Ways of working

Corneel den Hartogh

2024-01-16

Table of contents

Ways of working

CEDA is a project within the pilot hub 'Studiedata & AI' (Educational data & AI) of the Npuls program. ADD_LINK This repository is used to document the ways of working within CEDA. This document discusses what type of activities CEDA executes and how this is done. Another document will discuss what our goals are and why these are the goals. In addition, the Npuls program has also standard guidelines and procedures. These are not defined here.

Activities

CEDA delivers the following types of activities:

1. Data Analytics
   - Think of data preparation, modelling and dashboarding of educational data.
   - We build and actively share code for data analytics of national available data through public repositories
2. Data Engineering
   - Think of containerization, connectors to storage, analytics developer environments
   - We build and actively share code and technical configuration for open source tooling that supports the sharing and (re-)usage of the data analytics repositories
3. Alignment & Coordination
   - Think of community contact, reference architecture and the role of analytics in flexibilisation
   - We ensure that the results of the first two activities are aligned with:
     1. the practical needs of (data analists in) educational institutions
     2. the technical needs of the national IT infrastructure
     3. the strategic vision for Dutch tertiary education

These activities all have their own specific guidelines and requirements. However, some overarching ways of work can be defined.

Npuls Guidelines

Most of the work within Npuls is done via secondment. When an agreement is reached, Flex will contact you about the necessary paperwork to set you up. When this is done, you have to fill in your hours via SAM, for instructions see: . ADD_LINKS. You will get a SURF account etc, but we can also give you access to shared spaces via your institutional account if preferred.

Sharing progress

Since we all have other responsibilities (at our primary employer) and don't see each other daily, it is important to share our progress transparently and asynchronously. We do this via a Kanban board in JIRA ADD_LINK. You can see all the stories in the pilot hub, but is advised to focus on those of CEDA or only your own stories in CEDA.

Working together

- Bila's with project lead, half hour per 16 working hours
- Montly office day / afternoon with sprint in Utrecht
- Asynchronous communication? Teams / Slack

JIRA guidelines

For guidelines per story see:

> Task guidelines
>
> - Always use EPIC: Centre for Educational Data Analytics (CEDA)
> - Always use a label to signify the 'product' the task contributes to
> - A clear description per task of what is expected in max 5 sentences
> - A task should take approximately a day of actual work
> - A checklist is often helpful to monitor progress

Open Source

Open source is more than publishing code on github. Also, it is not only a technical or marketing choice but aligned with our place in the public sector. ADD_LINK to document about goals. Practically this means that there are specific guidelines set up for public code in public sector. CEDA strives to conform to these guidelines.

ADD_LINK to Publiccode.yaml

Data analytics

Data analytics guidelines: Repositories

Data analytics repositories are either one of the following:

1. Data product (from raw data to clean, validated and enriched data)
    - Does enriched belong to this or do we want to separate between clean & validated and enriched?
    - In practice it is often mixed (for isntance, you enrich it to help with validation)
2. Analysis product (based on data product but enriched with a model-based prediction)

3. Information product (based on either a data or analysis product, but a visualization or sort of summary)

    - Q: Is dashboard or final product a better name?

Data analytics guidelines: Language

Code of CEDA should be open source and machine readable. We support the following:

- Python
- R
- Tableau
- PowerBI

While Tableau and PowerBI are itself not open source, we can still deliver the logic of the dashboards open. We hope to be able to move from Tableau en PowerBI to fully open source solutions, but currently Tableau and PowerBI match the practical needs of our community

Q: Should we include other tooling or have at some point a survey about tooling?

Data analytics guidelines: Content of repository

1. Every repository has synthetic or dummy 'start' data on which the code runs successfully
2. Every repository is 'complete'. It has all the (synthetic) data, necessary metadata, documentation, etc within.
3. Every repository has a clear structure in line with the language-specific best practices for data science

    - As CEDA we strive to define our own language-specific structure

4. A build file. All code can be run (see 1) by triggering one file
5. Data dictionaries at start and end

    - All the start data has a clear dictionary with metadata and meaning. This also goes for the resulting product
        - As CEDA we strive to have common dictionaries based on the type of repository (data, analysis or information) regardless of specific language

6. There is a config file which has a setting for at least every institution-specific setting

    - For instance, links to file locations, the name or BRIN of the institution, etc

7. An instruction file which explains at least the goal of the data manipulation in this repository and the context (how the acquire the start data or a reference to a working example of a final product).
8. Well styled code, also language specific

   - As CEDA we strive to more precise guidelines about code, in order to make repositories more alike each other

9. All the data files at start and end of the repository are automatically checked by validation rules

   - As CEDA we strive to have uniform data validation rules

10. The main language in a repository is English. This goes for code, comments, documentation and variable names. For specific terms a glossary can be provided.

    - As CEDA we strive to align the glossaries over all repositories

11. All files are machine-readable (.py, .R, .csv., .yaml, md, qmd).

Q: Am I missing something? Or can things be removed?

Q: What about the order?

Data analytics guidelines: Project

One project will often have more than one repository (for instance: a data product repository and an information product repository). So in addition to the repository, there are additional goals:

1. A working demo example oriented at end-users based of synthetic data
2. A blog oriented at end-users to describe the whole project, some interesting real world insights about your own institution and potentially something about the embedding within your own institution
3. One or more workshops with data science professionals from other educational institutions to get it working for them

Data engineering

Data engineering guidelines: Activities

The first set of repositories is about getting from raw data to useful information products. The data engineering part is about making it easy to set up and develop these repositories and making it easy to share and re-use this.

- Developing useful functions and packages that can be used in the data analytics repositories

- Setting up Docker containers and Docker config files to support the usage other data science professionals usage of these repositories

- Experiment and potentially facilitate connection to SURF tooling and other open source solutions

Data engineering guidelines: Packages

Packages follow the standards for packages in the language they are written. This includes:

1. Publication in the most prominent package repositories: Pypi for Python and CRAN for R
2. Usage of package tools that enable (re)building of packages like poetry for Python or devtools for R
3. Github actions on every package repository to automatically check the (re)build
4. At least a README and Github Page about the specific package and its usage

Data engineering guidelines: Containerization

Docker is de-facto standard so we will use this as well. We have the concept of analytics-in-a-box-in-a-box-… highlighting the several possibilities of boxing.

- For instance, taking a data analytics repository, allowing the user to override the config settings to include local files, run all the code and provide the user locally with the resulting 'data product'

- One general repository with information about the different containerization and lessons learned