

Can PyCaret simplify Machine Learning?

Published : 19 June 2020 (updated 22 June 2020)
Status : CONCEPT
Analyst : Maikel Mardjan (www.nocomplexity.com)



(c) 2020 Nocomplexity.com - This work is licensed under the [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)

Key question

The key question of this simple report is:

Does PyCaret solve complexity when applying machine learning technology?

The problem

The key problem that PyCaret seems to solve is:

Making it easier to create and experiment with machine learning models.

So the claim is that you save you time when using this product.

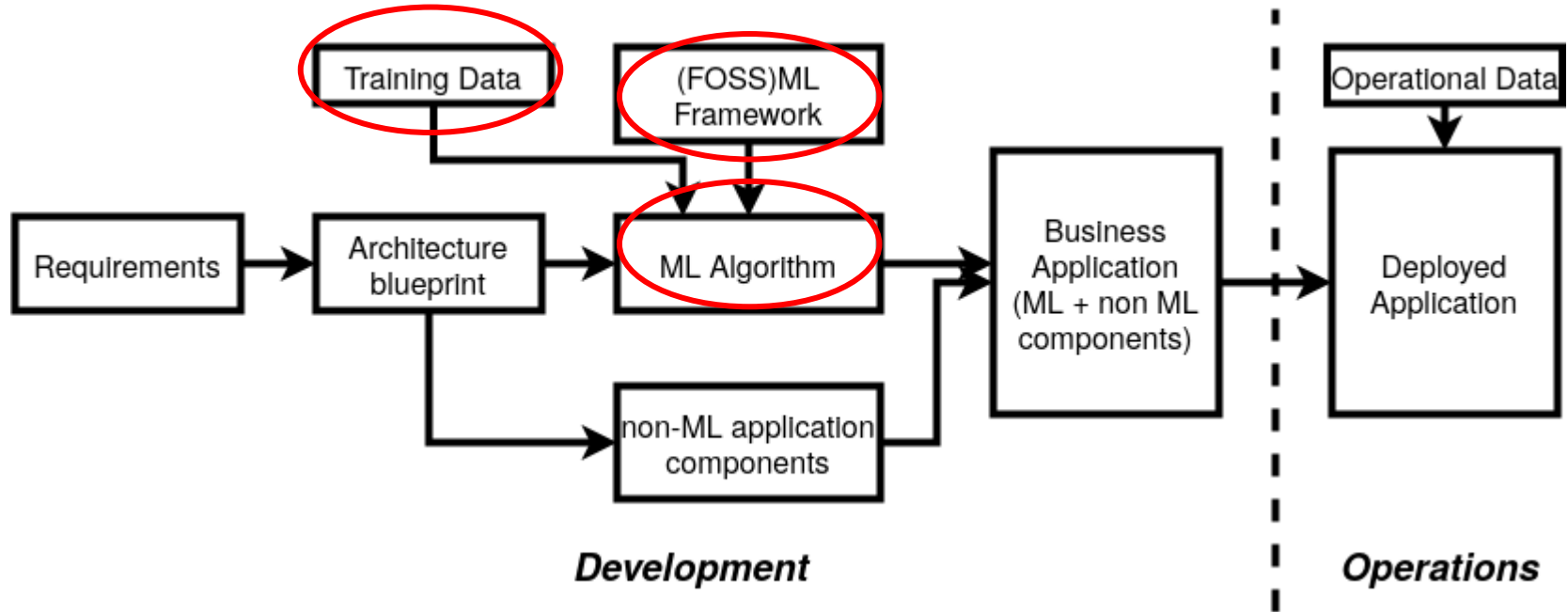
And saving time means:

- More time to implement ML into a business product
- Reduces time to market and
- Saving cost

Position in the business ML workflow

PyCaret aims to simplify ONLY the typical machine learning tasks.

So these are the tasks indicated with red in the figure below:



Observations

The PyCaret website (<https://pycaret.org/>) is loaded with buzzwords. Using buzzwords, especially when promoting complex IT technology, means you are faced with empty and misleading concepts. Machine learning is already technology that can be overwhelming complex. Buzzwords should be avoided.

- PyCaret has good looking documentation. Filled with code and examples. Documentation makes me wonder what was first: The tutorials or the docs.
- PyCaret has some nice time functions that can save you time.
- PyCaret documentation seems focussed on starting ML engineers.

Observations (2)

- PyCaret has no open QA process, including published tests sets on how code is automated tested. A regression test suite is missing.
- Git statistics:
 - Last (public) git commit Date: Tue Jun 2 19:56:51 2020 -0400
 - Initial commit: Sat Nov 23 13:40:48 2019
 - Number of authors on code :6
- Security: PyCaret use can introduce new risks. Depending on your specific context where you use PyCaret for real business use cases you MUST perform a risk assessment. By default PyCaret has no special features that solves common ML security issues.
- Privacy: PyCaret works on unencrypted data. Real use cases means too often working with privacy sensitive information. PyCaret offers no special protection for handling privacy sensitive data streams.

Observations (3)

Criteria	
Architecture and design	The architecture is based on creating an API layer on top of other ML libraries. Since ML APIs can have many options, PyCaret makes some nice default choices for you. The architecture and design of PyCaret is not (yet) complex. So clean and simple to follow.
Maintainability	Due to the clean and mean design, maintaining core PyCaret files is currently no complex job. But since PyCaret has many external dependencies there is a large risks that maintainability becomes complex and time consuming.
Flexibility	You can use the PyCaret functions in multiple ways. Since the source files are not that difficult to understand, creating improvements for your specific needs is no complex tasks.
Documentation	Good solid documentation. The way to use PyCaret is explained by code examples and tutorials.

Observations (4)

Criteria	
Integration aspects	Integrating PyCaret ML pipelines in to your 'own' CICD ML experimentation and deployment pipeline is possible. But you have to do this yourself. But since the core of PyCaret is based on defacto Python ML standards (IPython,numpy,sklearn, notebooks, etc) integration to suite your own needs is simple.
Support	There is currently (June 2020) a small but growing way to get support by using Github Issues.
Legal	Software is FOSS and contains a MIT License text (https://www.gnu.org/licenses/license-list.html#Expat)
Openness	PyCaret has (no) solid community yet. There is not yet an open process on the website that outlines how the projects works for new contributors. E.g. a description like the C4 guidelines (https://rfc.zeromq.org/spec/42/) would be nice.

Conclusions and recommendations

- PyCaret is a product that can offer a simpler way to deal with Machine Learning complexity in the experimental phase.
- Hands-on experimenting with machine learning algorithms requires deep knowledge on various field and strong (unix) development skills. E.g. ML Algorithms, Data science, Numpy, Python. Experimenting usually means that you need to do a real deep dive into used algorithms. The question is whether PyCaret makes this really easier or increases the complexity by adding an extra layer on top of used ML frameworks.
- Incorrect choice of algorithm for a chosen task is a common source for quality concerns when using ML. PyCaret helps improving quality by testing multiple algorithms.

Conclusions and recommendations (2)

- Besides programming errors the ML algorithms, also programming errors are possible in PyCaret. This makes already complex task of finding and solving problems when using ML not easier.
- PyCaret has no open QA process, including published test sets on how code is automated tested.

Summarized:

PyCaret has potential for simplifying the experimenting phase in machine learning.

It is now ready for early adopters, so give it a try and help the project forward!

