# CS481 Data Science

Assignment 1

*February 17, 2020*

**Bilal Hyder** 16I0262
**Bilal Khan** 16K3778

## Q1 Human Centric Data Cleansing Summary

The document/research paper discusses about Data Cleansing and Analysis in Data Sciences. It explains how businesses often collect large volumes of data to make key decisions. Having such a priority for a company there is few to no space of making mistakes or bad decisions. An opposing force regarding this scenario are incorrect or invalid values filled in data cells. The authors describe many ways these values can be discovered, inspected and changed with validation. These may contain a human being on an end or a fully automated system.
*Invalid values may occur due to reasons such as* human error, invalid editing without validation, missing or extra values within the collection and wrong automated collection such as wrong SQL data extraction or masking. Looking at existing data cleaning techniques, the authors describe the following observations:

- Human Involvement
- Using an Automated tool
- Semi Automated approach

Every method has its own pros and cons. **Human Involvement** requires data cleansing at the hand of a professional person who understands as much there is about the domain specially the subject at hand. Though budget consuming and requiring the dispensing of man power this process provides slim chances to occur for such a scenario, nonetheless *a human being is a human after all* and can makes mistakes. **Automated tools** removes the need for human interaction completely and shows promising results, but presents a serious hold up in case of a deadlock such as in a SQL database with columns dispensing on each other for changes, in this it will requires a human presence, this is where **Semi Automated approach** comes in providing an assist by human with most of the progress being handled by a tool such as the data cleansing system *NADEEF and KATARA* as directed by another author cited in the paper.
The paper emphasizes on an architecture for data cleansing with data user presenting error

to the analyst which is then repaired and validated by an expert user, while character human expertise as:

1. Detection - The detection of invalid cells
2. Repairing - Reported errors should be fixable by humans
3. Validation - Expert human analysis for validation on reparing
4. Specification - Humans are enabled to write specifications or a set of rules for detecting and repairing data errors

$$Expertise = \frac{correct}{validated} \tag{1}$$

**Task Allocation** is an important part of data cleansing where an automated tool assigns human experts with related tasks out of the machines domain or an available team of human experts do so among themselves, which then again presents a case of limited budget. **Identification of bottlenecks** represents specifying invalid values correctly with marking a score on each valid repair.

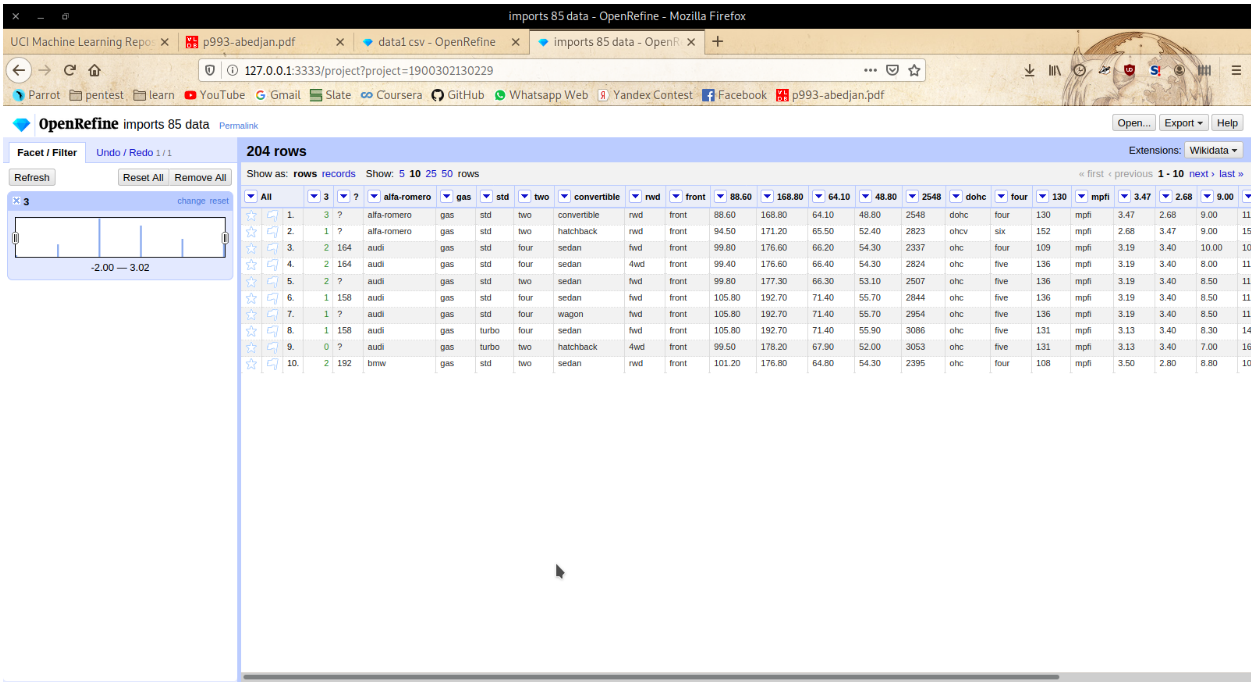
$$Quality = \frac{correct}{validated} \tag{2}$$

## Q2 Data Cleaning Tool

### OpenRefine

OpenRefine is one of the finest available open source data cleaning tools. Formerly known as *Google Refine*. OpenRefine provides many functionality to clean/remove duplicated, mark blanks, arrange datasets along with split and merge based on a specific rule defined. OpenRefine works on the basis of *Faucets* or buckets which isolate a feature or a column for further analysis. There are many types of faucets based on the datasets used in this analysis 2 commonly used were *numeric and alphabetic*. Dataset used were a locally created **data.csv** and a **automobile dataset** available on the UCI Machine Learning Repository.
Locally Created Data
The underlying problem here is missing values, invalid cells more specifically string in place of numeric and unknown header value. After extracting *Open Refine* and starting the localhost load the data in the create new project and click on next.

1. Firstly create a faucet on the id column in this case *column id* and set the cell transform to numeric to visualize the distribution across the column. The range of the data can be adjusted in the *faucet* available on the left pane.
2. To tend to missing values create a faucet on the column that needs adjusting and select blank faucet this will give all the empty cells available in the column, then select "all" and click remove duplicate values. Thus blank values are removed or in other case fill down the column to fill the empty cells with the value of the cell present in the above column edit >fill down.



| All | 3 | ? | alfa-romero | gas | std | two | convertible | rwd | front | 88.60 | 168.80 | 64.10 | 48.80 | 2548 | dohc | four | 130 | mpfi | 3.47 | 2.68 | 9.00 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | 3 | ? | alfa-romero | gas | std | two | convertible | rwd | front | 88.60 | 168.80 | 64.10 | 48.80 | 2548 | dohc | four | 130 | mpfi | 3.47 | 2.68 | 9.00 | 11 |
| 2. | 1 | ? | alfa-romero | gas | std | two | hatchback | rwd | front | 94.50 | 171.20 | 65.50 | 52.40 | 2823 | ohcv | six | 152 | mpfi | 2.68 | 3.47 | 9.00 | 15 |
| 3. | 2 | 164 | audi | gas | std | four | sedan | fwd | front | 99.80 | 176.60 | 66.20 | 54.30 | 2337 | ohc | four | 109 | mpfi | 3.19 | 3.40 | 10.00 | 10 |
| 4. | 2 | 164 | audi | gas | std | four | sedan | 4wd | front | 99.40 | 176.60 | 66.40 | 54.30 | 2824 | ohc | five | 136 | mpfi | 3.19 | 3.40 | 8.00 | 11 |
| 5. | 2 | ? | audi | gas | std | two | sedan | fwd | front | 99.80 | 177.30 | 66.30 | 53.10 | 2507 | ohc | five | 136 | mpfi | 3.19 | 3.40 | 8.50 | 11 |
| 6. | 1 | 158 | audi | gas | std | four | sedan | fwd | front | 105.80 | 192.70 | 71.40 | 55.70 | 2844 | ohc | five | 136 | mpfi | 3.19 | 3.40 | 8.50 | 11 |
| 7. | 1 | ? | audi | gas | std | four | wagon | fwd | front | 105.80 | 192.70 | 71.40 | 55.70 | 2954 | ohc | five | 136 | mpfi | 3.19 | 3.40 | 8.50 | 11 |
| 8. | 1 | 158 | audi | gas | turbo | four | sedan | fwd | front | 105.80 | 192.70 | 71.40 | 55.90 | 3086 | ohc | five | 131 | mpfi | 3.13 | 3.40 | 8.30 | 14 |
| 9. | 0 | ? | audi | gas | turbo | two | hatchback | 4wd | front | 99.50 | 178.20 | 67.90 | 52.00 | 3053 | ohc | five | 131 | mpfi | 3.13 | 3.40 | 7.00 | 16 |
| 10. | 2 | 192 | bmw | gas | std | two | sedan | rwd | front | 101.20 | 176.80 | 64.80 | 54.30 | 2395 | ohc | four | 108 | mpfi | 3.50 | 2.80 | 8.80 | 10 |

Figure 1: OpenRefine Automobile dataset

3. To cater missing headers select the column that needs renaming and edit column >Rename this column.

Automobile Dataset

The base problem regarding cleaning this dataset is removal of duplicates, missing values. It contains 26 features about the origin of an automobile such as its manufacturer, car type and engine etcetera Missing values have been marked with a question mark "?". A copy of the dataset has been provided with this file

- automobile.data
- name.names

1. Regarding duplicates, select a column as the identifier and set that column id to arranged by numeric from the smallest. This will give all the similar values at the top and then select *edit cells >blank down* which leaves the first occurrence of a duplicate and then blanks up all subsequent. After which it is the usual, select *all >remove duplicates*

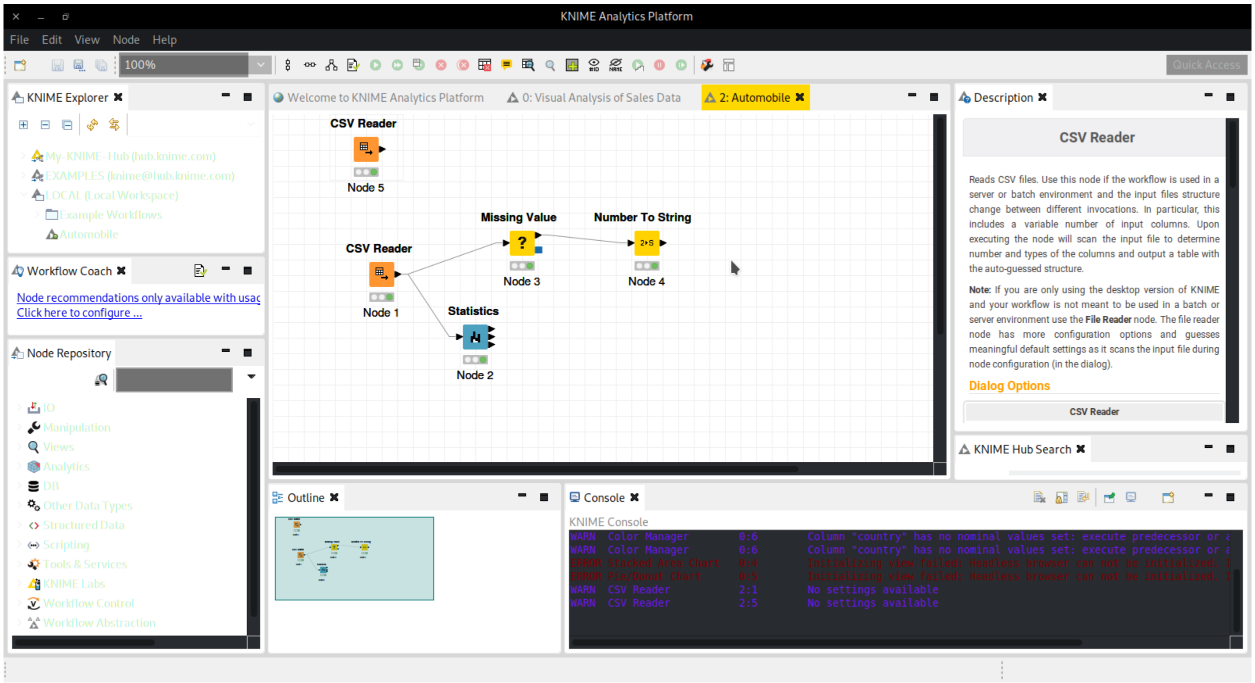
2. A name file has been provided with the dataset containing all the features and description. The headers can be easily renamed through this file.

## Knime

Knime pronounced (ni-em) is graphical based data cleaning tool, somewhat like *rapid miner* also able to implement models such as random forest. Running on Java runtime it works on a nodes based drag and drop structure opposed to the faucet based functionality in Open-Refine. Available through the GNU open source license *Knime* is as easy to use as easily available. For the purpose of simplification and keeping within the specified limit we will applying the same configurations to both the dataset i.e. **automobile and locally generated dataset**.

### Data cleaning in Knime

1. To add data to a project, we need to create a workflow first.
2. Add a *read csv* node to the workflow. Right click it to display further options out which we will select configure, or just hit shortcut F6.
3. In the configure dialogue box direct the node to the data stored in the system, for now we will be working with a csv format, hence select csv file. Click *Apply >Ok*. Remember to check \uncheck row and column header as per the need.
4. Right click on the node to display a dialogue box showing execute option and select it (shortcut F7). The node status will turn green and the uploaded data can now be displayed. String values present in a Interger column will automatically be removed and an empty cell be present to modify as per the user need.
5. Next, drag in a node for statistics, not for our main purpose, but visualizing data is always preferred.
6. Connect the read csv node to the statistics nodes. Similarly this node can be executed and the table viewed.
7. For data processing we will be filling in missing values next. Search for a missing values node and connect to the read csv node. The missing value node has a lot of configurations present for filling in missing values such as fill with mean of the column, previous value, next value or even mean average. You can even select different configuration for every column. Click Apply and then Ok after choosing your preferred settings.
8. How about an integer to string conversion node? This node will unify all columns and make our dataset consistent. Simply configure by adding and removing columns to which the operation is to be implemented on and click execute.



CSV Reader

Reads CSV files. Use this node if the workflow is used in a server or batch environment and the input files structure change between different invocations. In particular, this includes a variable number of input columns. Upon executing the node will scan the input file to determine number and types of the columns and output a table with the auto-guessed structure.

**Note:** If you are only using the desktop version of KNIME and your workflow is not meant to be used in a batch or server environment use the **File Reader** node. The file reader node has more configuration options and guesses meaningful default settings as it scans the input file during node configuration (in the dialog).

Dialog Options

CSV Reader

KNIME Console

```
WARN  Color Manager       0:6      Column "country" has no nominal values set: execute predecessor or a
WARN  Color Manager       0:6      Column "country" has no nominal values set: execute predecessor or a
ERROR Stacked Area Chart  0:4      Initializing view failed: Headless browser can not be initialized. I
ERROR Pie/Donut Chart     0:5      Initializing view failed: Headless browser can not be initialized. I
WARN  CSV Reader          2:1      No settings available
WARN  CSV Reader          2:5      No settings available
```

Figure 2: Knime workflow example