Data Visualization

DigitalLab@LaPlataforme

Reproducibility

The process we follow depends on the type of data product we are looking to obtain.

Analysis of a dataset

The final product is the description of the phenomenon:

- Population censuses
- Calculation of development indices
- Market segment analysis

Process:

1. Data collection
2. Analysis and exploration
3. Drawing conclusions

Final product:

1. Description and understanding of the phenomenon

Technology research

The final product is a prototype or novel methodology

- Improving the state-of-the-art in machine translation
- Comparison of Models for Recommending Grant Allocations

Process:

1. Data Collection
2. Analysis and exploration
3. Pre-processing of the data set
4. Experimentation to find the best model
5. Drawing conclusions

Final Product

1. Description and understanding of the phenomenon and models
2. Trained model

Data Driven Services

The product is a service that provides answers

- Song recommender
- Automatic translator

Process:

1. Training:
   a. Data collection
   b. Analysis and exploration
   c. Pre-processing of the data set
   d. Experimentation to find the best model
2. Productionalization:
   a. Collection of NEW data to predict
   b. Pre-processing of the data set
   c. Model Application

Final Product:

1. Predictive system

Reproducibility crisis in science

The booming field of artificial intelligence (AI) is grappling with a replication crisis, much like the ones that have afflicted psychology, medicine, and other fields over the past decade.
https://science.sciencemag.org/content/359/6377/725

(Facebook) When combined with the unavailability of code and models, the result is that the approach is very difficult, if not impossible, to reproduce study, improve upon, and extend.
https://arxiv.org/abs/1902.04522

(Google) ML systems have a special capacity for incurring technical debt, because they have all of the maintenance problems of traditional code plus an additional set of ML-specific issues.
https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf

Even the original author sometimes couldn't train the same model and get similar results!
https://petewarden.com/2018/03/19/the-machine-learning-reproducibility-crisis/

Reproducibility aspects

- During the development
  - Version control
  - Systematization of experiments
- Deployment or production start
  - Barriers to adoption and reuse
  - Support for different architectures
- Product consistency over time
  - Changes in the study population
  - Data changes

Common to all software developments

Common to all scientific developments

Particulars of data analysis

Recommendations to achieve better results

During all the process

Methodology

- Document, document, document... and update old documentation.
- Make the original data available. never overwrite them
- Have a Journal document where they informally write what conclusions they drew that day.

Methodology

- Keep a formal record of experimental results

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	Q
1			Dev Results											Dataset	Embedding
2	Log	Name	Accuracy	Ac-std	Precision	P-std	Recall	R-std	F1-Score	F1-std	Time/Feat	Activation	Attention	Config	Char embe
3	echr_none_none zx	18-11-14-14-49	0.661	0.070	0.655	0.086	0.661	0.070	0.652	0.082	None	None	No	Explore	cnn
4		18-11-14-16-39	0.611	0.071	0.630	0.061	0.611	0.071	0.611	0.065	None	None	No	Explore	None
5		18-11-14-18-15	0.635	0.050	0.702	0.060	0.635	0.050	0.641	0.048	None	None	No	Explore	cnn
6		18-11-14-20-17	0.620	0.056	0.669	0.082	0.620	0.056	0.628	0.058	None	None	No	Explore	lstm
7		18-11-14-22-02	0.616	0.036	0.637	0.036	0.616	0.036	0.617	0.028	None	None	No	Explore	None
8	echr_time_sigmoid	18-11-14-22-22	0.633	0.079	0.657	0.079	0.633	0.079	0.636	0.077	Word	Sigmoid	Yes	Explore	lstm
9		18-11-15-00-38	0.640	0.085	0.697	0.089	0.640	0.085	0.648	0.090	Word	Sigmoid	Yes	Explore	cnn
10		18-11-15-02-17	0.636	0.030	0.684	0.038	0.636	0.030	0.644	0.030	Word	Sigmoid	Yes	Explore	cnn
11		18-11-15-04-16	0.653	0.065	0.683	0.072	0.653	0.065	0.660	0.071	Word	Sigmoid	Yes	Explore	None
12		18-11-15-06-01	0.645	0.071	0.688	0.075	0.645	0.071	0.648	0.072	Word	Sigmoid	Yes	Explore	lstm
13		18-11-15-09-54	0.642	0.074	0.689	0.071	0.642	0.074	0.653	0.073	Word	None	Yes	Explore	None
14	echr_time_sigmoid_	18-11-15-10-16	0.651	0.057	0.687	0.053	0.651	0.057	0.651	0.061	Word	Sigmoid	Yes	Definitive	lstm

During the development

Notebooks

Advantages

- Fast configuration
- Fast prototyping
- Interaction during explosion
- Allows to add documentation to the analysis

Disadvantages

- Complicated to use in a control version tool
- Variables can be overwritten easily
- They can't be executed programatically. For example like an script.

Code structure

- Separate exploration from data pre-processing.
- **Not** include data files in your repository.
- Prepare scripts or organize your workflow so it can be done automatically.
- Extract the blocks of code that are repeated. For example: checks and transformations during data reading

Example of code structure

```
project_name
├── INSTALL.md
├── models
│   └── best_knn.py
├── notebooks
│   ├── Prices exploration.ipynb
│   ├── Coordinates exploration.ipynb
│   └── Experiment Results.ipynb
├── README.md
├── preprocess
│   ├── add_airbnb_data.py
│   └── impute_missing_years.py
├── run_preprocess.py
├── run_experiment_best_knn.py
└── tests
    └── test_best_knn.py
```

Environment Setup

- Use control version tools and repositories.
- Record any library that you are using in your project and their dependencies
  - Use virtual environment managers like Conda
  - Use container managers like Docker
- Use documents like README.md to inform possible users how to run your code and what your project is about.

Goal: Anyone can install and recreate your results within 1 year

During deployment

Evaluate the requirements

Find the right tool (which surely already exists). Examples:

- Code that accompanies a paper => make available through a repository
- Image classification library => package using Docker so it can run on any system.
- Processing 10TB of images => using Spark on top of a distributed file system

Does over-engineering of processes exist?

Effort it takes to learn and apply a specific tool

VS

Benefit provided by the tool

Additional material

- Guide: Essential Skills for reproducible Research Computing
- https://awesome.re/ Lists of active open source software recommended by the community. Sorted by teams or by language: