Abdoul-Nourou Yigo
Dr. Hien Nguyen

Big Data and Data Mining

PROJECT 3 REPORT

Blocking Matching

In terms of Blocking and Matching we have used a python Record Linkage toolkit. This library got the necessary functions for the manipulations of a considerable amount of data.
For the matching we got four attributes that are:

- console
- price
- title
- id

in each CSV dataset(all_gamestop.csv, jnl_half_edit2.csv) . According to the content of both CSV tables, the “console” columns are structured the same way. In other words, we could match the consoles from both tables without the need for determination of the threshold because they are exactly the same characters in the two tables. Therefore, we decided to do the blocking on the “title” column by using the BlockIndex( ) function from the recordlinkage library. This function is used to index the column that is blocked and then those indexes can be used to compare the records in the two datasets making subsets of the records instead of indexing the whole sets to make the comparison computation more efficient.

Therefore, for the comparison, we used a class from the recordlinkage library that is called compare() to compare the blocked of records that were blocked to each other. For the comparison of the records we needed to decide what processes to use to make the comparisons accurate because the comparison class got three methods that are:

- string
- exact
- numeric

to effectuate the comparison. As we said, knowing the “title” needed more attention for a good comparison we use the “string” method because the length of the string differs from one record to another one. Using the string function it is required to use another method with the appropriate matching algorithm and threshold. In our case, we use the edit distance algorithm that is the jarowinkler algorithm. This algorithm is scaled from 0 which means that there is not a matching and 1 that constitute an exact match according to Wikipedia. Therefore, according to the record linkage toolkits documentation it is advised to use 0.85 as a threshold that is a good approach to get more match in both datasets. For the “console” attribute we use the compare.exact() method because the console strings’ length is the same for each console receptively in both datasets. We did not compare the “price”. Since the datasets are from different web platforms a product(record) could be the same at different prices.

When everything is set up it was the matter of computing the records that could be matched. After computing the comparison, we need to make sure that there are not repeated records in the merged dataset.

Issues with the recordlinkage Library

We could say that the main problem we went through was to find what are the correct functions from the library to effectuate the matching because we were not familiar with it. We had used some functions that were not doing the matching correctly. For instance, when the did the blocking in the wrong column (attribute) we got inaccurate matching. With some research, we were able to figure out some functions properties that helped us to do the matching correctly.

Data Frame Sizes

Data Frames	all_gamestop.csv	jnl_half_edit2.csv	Pair indexes of all_gamestop.csv & jnl_half_edit2.csv	Matching.csv
Sizes	14118	7477	7954	2325

Additional Cleaning

With the cleaning we have done before matching the dataset was enough to match the data. We did not need to do any additional cleaning.