# **Data Challenges and Resolution Documentation**

Group Number – **08**

Project - **Medium Data Analytics Leaderboard System**

Team Members – **Piyush Kumar Sultania**

**Gauri Narendra Bhale**

**Nameera Mohsin**

## **Issue and Resolution - 01:**

**Challenge:** The source data was huge in volume with approximately 1.4 million records and approximately 108 columns. The source data is about the blogs and articles scrapped from “Medium.com”. Due to this high quantity of data, we faced issues while loading the data into Jupyter notebook to perform data profiling and cleansing. We observed a sudden spike in RAM and processing power needed in order to execute the data load in python notebook. Most of our team members even faced system crash couple of times while performing this activity.

**Resolution:** It was difficult to perform the data profiling and wrangling on this dataset, so we had to limit the volume of data to 50-60% of the original data and perform our profiling, cleansing and database load for it. We started loading data from 100k records and went till 700k records and observed an optimum performance on loading to python notebook as well as keeping a wide variety of data across the columns, which is good enough to perform our analytics and provide visualizations and reporting solutions. We performed duplicate check, NULL and Blank values handling, columns renaming and other practices to clean and produce our final data set for graph database design. We also observed that these much data was also getting loaded as nodes in our graph database without any performance defect. Although, this issue was not supposed to happen but as our dataset contains high number of columns the total dataset size was around ~650MB which was not getting easily processed by python execution engine. However, limiting the size of data load did not have any major impact on our deliverables as per our validations and testing results.

## **Issue and Resolution - 02:**

**Challenge:** During the analysis of dataset provided for the project, we figured that there are high number of scrapped columns with field names with wildcard/starts with Tag\_\* which has only Boolean values as 1 or 0. These does not have any representation mentioned anywhere in the present location. Based on the findings and brainstorming, we figured that these fields are just tags associated to each Title/Article present in the dataset. So, each article has 90 odd tags associated to them as columns but very low number of tags were indeed associated with each title (having Tag value as 1). So, lots of columns were insignificant for us and was not needed to be loaded to clean and analyze further. Thus, it was recommended that those certain columns needed to be pivoted as records (by creating column as Category and making those columns as rows) for only those associated tags which has value = 1. So, this was a complex code that we had to write which can not only pivot one column but 90 odd columns and simultaneously pick the active tags only too.

**Resolution:** Once the data was cleaned with operations such as null handling, column renaming etc., we used the “melt” function to current dataframe in order to pivot the Tag\_\* columns and store them as records. We used below columns as id\_vars (identifier columns) and pivot the story-tag columns into records. Thus, we had 14 columns in our main dataset out of 108 columns to analyze and build our visualizations. “Category” column contained all the story tags like Tag\_ai, Tag\_trump, Tag\_ux etc. We also cleaned the category tag values by removing the Tag keyword from the values and capitalizing the records and maintain the clean data protocols.

* Post\_ID
* Title
* Subtitle
* Image
* Author
* Publication
* Year
* Month
* Day
* Tag
* Reading\_Time
* Claps
* Url
* Author URL

## **Issue and Resolution - 03:**

**Challenge:** Based on the dataset profiling and analysis, it was figured that the metadata for the source dataset is not available at Kaggle where the dataset was retrieved from. It is quite evident that performing data wrangling and analytics without the awareness and visibility on metadata is highly difficult and the reports and visualizations presented won’t be highly efficient. It was difficult for the team to understand the KBEs without the metadata information and their relationships and then design the ER model.

**Resolution:** To tackle the issue with unavailability of metadata information, we reached out to the data owner and connected to get the information on the metadata. Although we were not able to fully get the visibility on the data but it was good enough to make sense of the domain and start analyzing and cleansing activities. Additionally, we did some research on the KBEs by the column headers of the source file and also some of the data values (realistic references) and figured the domain of the fields. Finally, we were able to complete our business metadata documentation as well based on our findings.

## **Issue and Resolution - 04:**

**Challenge:** The data profiling report showed that there were certain inconsistencies with the source data like missing values in Title and Author fields, null values and blank spaces, single or multiple whitelines in the column values for some of the fields. It was hard to determine whether the records have to be dropped or such inconsistent data should be replaced with other default values. With such null or blank spaces, the graph database would fail on data integration and load during the creation of database.

**Resolution:** To tackle this problem, we dropped the records having NULL values in Title and Author columns as they are the determinant for the analysis and reporting. Rest other columns were performed with replace strings as “No Subtitle” or “No Publication” etc. marking all the null values in this way. Thus, we were able to maintain the consistency of the data and were able to integrate this source dataset into our neo4j graph database and finally created the nodes and relationship.