From code banks to software

Lifeline of a coding project and key decisions developers must make

Alin Morariu[1]

[1]Department of Mathematics and Statistics
Lancaster University

Pydata Lancaster, January 2024

Who I am...

Currently

- Ph.D. candidate at Lancaster University
- Research focused on epidemic modeling

Who I am...

Currently

- Ph.D. candidate at Lancaster University
- Research focused on epidemic modeling

Lancaster University

Previously

- Student
  - Mathematics and statistics
- Data scientist
  - Air pollution modeling
  - Trading operations

ST. MICHAEL'S UNITY HEALTH TORONTO

ONTARIO TEACHERS' PENSION PLAN

Manulife

Who I am...

Currently

- Ph.D. candidate at Lancaster University
- Research focused on epidemic modeling

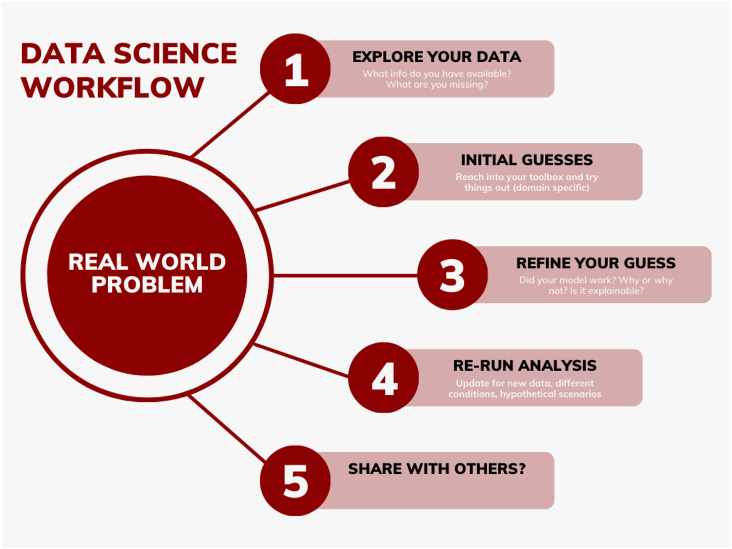
Previously

- Student
  - Mathematics and statistics
- Data scientist
  - Air pollution modeling
  - Trading operations

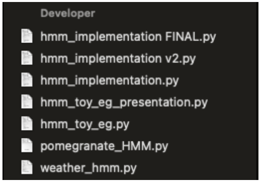
Never

- A software developer

Typical data science project


DATA SCIENCE WORKFLOW
REAL WORLD PROBLEM
1
EXPLORE YOUR DATA
What info do you have available? What are you missing?
2
INITIAL GUESSES
Reach into your toolbox and try things out (domain specific)
3
REFINE YOUR GUESS
Did your model work? Why or why not? Is it explainable?
4
RE-RUN ANALYSIS
Update for new data, different conditions, hypothetical scenarios
5
SHARE WITH OTHERS?

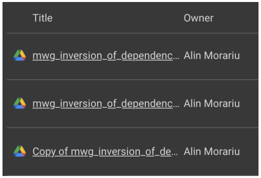
Fall out of data science projects



Developer

- hmm_implementation FINAL.py
- hmm_implementation v2.py
- hmm_implementation.py
- hmm_toy_eg_presentation.py
- hmm_toy_eg.py
- pomegranate_HMM.py
- weather_hmm.py

Figure: An *old* project of mine



Title	Owner
mwg_inversion_of_dependenc...	Alin Morariu
mwg_inversion_of_dependenc...	Alin Morariu
Copy of mwg_inversion_of_de...	Alin Morariu

Figure: A *recent* project of mine

- Overwhelming amount of files
- Duplicate code/work

Fall out of data science projects

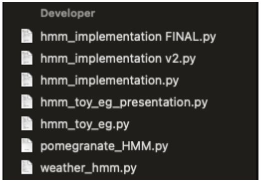
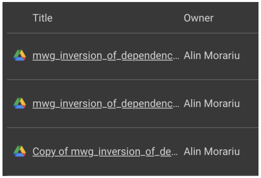

Developer
hmm_implementation FINAL.py
hmm_implementation v2.py
hmm_implementation.py
hmm_toy_eg_presentation.py
hmm_toy_eg.py
pomegranate_HMM.py
weather_hmm.py

Figure: An *old* project of mine



Title	Owner
mwg_inversion_of_dependenc...	Alin Morariu
mwg_inversion_of_dependenc...	Alin Morariu
Copy of mwg_inversion_of_de...	Alin Morariu

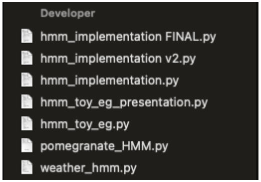
Figure: A *recent* project of mine

- Overwhelming amount of files
- Duplicate code/work

$\Rightarrow$ Leads to confusion, redundancy

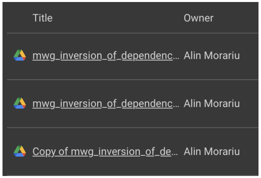
Checkpoint!

Fall out of data science projects


Developer

hmm_implementation FINAL.py
hmm_implementation v2.py
hmm_implementation.py
hmm_toy_eg_presentation.py
hmm_toy_eg.py
pomegranate_HMM.py
weather_hmm.py

Figure: An *old* project of mine



Title	Owner
mwg_inversion_of_dependenc...	Alin Morariu
mwg_inversion_of_dependenc...	Alin Morariu
Copy of mwg_inversion_of_de...	Alin Morariu

Figure: A *recent* project of mine

- Overwhelming amount of files
- Duplicate code/work

$\Rightarrow$ Leads to confusion, redundancy

Checkpoint!

Code bank

Fall out of data science projects

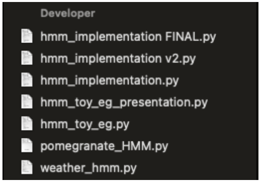
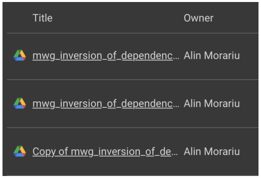

Developer
hmm_implementation FINAL.py
hmm_implementation v2.py
hmm_implementation.py
hmm_toy_eg_presentation.py
hmm_toy_eg.py
pomegranate_HMM.py
weather_hmm.py

Figure: An *old* project of mine



Title	Owner
mwg_inversion_of_dependenc...	Alin Morariu
mwg_inversion_of_dependenc...	Alin Morariu
Copy of mwg_inversion_of_de...	Alin Morariu

Figure: A *recent* project of mine

- Overwhelming amount of files
- Duplicate code/work

$\Rightarrow$ Leads to confusion, redundancy

Checkpoint!

Code bank

A collection of useful, but not necessarily organized bits of code that you can reference later

Fork in the code

Once you have a code bank, you have some decisions to make:

1. Leave it as is </>
   - *Pros*: No additional work required, solves a specific problem quickly
   - *Cons*: Easily forgotten, tends to be messy/disorganized, hard to share

Fork in the code

Once you have a code bank, you have some decisions to make:

1. Leave it as is </>
   - *Pros*: No additional work required, solves a specific problem quickly
   - *Cons*: Easily forgotten, tends to be messy/disorganized, hard to share
2. Share your work with the world!
   - *Pros*: Acts as a portfolio, makes knowledge more accessible, easier to reuse in the future
   - *Cons*: requires extra work (cleaning and generalizing)

Fork in the code

Once you have a code bank, you have some decisions to make:

1. Leave it as is </>
   - » *Pros*: No additional work required, solves a specific problem quickly
   - » *Cons*: Easily forgotten, tends to be messy/disorganized, hard to share
2. Share your work with the world!
   - » *Pros*: Acts as a portfolio, makes knowledge more accessible, easier to reuse in the future
   - » *Cons*: requires extra work (cleaning and generalizing)

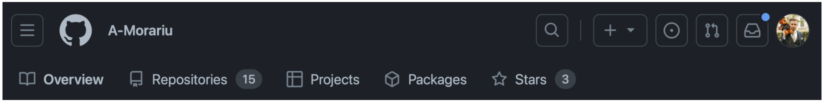


Figure: One option for sharing code is GitHub

Generalizing your code

Important question: How do I want others interacting with my code?

Generalizing your code

Important question: How do I want others interacting with my code?

- Start to separate code into layers

Generalizing your code

Important question: How do I want others interacting with my code?

- Start to separate code into layers
  - Front end - pieces that others are interacting with directly
  - Back end - code that ensures front end runs smoothly and predictably

Generalizing your code

Important question: How do I want others interacting with my code?

- Start to separate code into layers
  - Front end - pieces that others are interacting with directly
  - Back end - code that ensures front end runs smoothly and predictably

Development Trade-off

Rigid, structured interfaces	Flexible, easy to use interfaces
$\Rightarrow$ Makes developers happy	$\Rightarrow$ Makes users happy

Generalizing your code

Important question: How do I want others interacting with my code?

- Start to separate code into layers
  - Front end - pieces that others are interacting with directly
  - Back end - code that ensures front end runs smoothly and predictably

Development Trade-off	
Rigid, structured interfaces	Flexible, easy to use interfaces
⇒ Makes developers happy	⇒ Makes users happy

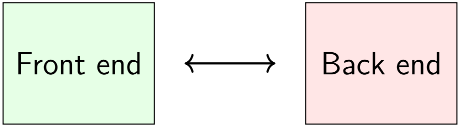


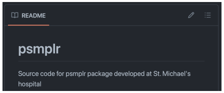
Figure: A software package

Documentation

We have to tell people **how** to use our code?

Easy

- Include a `README.md` file
- Some descriptions & comments within the code
- Fork a Github repository

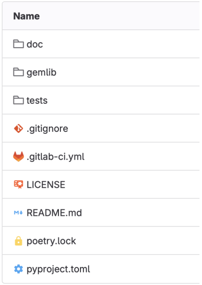

README

psmplr

Source code for psmplr package developed at St. Michael's hospital

Hard

- Include a `README` file
- Handle dependencies
- Detailed descriptions of functions and examples


Name
doc
gemlib
tests
.gitignore
.gitlab-ci.yml
LICENSE
README.md
poetry.lock
pyproject.toml

Putting your project manager hat on

- As libraries mature so do the requirements for managing them

Putting your project manager hat on

- As libraries mature so do the requirements for managing them
  - Early stages - research, development, expand feature set
  - Later stages- refine features, expand user base, handle bugs

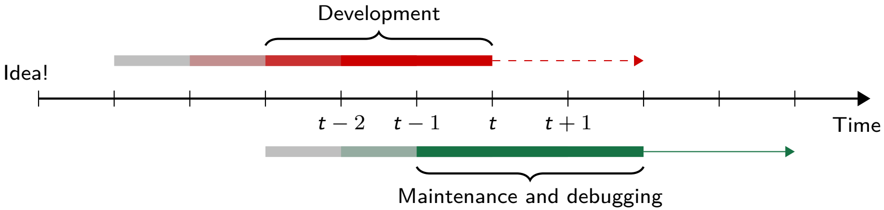


Figure: Time focus shifts over the lifeline of a project from research and development to maintenance and refinement

THANK YOU!

Hope you found some inspiration to take into your next data project! :)

Contact

Personal website: a-morariu.github.io[a]
Email: a.morariu@lancaster.ac.uk

[a]Slides can be found here

🙋?