KEEN-HACKATHON

AI-based Automation of

Industrial Data Pre-Processing

By ABB AG

Organizer: DECHEMA e.V.

Contact Person: Dr. Simone Rogg

Date: 05.03.2020

Exclusive Use for the KEEN Hackathon

1. Introduction

Basics

Machine Learning (ML) is much more than training and evaluating a model. While these steps are omnipresent and the focus of many tutorials, the full workflow or lifecycle associated with ML is longer. Figure 1 describes the steps described by Amershi et.al (Amershi *et al.*, 2019).

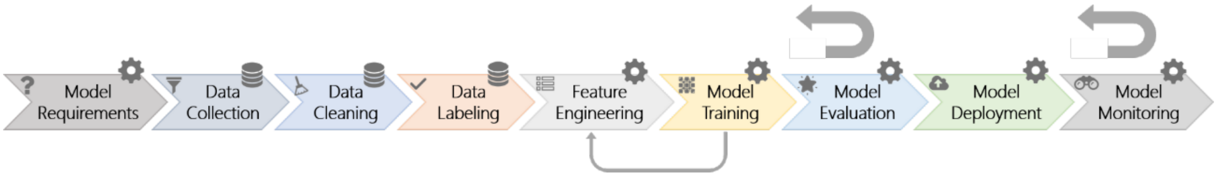


Figure 1- Representation of ML workflow taken from Amershi et.al. "Software Engineering for Machine Learning: A Case Study".

An important step that is often only implicitly mentioned is "data cleaning", which removes noise, low-level inconsistencies, and artefacts. Furthermore, strategies to fill gaps and missing values and flag anomalies and high-level inconsistencies are often applied during this phase. After the cleaning, individual sources of information are merged, technical peculiarities are abstracted, and the data is accessible in a homogeneous and consistent way. It is a key-factor for the overall success. The cleaning is usually done manually, and it is considered to be the "most time consuming and least enjoyable data science task" (Press, 2016).

In a typical industrial context, there are many sources that provide heterogenous, yet interconnected information. On the one hand, this makes the problem easier, as more background knowledge is available, and more checks can be applied. On the other hand, it is harder as the background information is not in a suitable format or marked as sensitive information, and industrial know-how like process engineering or automation engineering is required to consistently read the data.

Objective

The challenge is to create a pipeline, workflow, or tool that either automatically pre-processes a data set to obtain a cleaned version of the dataset or that significantly reduces the manual effort of data cleaning. The input and the cleaned data sets should have the same relevant characteristics and no additional biases or artifacts are introduced by the cleaning.

Why AI?

The motivation for the challenge is to simplify pre-processing and to aim for data-driven pre-processing. Cleaning data sets from industrial setups has often be done iteratively as a ping-pong played between domain experts from industry and data science. This can easily lead to bottle necks in communications as both domains have their own terminology and both types of experts have very limited time. Furthermore, manual cleaning builds upon existing code bases or gained experiences that can be inadvertent source of errors and biases. For example, this has been nicely illustrated in the xkcd comic on data pipelines as given in Figure 2.

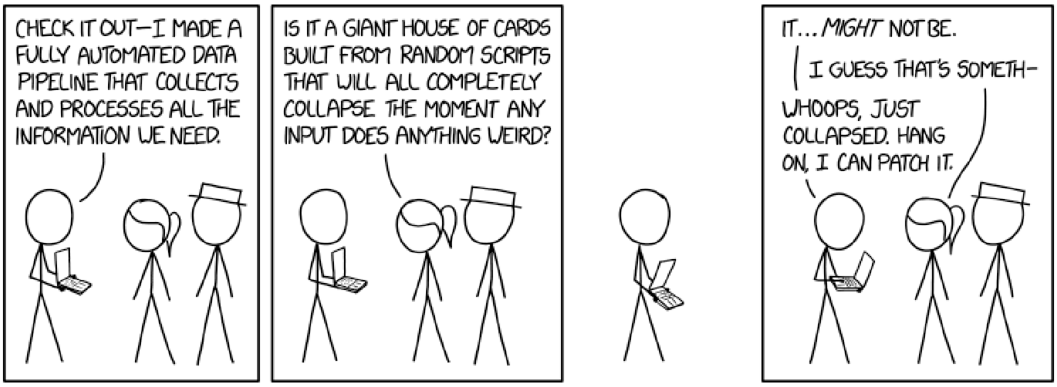

CHECK IT OUT—I MADE A FULLY AUTOMATED DATA PIPELINE THAT COLLECTS AND PROCESSES ALL THE INFORMATION WE NEED.
IS IT A GIANT HOUSE OF CARDS BUILT FROM RANDOM SCRIPTS THAT WILL ALL COMPLETELY COLLAPSE THE MOMENT ANY INPUT DOES ANYTHING WEIRD?
IT... MIGHT NOT BE.
I GUESS THAT'S SOMETH–
WHOOPS, JUST COLLAPSED. HANG ON, I CAN PATCH IT.

Figure 2 Comic on the fragile nature of data pipelines by xkcd in https://xkcd.com/2054/

2. Challenge Description

Setup

The challenge focuses on time-series information. A time-series is a collection of data points indexed by time (https://en.wikipedia.org/wiki/Time_series). While other types of data are present, time-series data is the most common one and requires a large range of pre-processing techniques. For a data set to be usable for the later stages of the ML workflow, a cleaned data set should have the following properties:

Data set (matrix): A comprehensive representation of all (relevant) information in matrix format. Rows correspond to time and columns to signals. The matrix is consistent and “irreducible”, i.e. does not contain redundant information.

Signals (columns): The column (for a signal) contains numeric values, is complete and individual entries in the same column have the same interpretation. Duplicated or highly similar columns should be avoided.

Time axis (row-index): The entries on the time axis are equidistant.

As such, the examples of data issues will include (but are not limited to) missing values, additional noise, outliers, timestamps with and without time zones and correlated signals.

Expected Properties of the Pipeline and its Output (the Cleaned Data Set)

Ideally, the pipeline can produce a cleaned data set as described above for every (reasonable) input. Given the requirement that the pre-processing should not introduce additional biases or make unreasonable assumption, it is accepted that an input set or parts of it cannot be processed. Thus, they should be excluded and for better traceability a log of these parts with optionally reasons or sets of possible actions can be produced instead of a fully cleaned data set.

Furthermore, the following two properties are welcome:

1. “Functional” – the pipeline only depends on the input, e.g. running the pipeline for the same input always produces the same cleaned data set
2. “Idempotent” – when the pipeline gets a cleaned data set as input, it gives the cleaned version which remains the same to the input.

The property being the same / being equal is meant in the sense of approximation of numeric and (normal) computer arithmetic.

The pipeline can optionally produce further information associated to a cleaned data set. This can include, but is not limited to:

- List of performed checks and activities
- Inferred assumptions or distributions
- Confidence of the produced result

Data sets and Restrictions

Each data set is provided as individual archive, containing one or multiple data files with the raw information and, optionally, descriptions or general metadata.

The data and metadata can be downloaded using the following link:

https://keen-austausch.de/s/NgzRTxXrgai7w4R

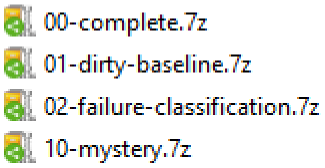
Password: Dwzr6DmG

The convention for data files is that files with the same name and different extension contain the same information, e.g. “data.csv”, “data.xlsx” and “data.pkl” contain the same information just stored in different format, namely comma-separated values, Excel and pickle format. The different formats are for your convenience. Files with the same prefix, but different full name, like “data-01.csv” and “data-02.csv” or “inp-202001.xlsx” and “inp-202002.xlsx”, contain different information that needs to be aggregated. Descriptions and metadata are provided as MarkDown (.md) and as JSON (.json), respectively. They are provided for your convenience and information, so that cross-checking can be done. For example, such file may contain information of time zones. This information is generally not available and relying on its presents should be avoided.

Data sets are further marked as “real” and “example”, respectively. All data sets with label “real” origin form a real plant or test-rig or are created with a process simulation; the number of signals and covered time are realistic. Please be aware, that recordings of test-rigs compared to real plants are denser, i.e. the time between to significant events is small. Data sets with label “example” are artificial simulation and meant to illustrate issues in isolation. The number of signals and total duration is significantly reduced compared to the real data sets.

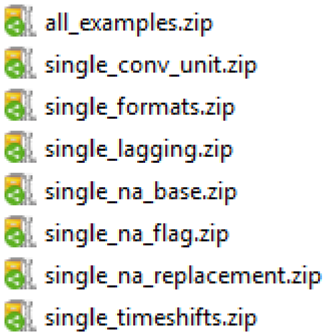
Details

In Figure 3 and Figure 4, the individual files of the two categories are shown.



00-complete.7z
01-dirty-baseline.7z
02-failure-classification.7z
10-mystery.7z

Figure 3 Files in Folder real



all_examples.zip
single_conv_unit.zip
single_formats.zip
single_lagging.zip
single_na_base.zip
single_na_flag.zip
single_na_replacement.zip
single_timeshifts.zip

Figure 4 Files in Folder example

Category “real”

- “00-complete.7z” is a complete example, where the original clean data and the dirty data is provided. The file “info.md” contains a description.
- “01-dirty-baseline.7z” is an archive of 50 examples that contain only the dirty version. Artifacts are like the example of “00-complete.7z”.
- “02-failure-classification.7z” is an archive that contains a variation of the previous cases. In the sub-archive “00-baseline.7z” contains the four scenarios, one normal operation and three with failures. It also contains a list when the failures approximately appeared. The sub-archive “01-with_labels.7z” contains 50 cases, that consists of dirty clones of the baseline cases. The filename can be used an indicator for the label. Similarly, sub-archive “02-without_labels.7z” contains 50 cases, that consists of dirty clones of the baseline cases without label information. Here, the cleaning can also be done in context of the classification problem to detect the proper labels for the cases.
- “10-mystery.7z” is an archive of 996 examples of realistic data sets. They come from a different scenario compared to the previous cases.

Category “example”:

- The archive “all_examples.zip” consists of all the files of the other archives in this folder.
- The archive “single_conv_unit.zip” consists of 500 examples that has (physical) units.
- The archive “single_formats.zip” consists of 500 examples that uses list-orientations instead of matrix-like structures.
- The archive “single_lagging.zip” consists of 500 examples that lagged signals, i.e. two approximately identical signals that are sightly shifted in time.
- The archives “single_na_base.zip”, “single_na_flag.zip” and “single_na_repalcement.zip” consists each of 500 examples that illustrates different modes of not available (na).
- The archive “single_partial_noise.zip” consists of 500 examples that has signals with additional noises for some time windows.
- The archive “single_timeshifts.zip” consists of 466 examples with different time formats and time zones.

Submissions

The jury consists of several senior data scientists and computer scientists that judge technical aspects, creativity, and reasonability of all submissions.

Each submission needs to contain at least a (textual) description of the tool, workflow, or pipeline and two cleaned instances from the data sets. To fully evaluate the submission, it will be tested on mystery data sets that are not provided. As such the provision of source code with instructions for execution and/or a packaged version is strongly recommended. Submissions that have limitations, e.g. cannot process certain examples from the examples, should clearly mark them. Submission without code or working demonstrator are possible and should provide sufficiently detailed information.

Additional materials such as a presentation of core ideas or a pitching video (up to 5 minutes) are welcome and will be considered.

The submission takes place via the platform ipOcean. A manual can be found under the "HELP" website.

Important Notice

The terms and conditions which were accepted while registering and/or participating/downloading the task description in ipOcean include the following section. They apply to all team members.

§4 Data Use

(4) The participants undertake to use the data, metadata, and associated materials provided by the data provider/organizer exclusively for the purpose of solving the task of the hackathon and to delete them from all data storages after the end of the event. A transfer of rights of use to third parties by the participants is not permitted.

(5) The participants undertake to use the data, metadata, and associated materials provided only for the purposes of this event, to treat them confidentially towards third parties and not to pass them on to third parties without the written consent of the data provider/organizer. This confidentiality obligation continues to exist for a period of three (3) years after the end of this event.

3. Additional Literature

- Schmidt, Andreas, Martin Atzmueller, and Martin Hollender. "Data preparation for big data analytics: methods and experiences." *Enterprise Big Data Engineering, Analytics, and Management*. IGI Global, 2016. 157-170.
- Zhang, Aoqian, et al. "Time series data cleaning: From anomaly detection to anomaly repairing." *Proceedings of the VLDB Endowment* 10.10 (2017): 1046-1057.
- Ding, Xiaoou, et al. "Cleanits: a data cleaning system for industrial time series." *Proceedings of the VLDB Endowment* 12.12 (2019): 1786-1789.

- Ratner, Alexander J., et al. "Snorkel: Fast training set generation for information extraction." *Proceedings of the 2017 ACM international conference on management of data*. 2017.
- *https://www.oreilly.com/radar/the-unreasonable-importance-of-data-preparation/*

4. Bibliography

Amershi, S. *et al.* (2019) 'Software Engineering for Machine Learning: A Case Study', in *Proceedings - 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice, ICSE-SEIP 2019*. Institute of Electrical and Electronics Engineers Inc., pp. 291–300. doi: 10.1109/ICSE-SEIP.2019.00042.

Press, G. (2016) *Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says*, *Forbes*. Available at: https://www.forbes.com/sites/gilpress/2016/03/23/data-preparation-most-time-consuming-least-enjoyable-data-science-task-survey-says/.