   All in all, whether we run the app in production or development mode, we would only get a blank mobydql dashboard, through which we could not set parameters and test on our NNRD table.



Error: <html> <head><title>502 Bad Gateway</title></head> <body> <center><h1>502 Bad Gateway</h1></center> <hr><center>nginx/1.19.0</center> </body> </html>

Dashboard

Sessions

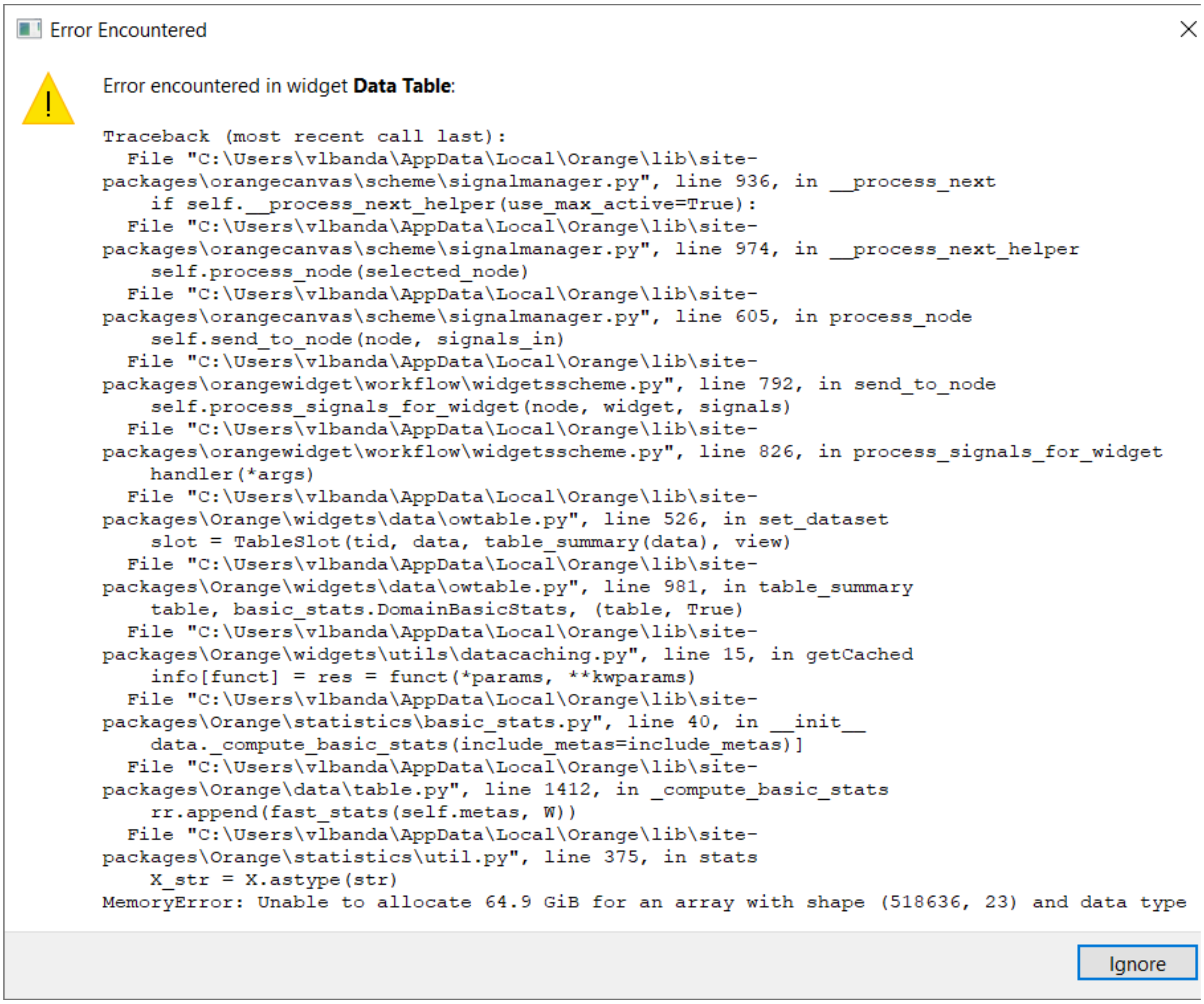
Session Id	Indicator Name	Session Status	Updated Date	Created By	Batch Id	Batch Status	Actions

The dataset:

Due to size of the data the NNRD has, and the type of profiling that was requested, we resolved to test on only a single table amongst the 12 data tables. The primary table was the obvious choice, with Neonatal Unit admissions of over 1,000,000. However due to the size of this table, Orange could not to the data, gave the "data too big to download to local memory" error.

We later on explored with a secondary table: AbdoXray, which had around 27,615 records but lacked date fields to help answer criteria 3 on timeliness.

The final choice was on a table with date fields, the TwoYearFollowUp table with a total of 518,636 records, which when loaded in Orange gave the "data too big to load into local memory" error.



Error encountered in widget **Data Table**:

```
Traceback (most recent call last):
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\orangecanvas\scheme\signalmanager.py", line 936, in __process_next
    if self.__process_next_helper(use_max_active=True):
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\orangecanvas\scheme\signalmanager.py", line 974, in __process_next_helper
    self.process_node(selected_node)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\orangecanvas\scheme\signalmanager.py", line 605, in process_node
    self.send_to_node(node, signals_in)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\orangewidget\workflow\widgetsscheme.py", line 792, in send_to_node
    self.process_signals_for_widget(node, widget, signals)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\orangewidget\workflow\widgetsscheme.py", line 826, in process_signals_for_widget
    handler(*args)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\widgets\data\owtable.py", line 526, in set_dataset
    slot = TableSlot(tid, data, table_summary(data), view)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\widgets\data\owtable.py", line 981, in table_summary
    table, basic_stats.DomainBasicStats, (table, True)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\widgets\utils\datacaching.py", line 15, in getCached
    info[funct] = res = funct(*params, **kwparams)
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\statistics\basic_stats.py", line 40, in __init__
    data._compute_basic_stats(include_metas=include_metas)]
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\data\table.py", line 1412, in _compute_basic_stats
    rr.append(fast_stats(self.metas, W))
  File "C:\Users\vlbanda\AppData\Local\Orange\lib\site-
packages\Orange\statistics\util.py", line 375, in stats
    X_str = X.astype(str)
MemoryError: Unable to allocate 64.9 GiB for an array with shape (518636, 23) and data type
```

I trimmed by only loading records with no missing AssessmentDate. This resulted in a dataset of 49,515 records, with which I ran Orange and DataCleaner on.

Criteria 2: Can the Tool profile/calculate the following?

Assessment values:

- Yes – Tool can do this
- No – Tool cannot do this
- Partial – Tool can partially do this

Category	#	Feature	Orange	Data-Cleaner
Single Column – Cardinalities	1	Number of rows	Yes	Yes
	2	Number of nulls	Yes	Yes
	3	Percentage of nulls	Yes	No

	4	Number of distinct values (cardinality)	No	Yes
	5	Percentage of distinct values (Number of distinct values divided by the number of rows)	No	No
Single Column - Value distributions	6	Frequency histograms (equi-width, equi-depth, etc.)	Yes	Yes
	7	Minimum and maximum values in a numeric column	Yes	Yes
	8	Constancy (Frequency of most frequent value divided by number of rows)	No	No
	9	Quartiles (3 points that divide the **numeric** values into 4 equal groups)	No	No
	10	Distribution of first digit in numeric values (to check Benford's law)	No	No
Single Column - Patterns, datatypes, and domains	11	Basic types (e.g. numeric, alphanumeric, date, time)	Yes	Yes
	12	DBMS-specific data type (e.g. varchar, timestamp)	No	Yes
	13	Measurement of Value length (minimum, maximum, average, median)	Yes	Yes
	14	Maximum number of digits in numeric values	No	No
	15	Maximum number of decimals in numeric values	No	No
	16	Histogram of value patterns (Aa9...)	Yes	Yes
	17	Generic semantic data type (e.g. code, date/time, quantity, identifier)	No	No
	18	Semantic domain (e.g. credit card, first name, city)	No	No
Dependencies	19	Unique column combinations (UCCs) (key discovery)	No	No
	20	Relaxed unique column combinations	No	No
	21	Inclusion dependencies (INDs) (foreign key discovery)	No	Yes
	22	Relaxed inclusion dependencies	No	No
	23	Functional dependencies	No	No
	24	Conditional functional dependencies	No	No
Advanced Multi Column profiling	25	Correlation analysis	Yes	No
	26	Association rule mining	Yes	No
	27	Cluster analysis	Yes	No
	28	Outlier detection	Yes	No
	29	Exact duplicate tuple detection	Yes	No
	30	Relaxed duplicate tuple detection	Yes	No

Criteria 3: How easily can the tool profile the data quality dimension measure criteria?

Ratings range from 0.0 to 5.0:

	0.0 = Tool cannot calculate this 1.0 = Poor: most or all defined requirements not achieved 2.0 = Fair: some requirements not achieved 3.0 = Good: meets requirements 4.0 = Excellent: meets or exceeds some requirements 5.0 = Outstanding: significantly exceeds requirements

DAMA Dimension	Measure (key elements)	Orange	Data-Cleaner
Completeness	Percentage of requisite information available	0.0	0.0
	Percentage of missing data values (null / empty string)	5.0	0.0
	Row counts	5.0	5.0
	Highest and lowest value of key elements	5.0	5.0
	Number of data values in an unusable state	0.0	0.0
Uniqueness	(Number of things in the real world) Number of incorrect spellings etc. of same data in an element e.g. address (duplicate values)		5.0
	(Number of recodes describing different things) Number of data items in adherence to expected/described data element value (distinct values at ID level)	0.0	5.0
	(Number of things in real world i.e. duplicates)/ (Number of records describing different things i.e. distinct records)	0.0	5.0
Timeliness	Difference between Lowest date value and Highest Date Value	0.0	2.0
	Number of records per month	0.0	2.5
Validity	Percentage of data values that comply with the specified formats (data types, ranges etc.)	0.0	0.0
	Percentage of data values that don't comply to specified formats	0.0	0.0
	Number of Missing values indicated e.g. with fill values	0.0	0.0
	Number of Values in Specified Range	0.0	1.0
	Number of values not in Specified Range	0.0	0.0
Accuracy	Number of accurate data values	0.0	5.0
	Number of inaccurate data values	0.0	5.0
	Actual data value count versus predicted data value count	0.0	0.0
	Number of rows and columns against expectations	0.0	5.0
	Number of duplicates at ID level	0.0	5.0
	Number of blank columns, large % of blank data, high % of same data	0.0	2.0
	Distribution across various segments	5.0	3.0
	Outliers on key variables	5.0	0.0
	((Count of accurate objects)/ (Count of accurate objects + Counts of inaccurate objects)	0.0	0.0

Consistency	Analysis of pattern and/or value frequency	3.0	0.0

Tools to Evaluate

1. Orange

Link(s):	• http://orange.biolab.si/ • https://orange.biolab.si/getting-started/ • https://orange-data-mining-library.readthedocs.io/en/latest/index.html • https://www.javatpoint.com/orange-data-mining
License:	GNU [GPL-3.0]+ license
Version:	3.24.1
OS:	Windows, macOS, Linux

Description:

Open source machine learning and data visualization for novice and expert. Interactive data analysis workflows with a large toolbox. Orange consists of a canvas interface onto which the user places widgets and creates a data analysis workflow. Widgets offer basic functionalities such as reading the data, showing a data table, selecting features, training predictors, comparing learning algorithms, visualizing data elements, etc. The user can interactively explore visualizations or feed the selected subset into other widgets.

Data mining is done through visual programming or Python scripting. Python scrips can run in a terminal window, integrated environments like PyCharm and PythonWin, or shells like iPython.

The tool has components for machine learning, add-ons for bioinformatics and text mining and it is packed with features for data analytics.

Orange consists of a canvas interface onto which the user places widgets and creates a data analysis workflow. Widgets off basic functionalities such as reading the data, showing a data table, selecting features, training predictors, comparing learning algorithms, visualizing data elements, etc. The user can interactively explore visualizations or feed the selected subset into other widgets.

In Orange, data analysis process can be designed through visual programming. Orange remembers the choices, suggests most frequently used combinations. Orange has features for different visualization, such as scatterplots, bar charts, trees, to dendorograms, networks and heatmaps.   There are over 100 widgets for standard data analysis and specialized add-ons for Bioorange for bioinformatics.

Core Orange supports Excel, comma- and tab-delimited files (.xlsx, .csv, .tab). It also reads online data, such as Google Spreadsheets. SQL widget supports PostgreSQL and MSSQL databases. Add-ons can load additional formats. For example, Orange3-ImageAnalytics add-on can import images (.jpg, .tiff, .png) and Orange3-Text add-on can import text files (.txt, .docx, .pdf).

PRODUCT FEATURE SHEET:

Canvas: graphical front-end for data analysis

Widgets:

- Data: widgets for data input, data filtering, sampling, imputation, feature manipulation and feature selection
- Visualize: widgets for common visualization (box plot, histograms, scatter plot) and multivariate visualization (mosaic display, sieve diagram).
- Classify: a set of supervised machine learning algorithms for classification
- Regression: a set of supervised machine learning algorithms for regression
- Evaluate: cross-validation, sampling-based procedures, reliability estimation and scoring of prediction methods
- Unsupervised: unsupervised learning algorithms for clustering (k-means, hierarchical clustering) and data projection techniques (multidimensional scaling, principal component analysis, correspondence analysis).

Add-ons:

- Associate: widgets for mining frequent item sets and association rule learning
- Bioinformatics: widgets for gene set analysis, enrichment, and access to pathway libraries
- Data fusion: widgets for fusing different data sets, collective matrix factorization, and exploration of latent factors
- Educational: widgets for teaching machine learning concepts, such as k-means clustering, polynomial regression, stochastic gradient descent
- Geo: widgets for working with geospatial data
- Image analytics: widgets for working with images and ImageNet embeddings
- Network: widgets for graph and network analysis
- Text mining: widgets for natural language processing and text mining
- Time series: widgets for time series analysis and modeling
- Spectroscopy: widgets for analyzing and visualization of (hyper)spectral datasets

2. DataCleaner

Link(s):	• http://datacleaner.org/docs

	• https://github.com/datacleaner/DataCleaner • https://travis-ci.org/datacleaner/DataCleaner
License:	GNU Lesser General Public License v3.0
Version:	5.7.0
OS:	Windows, macOS, Linux
Description:	

Community Edition is free, otherwise you need a subscription.

Open source data quality tool data profiling, data cleaning, and data integration. DataCleaner is a Data Quality toolkit that allows you to profile, correct and enrich your data. People use it for ad-hoc analysis, recurring cleansing as well as a swiss-army knife in matching and Master Data Management solutions.

DataCleaner is a strong data profiling engine for discovering and analyzing the quality of user's data. It is built to handle data both big and small from CSV files, Excel spreadsheets to RDBMs and NoSQL databases. User can build their own cleansing rules and compose them into several use scenarios or target databases whether it is simple search/replace rules, regular expressions, and pattern matching or completely custom transformations.

DataCleaner's Monitoring establishes the starting point and goals, and to ensure a process of following up on data quality issues. The monitoring server of DataCleaner enables users to make point-in-time profiles of users' data, and to schedule periodic data quality checks and receive notifications if quality KPIs get out of control.

DataCleaner's Data Quality Eco-System delivers out-of-the-box functionality, and application extensions, integrations and shared content.

DataCleaner's Duplicate detection feature, builds on Machine Learning principles for ease of configuration and improved inferential matching

3. MobyDQ

Link(s):	• https://github.com/ubisoftinc/mobydq • https://ubisoftinc.github.io/mobydq/ • https://ubisoftinc.github.io/mobydq/pages/productiondeployment/
License:	Apache License 2.0
Version:	1.0
OS:	Windows, Linux
Description:	

MobyDQ is a tool for data engineering teams to automate data quality checks on their data pipeline, capture data quality issues and trigger alerts in case of anomaly, regardless of the data sources they use.

Free and open-source Data Quality solution that aims to automate Data Quality checks during data processing, storing Data Quality measurements and metric results, and triggering alerts in case of anomaly. The framework can be used to access different data sources. It installed quickly and straightforward, based on the detailed documentation provided on GitHub. MobyDQ does not provide any data profiling functionality, because its focus is on the creation, application and automation of Data Quality checks.

MobyDQ provides a toolbox for data engineering teams to design data quality indicators with the objective to answer the following questions:

- Is all the necessary data present in the system?
- Is the data available at the time needed for its usage?
- Is the data compliant with validation or business rules?
- Does the data reflect real world objects?

These questions can be answered using the following types of indicators:

- Anomaly detection: Machine learning algorithms to detect outlying values. **Work in progress**.
- Completeness: Difference in percentage between a measure computed in the source system and the same measure computed in the target system.
- Freshness: Difference in minutes between the current timestamp and the last updated timestamp in the target system.
- Latency: Difference in minutes between the last updated timestamp in the source system and the last updated timestamp in the target system.
- Validity: Any measure computed in target system which does not comply with a validation or business rules.

Inspirata Overview

Inspirata is a revolutionary healthcare IT company, using cutting edge technology to help organizations and patients realize more value from healthcare data. Our goal is to transform care by empowering health systems and providers with the real-time, system-wide data and intelligence needed to improve care quality and the patient experience, and ultimately, the economics of health and wellness.

Inspirata combines the power of an open technology platform, natural language processing and AI, and collaborative clinical applications to bring together disparate patient data and transform it into intelligence. We enable our customers and partners to take advantage of the platform to innovate—rapidly generating a new era of applications to improve healthcare and deliver superior outcomes to patients. Our goal is to make it easy for caregivers across the entire healthcare continuum to gain the insight they need to collaborate and provide the best patient care possible.

Inspirata's Clinical Data Platform provides healthcare institutions the ability to bring all their data together into one place. From this megastore, numerous applications can be built, such as: analytical and data mining engines, customized workflow management, and simplified sharing of critical information with physicians and other healthcare providers worldwide.

Our industry-leading oncology specific Natural Language Processing (NLP) technology provides value by automating manual data curation processes across the oncology service line including cancer case-finding, cancer reporting, clinical trial candidate identification and quality monitoring.

Inspirata's solutions enable our customers to achieve efficiencies, delivery higher quality care, reduce costs, engage in ground-breaking research programs, and participate in partnerships with pharma/biotech companies.