The Generic and Semantic Profiling of Big Datasets: NYC Open Data

Theodore Hadges[1], Ankush Jain[1], Ruinan Zhang[1]

[1]NYU Tandon School of Engineering

1. Introduction

1.1 Background

Profiling big data is among the most challenging endeavors a data scientist can take on. Millions of human hours have been spent parsing and cleaning data so that it can fit the user's needs and enable them to make better business decisions. In this study, we ran Apache Spark over NYU's 48-node Hadoop cluster, running Cloudera CDH 5.15.0, to generically and semantically profile 1900 datasets from NYC Open Data. We refer to these two profiling methods as Task 1 and Task 2, respectively.

We processed many datasets for testing and optimization of our code, yet still run into memory issues or errors when we attempt to process all 1900 datasets in one pipeline. Therefore, we present the following quantitative results of a subset, 1159 datasets, with the disclaimer that the full collection consists of 1900 datasets and our results describe only this small subset. However, the methods we defined are designed for big datasets and can be used for the entire 1900 dataset collection.

Of the 1159 files we profiled in Task 1, we found 11674 integer columns, 13646 text columns, 1137 date/time columns, and 4527 real number columns. For Task 2, we analyzed 260 columns and we were able to identify the semantic types for 210 columns with a precision of 72.40%.

Aggregation of the 5 most frequent values across single-word text data columns.

1.2 Questions and Hypothesis

Question 1: How can we profile all of this data in the most time efficient and precise manner?

Hypothesis 1: String formatted columns must be composed of multiple data types and it will bottleneck the code execution

Question 2: How can we differentiate between similar classes like neighborhoods and street names?

Hypothesis 2: Perform NER, string matching, regex, soundex or similarity techniques to infer relationships among groups. (If A like B and C like B, then A and C are similar, and of same type as B.)

2. Objective

The aim of this work is to discover, learn, and establish a method to both generically and semantically profile many large data sets in parallel. Task 1, generic profiling, involves the classifying of data types of each column along with basic statistics for that column. Task 2 involves semantic profiling; for example, classifying a column as a "school name", "city agency", or "parks & playgrounds".

3. Methods

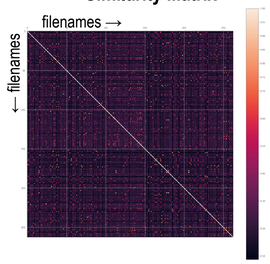
3.1 Task 1: Generic Profiling

We imported all data as pyspark dataframes and used inferSchema() to narrow down the possible data types before classifying. If spark classifies a column as int, real, or date_time then every row in that column is of that type. Otherwise, if there is one entry which is of a different type, spark will classify the whole column as a string. Therefore, if it it infers any type other than string, we can use that type as the classification. In most cases, this doesn't work since many columns are heterogenous. Our next approach is to iterate through the columns of each file and perform builtin functions or use regex. Our main bottleneck is date_time, since for high accuracy classifications of this type, many regex checks need to be made. We solve this by having three different date_time checks ranging from low accuracy and fast to high accuracy and slow, and apply the most approproate one for a given column given its size.

3.2 Task 2: Semantic Profiling

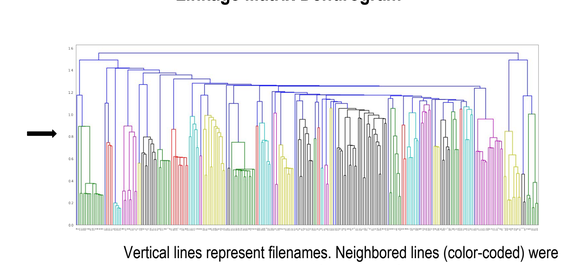
We used regex, NLP, NER, Soundex, and similarity to infer semantics types. Regex is useful for perfect format matches, such as zip code. NLP and NER are useful for names and cities, and similarity is useful for types which have many repeated values, such as areas of study. We calculated the cosine similarity between fille names and created a similarity matrix.

Similarity Matrix



The presence of some bright (similar) dots in the similarity matrix motivated us to find out if files could be clustered by file name.

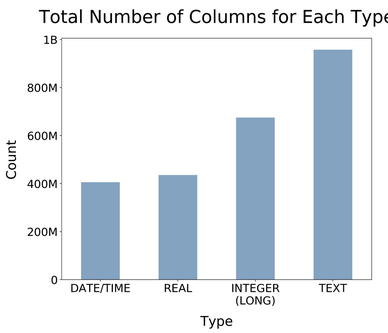
Linkage Matrix Dendrogram

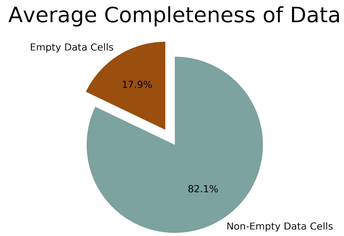


Vertical lines represent filenames. Neighbored lines (color-coded) were mapped to same cluster.

4. Results

Summary of Generic Profiling





- Most data can be classified as either INTEGER or TEXT.
- Upon performing much more precise and resource intensive mining, it might be possible to extract more Date/Time data.
- On average, 82% of the cells in a given dataset are non-empty.
- The 18% empty data cells contribute towards the sparseness in datasets.

4. Results

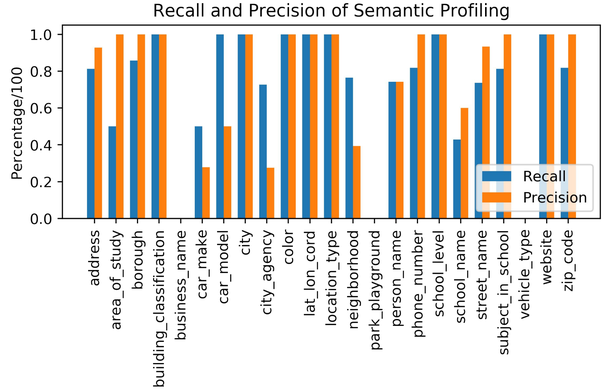
4.1 Task 1 Results: Quantitative Analysis of Generic Profiling

Quantitative Summary (1159 datasets)

Type	mean	std	min_count	max_count	mode
empty cells	20682.11	336203.51	0	16385532	0
non-empty cells	94945.66	1150276.58	0	50846562	212337.6
distinct values	8389.23	201663.02	0	12099654	2
columns	21.42	35.02	1	638	16

4.2 Task 2 Results: Quantitative Analysis of Semantic Profiling

Many categories were classified with high accuracy.



5. Conclusion and Future Study

Our results represent a subset of the 1900 datasets. We need to continue this study but improving the efficiency of our code such that we are able to process all datasets within a few hours. There are also many areas of improvement for accuracy. Once such improvement is the use of frequent itemsets in sentiment analysis. We would like to use frequent itemsets of size 2 or 3 in some cases (rather than just a frequent itemset of singletons). Further we would like to perform NLP processes in distributed manner to scale to to more classes. In the future, we plan on adding this functionality so that we can have better accuracy during the row classification step.

6. References

1. Efficient Algorithms for Mining Outliers from Large Data Sets. Ramaswamy et al., SIGMOD 2000
2. Anomaly Detection: A Survey. Chandola et al, CSUR 2009
3. Ming Hua, Jian Pei: Cleaning disguised missing data: a heuristic approach. KDD 2007: 950-958
4. Ming Hua, Jian Pei: DiMaC: a system for cleaning disguised missing data. SIGMOD Conference 2008: 1263-1266
5. Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD'96)
6. https://www.kdnuggets.com/2018/12/four-techniques-outlier-detection.html
7. Doraiswamy et al., Topological Analysis to Support Event-Guided Exploration in Urban Data. IEEE TVCG, 20(12): 2634-2643, 2014

7. Acknowledgements

Thank you to Professor Julia Stoyanovich and Professor Juliana Freire.