PySpark powered Data Health Check

Automation Using Spark SQL

PySpark has been increasingly becoming popular for data acquisition, profiling and providing the detailed insights of the existing dataset. Data profiling takes the majority of time and effort in any typical data migration/integration implementation activities. Any data professional would love to have an accelerator tool that helps optimize the time spent on this profiling activity

One key aspect of data Profiling is showcasing Health of the Data, where the input data sets/streams are analysed on several dimensions for suitability of end use in the project. It's well established that, if the Health of the dataset can be determined at the initial stage of the implementation, a lot of time can be saved overall, and the project can have telescopic benefits w.r.t. time to final solution. With an effective and quicker way of looking at the data sets in an exploratory way for insights, a lot can be achieved in a shorter time w.r.t. creating the final data sets that can be pushed to model building, evaluation and eventually production.

Data Health Check Solution helps in better and intuitive understanding of the data, gain insight from, and come up with conclusions about the dataset and its domain readiness.

This challenge, in that end, looks forward to come up with a highly functional, user-friendly open source solution (Pyspark based) which can extract the data from heterogeneous sources and profile the datasets using automated manners. The solution should also depict the final dashboard/s to present the health of the data in any open source visualization tool. The solution is expected to bring all these rules finesse to the fore to present the Data Health check

Applied Level	Rule Name	Rule Description
Table	Column Count	No. of Columns in the table
Table	No. Of Empty Columns	No. of Columns having no data for all the Records.
Table	Row Count	Total No. of Records in a Table
Table	Completeness	(Total Non-Empty Records across all Columns)*100/(No. of Columns * No. of Records)
Table	Completeness Excluding Empty Columns	(Total Non-Empty Records for Columns Excluding Fully Empty Columns)*100/(No. of Columns * No. of Records)
Table	Uniqueness	No. or Unique Rows excluding Primary Keys
Table	Health Score	Displays Average of Completeness & Uniqueness
Column	Data Type	Generic Data Types as String, Int, Float, Double, Timestamp based on the Data
Column	Data Fill Ratio	(Total Non-Empty Rows for the Column)*100/(Total No. of Rows in the Table)

Column	Non Empty Count	Includes Non-Empty Values count for a Column
Column	Empty Count	Includes Empty Values count after trimming data along with Values as “NULL” & “null” for a Column
Column	Duplicate Data	Count of Duplicate Values for the Column
Column	Unique Count	Count of Distinct Data Values for the Column
Column	Min Value	Provides Minimum Value of a specific Column
Column	Max Value	Provides Maximum Value of a specific Column
Column	Min Length	Provides Minimum Length of a specific Column
Column	Max Length	Provides Maximum Length of a specific Column
Column	Standard Deviation	Provides Standard Deviation Calculation only for Numeric Column
Column	Average	Provides Average Calculation only for Numeric Column
Column	Median	Provides Median Calculation only for Numeric Column
Column	Variance	Provides Variance Calculation only for Numeric Column

Below are the key aspects to be of the solution which need to be considered while building solution

- Capability of Data extraction from heterogeneous sources
- Automated and reusable approach for Data Profiling
- Entire solution must be built on Open Source tools (leveraging Spark SQL). Choice of Database and Visualization tool is up to the team, however they need to be Open Source